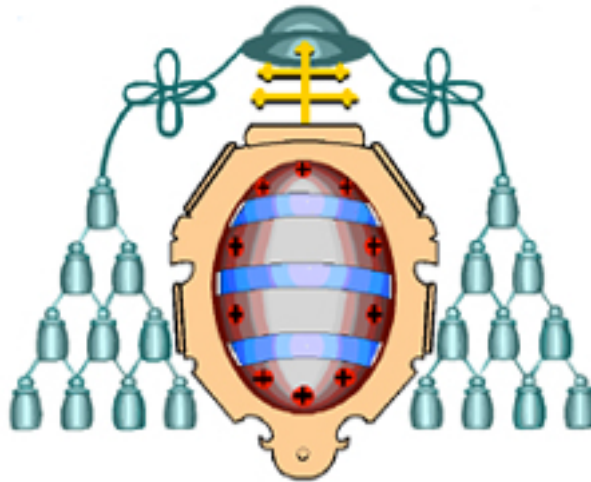


# UNIVERSIDAD DE OVIEDO



**Departamento de Informática**

**Tesis Doctoral**

**Metodología para la aplicación de ingeniería  
dirigida por modelos a la construcción de  
aplicaciones multigénero y multiplataforma  
por parte de los expertos del dominio**

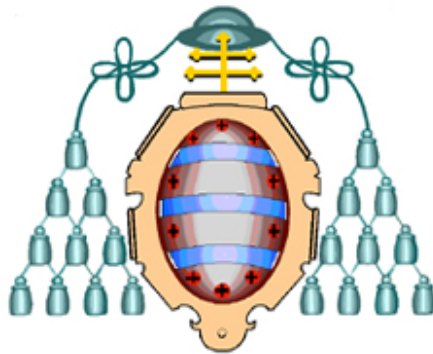
AUTOR: Jaime Solís Martínez

DIRECTORES: Dr. Juan Manuel Cueva Lovelle

Dr. Vicente García Díaz



**Metodología para la aplicación de ingeniería  
dirigida por modelos a la construcción de  
aplicaciones multigénero y multiplataforma  
por parte de los expertos del dominio**



**UNIVERSIDAD DE OVIEDO**  
**DEPARTAMENTO DE INFORMÁTICA**

TESIS DOCTORAL

Presentada por

Jaime Solís Martínez

Para la obtención del título de Doctor por la Universidad de Oviedo

Dirigida por

Dr. Juan Manuel Cueva Lovelle y Dr. Vicente García Díaz





## RESUMEN

El modelado de procesos de negocio (BPM en inglés) es la disciplina que permite a los expertos de un dominio concreto modelar sus procesos para que luego éstos sean interpretados por el personal informático. De esta forma los modelos pueden servir para realizar la captura de los requisitos. Para la realización del modelado existe una notación estándar denominada BPMN (Business Process Modeling Notation) que cubre un rango amplio de posibilidades en lo que a propiedades de los procesos se refiere. Sin embargo, esta circunstancia se vuelve en contra de la notación debido al aumento de complejidad que requiere tener soporte para todo tipo de procesos, tal y como se ha demostrado en algunos estudios citados dentro de este documento.

Es por ello que uno de los objetivos de este trabajo es definir una metodología por niveles para la aplicación de BPM que permita realizar el modelado de los procesos de negocio a través de un conjunto de entidades que se adapte al nivel de conocimientos del experto del dominio y a las necesidades del proceso en si. De esta forma se busca reducir la complejidad de uso existente en BPMN. Además, utilizando los artefactos proporcionados por la plataforma .NET, se pretende construir dos aplicaciones que sirvan de apoyo para esta propuesta: en primer lugar una herramienta de modelado que utilice las entidades contenidas en la metodología para definir los procesos y almacenarlos en formato electrónico, y en segundo lugar un generador de código que analice los ficheros de los procesos de negocio y genere una aplicación específica que ofrezca soporte para el proceso modelado.

A través de la utilización de la metodología para el modelado de varios procesos de negocio reales intentamos comprobar su adaptación a los conocimientos de los usuarios y a los distintos dominios. Al mismo tiempo se han establecido pruebas que nos permitan conocer algunos aspectos fundamentales como el grado de usabilidad de la herramienta de modelado, la complejidad de comprensión de las entidades disponibles en los niveles establecidos dentro de la metodología y la mejora que supone la aplicación de la metodología para la construcción de aplicaciones informáticas.



## **PALABRAS CLAVE**

BPM, BPMN, jPDL, SBPMN, redes de Petri, WF-Nets, BPLOM, BPMN MUSIM, metodología de modelado, generación de código.



## **ABSTRACT**

Business Process Modeling (BPM) is the discipline that allows experts from a specific domain to model their processes so these can be interpreted later on by the informatics personnel. In such way, these models can be used to capture the requirements of the future application. In order for the experts to model their processes there is a standard notation called BPMN (Business Process Modeling Notation) that tries to give support for a wide range of process types. However, this aspect has turned against the notation due to the increased complexity required to have support for all types of processes, as has been shown in some studies cited in this document.

Thus, one of the objectives of this work is to define a level oriented methodology for the usage of BPM that enables the modeling of business processes through a set of symbols that adapts to user's knowledge as well as to the specific process' needs, trying to reduce the complexity found in BPMN. Furthermore, using the artifacts provided by the .NET platform we aim to build two applications that support this proposal: first, a modeling tool that uses the methodology's elements to define processes and store them in electronic format and, secondly, a code generator to analyze a business process file and create a specific application with support for the business process modeled.

Through the use of this methodology for modeling various real business processes we try to demonstrate its suitability for adapting to user's knowledge and to the different domains. At the same time we have scheduled some usability tests which allow us to explore several key aspects like the degree of usability of the modeling tool, the complexity of the different levels present in the methodology and the improvement introduced by the methodology in the software development process.



## **KEYWORDS**

BPM, BPMN, jPDL, SBPMN, Petri nets, WF-Nets, BPLOM, BPMN MUSIM, modeling methodology, code generation.





## **AGRADECIMIENTOS**

*Esta Tesis ha sido la culminación de la etapa de estudios que comenzó en el año 2004 con la Ingeniería Técnica en Informática de Sistemas. Desde entonces he colaborado con muchas personas a las que me gustaría mostrar mi agradecimiento.*

*En primer lugar quisiera agradecer a mis directores, Dr. D. Juan Manuel Cueva Lovelle y Dr. D. Vicente García Díaz, todo el trabajo realizado conmigo. Desde que comenzara mis estudios universitarios hasta este mismo momento me han guiado, aconsejado y ayudado.*

*Agradezco a Jordán y al resto de mis compañeros del OOTLAB, situado en la Facultad de Ciencias, la colaboración que me han brindado durante la realización de este trabajo. No sería justo olvidarme en este momento de mis compañeros del Máster de Ingeniería Web, con los que compartí las clases y trabajos que hicieron posible mi entrada a los estudios de doctorado: ¡Cómo iba a olvidarme de los lunes de entrega!. En esta ronda de agradecimientos tampoco quisiera olvidarme de mis compañeros de carrera, con los que comencé a meterme en el mundo de la informática a través de aquellas asignaturas “poco informáticas” que tan poco nos gustaban: física, cálculo, numérico... mejor no seguir con la lista.*

*Y por supuesto quisiera agradecer el apoyo constante de mis padres y de Lola, que han hecho muchos esfuerzos para que yo haya llegado a este punto. Habéis vivido tanto los malos como los buenos momentos durante el desarrollo de este trabajo, muchas gracias por “soportarme” durante todos ellos.*

***¡Muchas gracias a todos!***

*Oviedo, Asturias (años 2011 a 2014)*



# ÍNDICE DE CONTENIDOS

## Planteamiento del Problema ..... 29

Introducción.....	31
1.1 Introducción .....	31
1.2 Motivaciones.....	32
1.3 Hipótesis.....	34
1.4 Objetivos de la tesis .....	35
1.5 Organización del documento.....	38
Metodología y desarrollo de la investigación .....	41
2.1 Introducción .....	41
2.2 Metodología de trabajo .....	42
2.3 Desarrollo temporal de la investigación.....	44

## Conceptos, Soluciones y Tecnologías

## Existentes..... 47

Business Process Modeling.....	49
3.1 Introducción .....	49
3.2 Esquema de funcionamiento de BPM .....	50
3.3 Beneficios derivados de la aplicación BPM .....	51
3.4 Notaciones de modelado relacionadas con BPM .....	52
3.5 Redes de Petri: Otra alternativa para el modelado de procesos.....	79
3.6 Conclusiones.....	89
Herramientas y plataformas con soporte para el modelado de procesos de negocio .....	91
4.1 Introducción .....	91
4.2 Herramientas con soporte para el modelado de procesos.....	92
4.3 Plataformas con soporte para el modelado y la ejecución de procesos .....	98
4.4 Conclusiones.....	109
MDE: Ingeniería dirigida por modelos .....	111
5.1 Introducción .....	111
5.2 Problemas del desarrollo de software tradicional.....	112
5.3 Ciclo de vida del desarrollo de software .....	116
5.4 Conceptos generales de la ingeniería dirigida por modelos.....	117
5.5 El metamodelo: nexo de unión con el modelado de procesos .....	120
5.6 Comparación de MDE con el desarrollo tradicional.....	129

5.7 Conclusiones .....	131
<b>Desarrollo de la propuesta .....</b>	<b>133</b>
Metodología de Modelado por Niveles para BPM .....	135
6.1 Introducción .....	135
6.2 BPLOM: Business Process Level Oriented Methodology .....	136
6.3 Presentación de los niveles de la metodología.....	139
6.4 Entidades descartadas en la versión inicial de la metodología.....	148
6.5 Comparación de ejemplo de modelado entre BPMN y BPLOM .....	151
6.6 Conclusiones .....	155
Caso de estudio .....	157
7.1 Introducción .....	157
7.2 Tipos de aplicaciones para la aplicación de la metodología.....	158
7.3 Modelado de procesos de negocio tradicionales con la metodología .....	160
7.4 Modelado de la lógica de un videojuego para dispositivos móviles .....	191
7.5 Modelado de una aplicación de catálogo para dispositivos móviles .....	211
7.6 Conclusiones .....	216
<b>Desarrollo de prototipos y evaluación.....</b>	<b>217</b>
Desarrollo de prototipos .....	219
8.1 Introducción .....	219
8.2 Selección de la plataforma para el desarrollo de los prototipos .....	220
8.3 Estructura de aplicación de la metodología por niveles .....	223
8.4 BPLLevel Modeler: herramienta de modelado .....	227
8.5 BPLLevel generator: herramienta de generación de código.....	234
8.6 Ejemplos de uso de los prototipos para la generación de aplicaciones .....	236
8.7 Conclusiones .....	243
Evaluación .....	245
9.1 Introducción .....	245
9.2 Evaluación de la metodología por niveles.....	246
9.3 Evaluación de los prototipos presentados .....	256
9.4 Conclusiones .....	275
<b>Conclusiones .....</b>	<b>277</b>
Conclusiones y trabajo futuro .....	279
10.1 Síntesis del trabajo desarrollado .....	279

10.2 Verificación y evaluación de los objetivos .....	281
10.3 Ámbitos de aplicación.....	283
10.4 Principales aportaciones .....	284
10.5 Principales resultados de la investigación.....	285
10.6 Líneas de investigación y trabajo futuro .....	286
<b>Bibliografía y referencias .....</b>	<b>289</b>
<b>Apéndices.....</b>	<b>297</b>
<b>Glosario y diccionario de datos .....</b>	<b>311</b>
<b>Índice alfabético .....</b>	<b>315</b>



## ÍNDICE DE FIGURAS

Figura 1: Metodología de investigación utilizada.....	43
Figura 2: Eventos incluidos en BPMN [BPMN] .....	57
Figura 3: Estructura de contenedores y contenido.....	60
Figura 4: Flujo de mensaje entre contenedores .....	60
Figura 5: Diagrama de conversación [BPMN] .....	62
Figura 6: Representación de tarea de coreografía .....	62
Figura 7: Indicador de múltiples participantes .....	62
Figura 8: Subproceso de coreografía.....	63
Figura 9: Diagrama de coreografía [BPMN] .....	63
Figura 10: Ejemplo de diagrama BPMN [BPMN] .....	64
Figura 11: Gráfica de uso de símbolos de BPMN.....	65
Figura 12: Representación de una actividad en UML AD .....	66
Figura 13: Acción de UML Activity Diagrams .....	67
Figura 14: Restricciones aplicadas a una acción.....	67
Figura 15: Punto de inicio .....	67
Figura 16: Punto de fin para actividades.....	68
Figura 17: Punto de fin de flujo .....	68
Figura 18: Conexión de dos acciones .....	68
Figura 19: Entidad decisión de UML AD .....	68
Figura 20: Entidad fork .....	69
Figura 21: Representación de un objeto.....	69
Figura 22: Flujo de un objeto factura .....	69
Figura 23: Objeto de tipo datastore .....	70
Figura 24: Representación de una partición.....	70
Figura 25: Ejemplo de diagrama de actividad UML.....	71
Figura 26: Punto de inicio de proceso en jPDL.....	72
Figura 27: Puntos de fin de jPDL.....	72

Figura 28: Actividad o estado del lenguaje jPDL .....	73
Figura 29: Decisión y fork de jPDL.....	73
Figura 30: Proceso de negocio de ejemplo modelado con jPDL.....	74
Figura 31: Representación de las tareas en SBPMN.....	75
Figura 32: Conjunto de eventos definidos en SBPMN [Fernández 2008].....	76
Figura 33: Representación de la decisión en SBPMN .....	76
Figura 34: Conjunto de objetos de datos propuestos en SBPMN [Fernández 2008].....	77
Figura 35: Ejemplo de proceso de negocio modelado con SBPMN.....	78
Figura 36: Red de Petri representando una máquina .....	79
Figura 37: Representación de las condiciones AND, OR y XOR.....	84
Figura 38: Disparadores de las WF-Nets.....	86
Figura 39: Interfaz de usuario de la herramienta Microsoft Visio .....	92
Figura 40: Interfaz de usuario de Enterprise Architect.....	93
Figura 41: Interfaz gráfico de Magic Draw .....	94
Figura 42: Interfaz de Altova UModel.....	94
Figura 43: Yaoqiang BPMN Editor mostrando un proceso de negocio.....	95
Figura 44: Herramienta BP-VA.....	96
Figura 45: Interfaz de usuario de Facilis BPMN.....	96
Figura 46: Esquema de componentes de ARIS Platform.....	99
Figura 47: Interfaz de Appian Process Modeler.....	100
Figura 48: Interfaz de usuario de Bonita Studio .....	101
Figura 49: Entorno de trabajo de jBPM dentro de Eclipse .....	102
Figura 50: Proceso de inserción de incidencias en jBPM.....	103
Figura 51: Proceso de resolución de incidencias en jBPM .....	103
Figura 52: Entorno de trabajo de Windows Workflow Foundation.....	105
Figura 53: Proceso de inserción de incidencias según WWF .....	106
Figura 54: Proceso de resolución de incidencias según WWF .....	106
Figura 55: Comparativa de ciclos de vida .....	116



Figura 56: Diagrama de conceptos básicos de MDE .....	117
Figura 57: Diagrama de paquetes de BPDM.....	121
Figura 58: Modelo de composición de BPDM.....	122
Figura 59: Modelo de proceso o procedimiento de BPDM .....	122
Figura 60: Núcleo del modelo de actividad de proceso de BPDM .....	123
Figura 61: Flujos de acción del modelo de actividad de proceso de BPDM .....	123
Figura 62: Paso de proceso del modelo de interacción de proceso de BPDM.....	124
Figura 63: Flujo de mensaje del modelo de interacción de proceso de BPDM .....	124
Figura 64: Interacción del modelo de interacción de proceso de BPDM.....	125
Figura 65: Esquema del metamodelo de BPMN 2.0.....	126
Figura 66: Jerarquización del elemento compuerta.....	127
Figura 67: Jerarquización del elemento flujo secuencial.....	127
Figura 68: Diagrama BPMN de ejemplo .....	128
Figura 69: Pirámide de complejidad y expresividad de BPLOM .....	137
Figura 70: Estudio sobre elementos BPMN más usados según Muehlen y Recker .....	140
Figura 71: Comparación entre BPMN y SBPMN .....	141
Figura 72: Proceso de encargo de pizza modelado con BPMN .....	151
Figura 73: Subproceso de realización del encargo de la pizza.....	152
Figura 74: Subproceso de tratamiento del encargo.....	153
Figura 75: Subproceso de preparación del encargo .....	153
Figura 76: Subproceso de reparto del encargo.....	154
Figura 77: Subproceso de inserción de incidencias según el nivel 0 .....	161
Figura 78: Subproceso de resolución de incidencias según el nivel 0.....	162
Figura 79: Subproceso de inserción de incidencias según el nivel 1 .....	163
Figura 80: Subproceso de resolución de incidencias según el nivel 1.....	163
Figura 81: Subproceso de inserción de incidencias según el nivel 2 .....	164
Figura 82: Subproceso de resolución de incidencias según el nivel 2.....	165
Figura 83: Subproceso de inserción de incidencias según el nivel 3 .....	166

Figura 84: Subproceso de resolución de incidencias según el nivel 3.....	166
Figura 85: Subproceso de inserción de incidencias según el nivel 4.....	167
Figura 86: Subproceso de resolución de incidencias según el nivel 4.....	168
Figura 87: Subproceso de solicitud de nuevo empleado según el nivel 0.....	171
Figura 88: Subproceso de valoración de solicitud según el nivel 0.....	171
Figura 89: Subproceso de valoración de proceso según el nivel 0.....	172
Figura 90: Subproceso de reclutamiento según el nivel 0.....	172
Figura 91: Subproceso de valoración de candidatos por RR.HH según el nivel 0.....	173
Figura 92: Subproceso de valoración de candidatos por C.I según el nivel 0.....	173
Figura 93: Subproceso de oferta de contratación según el nivel 0.....	174
Figura 94: Subproceso de solicitud de nuevo empleado según el nivel 1.....	175
Figura 95: Subproceso de valoración de solicitud según el nivel 1.....	175
Figura 96: Subproceso de valoración de proceso según el nivel 1.....	176
Figura 97: Subproceso de reclutamiento según el nivel 1.....	176
Figura 98: Subproceso de valoración de candidatos por RR.HH según el nivel 1.....	177
Figura 99: Subproceso de valoración de candidatos por C.I según el nivel 1.....	177
Figura 100: Subproceso de oferta de contratación según el nivel 1.....	178
Figura 101: Subproceso de solicitud de nuevo empleado según el nivel 2.....	179
Figura 102: Subproceso de valoración de solicitud según el nivel 2.....	179
Figura 103: Subproceso de valoración de solicitud según el nivel 2.....	180
Figura 104: Subproceso de reclutamiento según el nivel 2.....	180
Figura 105: Subproceso de valoración de candidatos por RR.HH. según el nivel 2.....	181
Figura 106: Subproceso de valoración de candidatos por C.I. según el nivel 2.....	181
Figura 107: Subproceso de oferta de contratación según el nivel 2.....	182
Figura 108: Subproceso de solicitud de nuevo empleado según el nivel 3.....	183
Figura 109: Subproceso de valoración de solicitud según el nivel 3.....	183
Figura 110: Subproceso de valoración de proceso según el nivel 3.....	184
Figura 111: Subproceso de reclutamiento según el nivel 3.....	185

Figura 112: Subproceso de valoración de candidatos por RR.HH según el nivel 3.....	185
Figura 113: Subproceso de valoración de candidatos por C.I según el nivel 3 .....	186
Figura 114: Subproceso de oferta de contratación según el nivel 3.....	186
Figura 115: Subproceso de solicitud de nuevo empleado según el nivel 4 .....	187
Figura 116: Subproceso de valoración de solicitud según el nivel 4.....	188
Figura 117: Subproceso de valoración de proceso según el nivel 4 .....	188
Figura 118: Subproceso de reclutamiento según el nivel 4 .....	189
Figura 119: Subproceso de valoración de candidatos por RR.HH según el nivel 4.....	189
Figura 120: Subproceso de valoración de candidatos por C.I según el nivel 4 .....	190
Figura 121: Subproceso de oferta de contratación según el nivel 4.....	190
Figura 122: Captura de pantalla del juego Risk.....	192
Figura 123: Captura de pantalla del juego Jetpack Joyride .....	192
Figura 124: Captura de pantalla del juego Temple Run.....	193
Figura 125: Captura de pantalla de un juego de la saga Metal Slug.....	194
Figura 126: Captura de pantalla del juego Puzzle Bobble.....	194
Figura 127: Captura de pantalla de Atriviate .....	195
Figura 128: Juego de estrategia por turnos según el nivel 0.....	196
Figura 129: Juego de estrategia por turnos según el nivel 1.....	197
Figura 130: Juego de estrategia por turnos según el nivel 2.....	198
Figura 131: Juego de estrategia por turnos según el nivel 3.....	199
Figura 132: Juego de estrategia por turnos según el nivel 4.....	200
Figura 133: Juego touch según el nivel 0 .....	201
Figura 134: Juego touch según el nivel 1 .....	202
Figura 135: Juego touch según el nivel 2 .....	203
Figura 136: Juego touch según el nivel 3 .....	204
Figura 137: Juego touch según el nivel 4.....	205
Figura 138: Juego de plataformas según el nivel 0 .....	206
Figura 139: Juego de plataformas según el nivel 1 .....	207

Figura 140: Juego de plataformas según el nivel 2 .....	208
Figura 141: Juego de plataformas según el nivel 3 .....	209
Figura 142: Juego de plataformas según el nivel 4 .....	210
Figura 143: Proceso de negocio de aplicación de catálogos según el nivel 0.....	212
Figura 144: Proceso de negocio de aplicación de catálogos según el nivel 2.....	213
Figura 145: Proceso de negocio de aplicación de catálogos según el nivel 3.....	214
Figura 146: Proceso de negocio de aplicación de catálogos según el nivel 4.....	215
Figura 147: Estructura de aplicación de BPLOM en un dominio específico .....	224
Figura 148: Comparación entre BPMN y BPLOM.....	225
Figura 149: Aplicación de la propuesta al dominio de los videojuegos.....	226
Figura 150: Interfaz gráfico de BPLLevel Modeler mostrando las zonas.....	227
Figura 151: Entorno de definición del interfaz .....	230
Figura 152: Definición de los elementos del menú.....	231
Figura 153: Definición de las zonas principales .....	233
Figura 154: Interfaz gráfico de BPLLevel Generator .....	235
Figura 155: BPLLevel Modeler mostrando el subproceso de inserción de incidencias.....	237
Figura 156: BPLLevel Modeler mostrando el proceso del videojuego de estrategia por turnos.....	240
Figura 157: Funcionamiento interno de BPLLevel Generator al generar videojuegos .....	241
Figura 158: Test de usabilidad sobre BPMN MUSIM .....	246
Figura 159: Gráfico de porcentaje sobre falta de símbolos en BPMN MUSIM .....	248
Figura 160: Gráfico de barras sobre comprensión de símbolos .....	248
Figura 161: Gráfico de porcentajes sobre la idoneidad de la ordenación de los niveles ...	251
Figura 162: Gráfico de barras sobre aumento de complejidad entre niveles .....	252
Figura 163: Gráfico de importancia de las entidades de la metodología BPLLOM.....	254
Figura 164: Dificultad media de las operaciones básicas en BPLLevel Modeler.....	258
Figura 165: Comparativa de parámetros monitorizados.....	260
Figura 166: Comparativa de errores y complejidad .....	260
Figura 167: Gráfico comparativo de repetición del modelado con BPLLevel Modeler .....	261

Figura 168: Coste medio del desarrollo de la aplicación web de inserción de incidencias (I)	265
Figura 169: Coste medio del desarrollo de la aplicación web de inserción de incidencias (II)	267
Figura 170: Coste medio del desarrollo de la aplicación web de resolución de incidencias (I)	268
Figura 171: Coste medio del desarrollo de la aplicación web de resolución de incidencias (II)	268
Figura 172: Coste medio del desarrollo de un videojuego de plataformas para iOS (I)	270
Figura 173: Coste medio del desarrollo de un videojuego de plataformas para iOS (II)	272
Figura 174: Coste medio del desarrollo de un videojuego touch para iOS (I)	273
Figura 175: Coste medio del desarrollo de un videojuego touch para iOS (II)	274



## ÍNDICE DE TABLAS

Tabla 1: Actividades y elementos relacionados con éstas incluidos en BPMN .....	56
Tabla 2: Compuertas de BPMN [BPMN].....	58
Tabla 3: Tipos de datos contemplados en BPMN.....	59
Tabla 4: Conversaciones de BPMN.....	61
Tabla 5: Entidades de BPMN MUSIM, nivel inicial de la metodología .....	142
Tabla 6: Entidades de la extensión de decisiones.....	143
Tabla 7: Entidades de la extensión de eventos .....	145
Tabla 8: Entidades de la extensión de actividades.....	146
Tabla 9: Entidades de la extensión de datos .....	147
Tabla 10: Elementos de menú y funcionalidad asociada.....	232
Tabla 11: Respuestas sobre falta de símbolos en BPMN MUSIM.....	247
Tabla 12: Respuestas sobre facilidad de comprensión de símbolos .....	247
Tabla 13: Idoneidad de la ordenación de los niveles de la metodología BPLOM.....	250
Tabla 14: Aumento de complejidad resultante del cambio de nivel .....	250
Tabla 15: Importancia de las entidades incluidas en la metodología BPLOM .....	253
Tabla 16: Dificultad de las operaciones básica del modelado.....	257
Tabla 17: Valores medios de los parámetros monitorizados en la comparativa.....	259
Tabla 18: Comparativa de repetición del modelado con BPLLevel Modeler.....	261
Tabla 19: Parámetros analizados en los modelos de proceso.....	262
Tabla 20: Parámetros analizados en el código.....	263
Tabla 21: Valores de los parámetros para los modelos de proceso de inserción de incidencias (I) .....	264
Tabla 22: Valores de los parámetros para el código del proceso de inserción de incidencias (I) .....	265
Tabla 23: Valores de los parámetros para los modelos de proceso de inserción de incidencias (II) .....	266
Tabla 24: Valores de los parámetros del modelo de proceso para el videojuego de plataformas (I) .....	269

*Índice de tablas*

Tabla 25: Valores de los parámetros de código para el videojuego de plataformas.....	270
Tabla 26: Valores de los parámetros del modelo de proceso para el videojuego de plataformas (II) .....	271
Tabla 27: Valores de los parámetros del modelo de proceso para el videojuego touch (I) .....	272
Tabla 28: Valores de los parámetros de código para el videojuego touch .....	273
Tabla 29: Valores de los parámetros del modelo de proceso para el videojuego touch (II) .....	274



---

---

# PARTE I

## PLANTEAMIENTO DEL PROBLEMA

---

---



# CAPÍTULO 1

## INTRODUCCIÓN

---

---

### *1.1 INTRODUCCIÓN*

Durante este capítulo se presenta la justificación y los objetivos de esta tesis doctoral.

Se comienza exponiendo las motivaciones de la investigación, donde se describen los problemas existentes en la situación actual, para posteriormente describir los requisitos generales y específicos que componen este trabajo. Para finalizar, se mencionarán las aportaciones y beneficios resultantes.

## 1.2 MOTIVACIONES

El proceso de desarrollo de software es una de las tareas que menos cambios ha experimentado en esta época de grandes cambios que acompaña a la informática de un tiempo a esta parte; es más, algunos de los problemas existentes en la actualidad ya fueron señalados en los años 70 por expertos de la materia y aún hoy siguen sin estar resueltos.

Uno de estos puntos conflictivos dentro del ciclo de desarrollo de software es la toma o captura de los requisitos, una fase crítica en el desarrollo de cualquier programa informático. La complejidad de este apartado radica en la necesidad de comprender a la perfección las explicaciones de un experto en el dominio de la aplicación para poder plasmar en nuestro sistema todos los detalles del funcionamiento del mismo. Así, la unión de la ambigüedad del lenguaje natural y del desconocimiento del dominio de la aplicación, tal y como establecen Kotonya y Sommerville en una de sus publicaciones [**Kotonya and Sommerville 1996**], provocan la existencia de errores en la captura de requisitos. Estos errores se propagarán durante el desarrollo de software y conllevarán continuos retrasos y modificaciones en el sistema, pudiéndose observar su efecto en el estudio de Lauesen y Vinter [**Lauesen and Vinter 2001**].

Una posible solución a este problema surgió de la mano del BPMI [**BPMI**] (Business Process Modeling Initiative), una iniciativa propuesta por el Object Management Group [**OMG**], también conocido por sus siglas OMG, que promovía el modelado de los procesos de un sistema por parte de los expertos de los mismos a través de la utilización de una notación de modelado que supusiera un lenguaje a medio camino entre los expertos informáticos y los del dominio. En esta línea, el BPMI propuso una notación estándar denominada BPMN (Business Process Modeling Notation) [**BPMN**] para este cometido. Sin embargo, el hecho de convertir esta notación en estándar supuso la necesidad de ofrecer soporte para una gran cantidad de escenarios de negocio posibles, lo que implicó el crecimiento en complejidad y extensión de la misma introduciendo la necesidad de otorgar formación a los expertos del dominio para que utilicen BPMN.

Algunos estudios actuales, como los llevados a cabo por Muehlen y Recker [**Muehlen and Recker 2008**] [**Recker 2008**], indican la baja utilización de algunos de los símbolos incluidos en BPMN y las dificultades de los usuarios a la hora de utilizarla. Esta circunstancia favoreció la aparición de notaciones simplificadas para el modelado de procesos de negocio. Entre ellas se encuentra una investigación realizada por otros compañeros de la Universidad de Oviedo donde se promovía una simplificación de BPMN denominada Simple BPMN (SBPMN) [**Fernández et al. 2009**] [**Fernández et al. 2010**]. SBPMN obtuvo unos buenos resultados de usabilidad en los test a los que fue sometida, existiendo una disminución de errores en el modelado de procesos con dicha notación. Sin embargo, los beneficios obtenidos a través de la simplificación de una notación producen una pérdida de expresividad de esta, siendo la expresividad una de las características más representativas e importantes de una notación gráfica de definición de procesos de negocio.

La ingeniería dirigida por modelos, también conocida por sus siglas en inglés MDE [**Kent 2002**], es un enfoque que permite la construcción de sistemas a partir de modelos que pueden ser comprendidos de forma más sencilla que la sintaxis de un lenguaje de

programación. Esta especificación de los sistemas a partir de modelos permite evitar la aparición de detalles relacionados con las distintas plataformas sobre las que se puede desplegar el sistema en cuestión. Para la conversión de estos modelos en sistemas informáticos MDE propone una serie de transformaciones que permiten convertir el modelo de entrada, independiente de la plataforma, en un sistema o programa informático concreto que ofrece soporte a las condiciones establecidas en el modelo construido en primer término. La ejecución de las transformaciones, realizada de manera automática por un equipo informático, permite acelerar el proceso de construcción de las soluciones software para el sistema que se ha modelado.

Motivados por este escenario, donde la captura de requisitos está expuesta a los problemas de desconocimiento del dominio y los expertos del dominio encuentran dificultades a la hora de utilizar las notaciones de modelado existentes por falta de conocimientos técnicos y de modelado, consideramos que es posible aunar los beneficios de las notaciones de modelado de procesos de negocio con dos de las características del enfoque propuesto por MDE: transformación automática de modelos a artefactos software y ausencia de detalles técnicos en los modelos.

A través de la solución que se propondrá en este documento, los expertos del dominio contarán una metodología de modelado que promueve la adaptación del lenguaje o notación de modelado tanto al nivel de conocimientos que poseen como al ámbito o dominio sobre el que trabajan, intentando mitigar algunos de los problemas existentes con los enfoques actuales. Además, la propuesta ofrecerá a los técnicos informáticos las siguientes posibilidades: simplificar el proceso de captura de requisitos, reduciendo la posibilidad de cometer errores durante esta fase del proceso de desarrollo; y acelerar el proceso de construcción de las aplicaciones informáticas que ofrezcan soporte a los procesos de negocio modelados por los expertos, haciendo uso del sistema de transformaciones de modelos propuesto por MDE.

### **1.3 HIPÓTESIS**

Es posible definir una metodología de modelado que facilite la implicación de los expertos del dominio en el modelado de los procesos de negocio, independientemente de su nivel de conocimiento técnico, y que simplifique y automatice, utilizando los modelos de proceso resultantes, la generación de artefactos que ofrezcan soporte a los sistemas representados por dichos modelos de proceso.

## 1.4 OBJETIVOS DE LA TESIS

Debido a la situación descrita con anterioridad, se plantea como trabajo de investigación para esta tesis la especificación de una metodología de modelado que se adapte tanto al nivel de conocimiento del experto de negocio que la utiliza como al dominio sobre el que se esté aplicando. La especificación de dicha metodología se verá cumplimentada con la creación de dos aplicaciones compatibles: una herramienta de modelado de procesos de negocio que permita definir el modelo de proceso para una aplicación y una herramienta de generación de aplicaciones que tome como elemento de entrada un modelo de proceso realizado con la herramienta de modelado que ofrece soporte a la metodología propuesta y lo convierta, de forma automática, en una aplicación informática cuyo comportamiento se corresponda con el modelo de proceso diseñado. Es necesario señalar que las aplicaciones que se generen a partir de la propuesta descrita anteriormente deberán poder utilizarse “out of the box”, reduciendo al máximo posible el proceso de configuración e/o instalación de las mismas.

Tanto la metodología de modelado de procesos de negocio como las dos aplicaciones con soporte para el uso de la misma deberán construirse de tal forma que permitan, en un tiempo futuro, agregar los componentes y mecanismos necesarios para dar soporte a nuevas tecnologías o lenguajes informáticos.

Este planteamiento se divide en los siguientes objetivos de carácter general y específico.

### 1.4.1 Objetivos generales

Para el desarrollo de esta tesis, se plantean los siguientes objetivos de carácter general:

- Definir una metodología de modelado por niveles que adapte su grado de expresividad y complejidad al grado de conocimientos que tiene el experto del dominio que la utilice.
- Garantizar la capacidad de adaptación de la metodología a dominios de distinta naturaleza.
- Ofrecer los mecanismos de soporte necesarios para favorecer la utilización de la metodología de modelado por parte de los expertos del dominio.
- Automatizar la construcción de aplicaciones informáticas a partir de los modelos de proceso definidos con la metodología. Es necesario evitar la aparición de detalles de carácter técnico en la especificación de los modelos, de tal manera que el experto de dominio experimente un nivel de abstracción elevado e interactúe únicamente con detalles y características que conoce.

### **1.4.2 Objetivos específicos**

Los objetivos generales expuestos anteriormente tienen correspondencia con los siguientes objetivos específicos:

- Realizar un análisis global de los lenguajes y notaciones de modelado de procesos de negocio existentes en la actualidad, estudiando sus características e identificando sus puntos fuertes y débiles.
- Realizar un estudio de las plataformas y aplicaciones con soporte para la utilización de modelos de proceso, haciendo distinción entre aquellas que solamente permiten modelar los procesos y otras que permiten modelar y ejecutar los procesos.
- Realizar un recorrido por los géneros de aplicaciones más utilizados en la actualidad, intentando establecer cuáles serían apropiados para la introducción de la propuesta y si en estos géneros existe alguna solución actual similar a la que se propone.
- Establecer los niveles de la metodología que permitan la adaptación de ésta al nivel de conocimiento del experto del dominio.
- Crear los prototipos software que permitan modelar los procesos de negocio y generar aplicaciones multigénero y multiplataforma con soporte para los procesos de negocio utilizando la metodología propuesta.
- Utilizar la metodología y los prototipos para modelar un conjunto de aplicaciones de distintas características, mostrando el grado de adaptación de la propuesta a los distintos ámbitos de aplicación y generando el código que ofrezca soporte a la funcionalidad descrita en los modelos de proceso.
- Comparar las aplicaciones modeladas y generadas utilizando la metodología propuesta con otras aplicaciones de similares funcionalidades desarrolladas con otras tecnologías, con el fin de analizar el coste asociado a la creación de las mismas en base a una serie de parámetros.
- Validar que la metodología de modelado propuesta se ajuste a los objetivos marcados en la investigación.

### **1.4.3 Aportaciones y Beneficios**

La propuesta recogida en este trabajo de investigación permite involucrar de una manera directa a los expertos del dominio en el proceso de captura de requisitos, la primera tarea dentro del desarrollo de software. Este escenario permite reducir algunos de los



problemas o inconvenientes resultantes del enfoque clásico de captura de requisitos: la ambigüedad del lenguaje natural, el desconocimiento del dominio del problema, etc. Esto permite minimizar la aparición de errores derivados de los inconvenientes mencionados anteriormente, y por tanto actuar sobre el punto de aparición de errores más costoso de todo el proceso de desarrollo de software **[Kotonya and Sommerville 1996]**.

El esquema de funcionamiento de BPM que, tal y como se explica en el [capítulo 3](#) de este documento, se divide normalmente en cinco fases, sufre una ligera variación con la propuesta que se presenta en este trabajo de investigación. Durante la segunda fase de dicho esquema, se producen una serie de reuniones en las que el analista recibe, en forma de documentación escrita, una definición de las características del proceso en cuestión y una serie de diagramas de alto nivel que lo representan. Las reuniones tienen como objetivo comprobar que el analista ha sido capaz de entender la descripción del sistema a la perfección, ya que éste será el encargado de modelar el sistema en cuestión utilizando la notación y la herramienta adecuadas. Sin embargo, el enfoque que se desprende de la propuesta elimina la necesidad de contar con la participación del analista, ya que serán los expertos del dominio los que modelen el proceso de forma completa utilizando los niveles de la metodología que se adapten a su nivel de conocimiento.

Otra de las aportaciones que se desprende de la propuesta que se realiza en esta investigación es una mejora de la productividad a la hora de construir las aplicaciones que ofrecen soporte a los sistemas que se modelan. El desarrollo de software tradicional supone un coste temporal elevado, ya que una vez que los modelos han sido realizados es necesario que estos sean implementados por parte de los desarrolladores. El esquema de funcionamiento de BPM propone la traducción a código de los modelos realizados por los analistas, tarea que puede ser realizada de forma más o menos automática por una herramienta. En el caso de esta investigación, uno de los prototipos que acompaña a la metodología de modelado que compone el núcleo central de la propuesta que aquí se realiza permite la traducción del modelo a código y la generación de una aplicación específica con soporte para dicho modelo.

## 1.5 ORGANIZACIÓN DEL DOCUMENTO

La memoria de esta tesis está estructurada en secciones y capítulos para facilitar su lectura. En este apartado se presenta de forma esquemática la organización de este documento.

La **Parte I “Planteamiento del problema”** incluye los capítulos que sirven de introducción a este trabajo de investigación, en los que se presentan los problemas, la hipótesis y el contexto. En el **capítulo 1, “Introducción”**, se describen la justificación, motivación y planteamiento del problema; además, se establece la hipótesis y se marcan los objetivos de la investigación, dando por terminado el capítulo con una descripción de la organización del documento. El **capítulo 2**, denominado **“Metodología y desarrollo de la investigación”**, se presenta la metodología seguida para el desarrollo de esta tesis.

En la **Parte II “Conceptos, soluciones y tecnologías existentes”** se describen los conceptos y tecnologías que sirven de fundamento para la realización de esta tesis doctoral. El **capítulo 3**, cuyo nombre es **“Business Process Modeling”**, sirve para presentar el modelado de procesos de negocio. Para ello se realiza un recorrido por sus características más significativas y se presentan algunas de las notaciones más utilizadas, señalando tanto sus puntos fuertes como sus puntos débiles. El **capítulo 4, “Herramientas con soporte para Business Process Modeling”**, realiza un recorrido por los dos tipos de herramientas y plataformas que ofrecen soporte a este enfoque: unas solo para el modelado de los procesos y otras para el modelado y ejecución de los procesos. El **capítulo 5**, denominado **“MDE: Ingeniería dirigida por modelos”**, realiza un recorrido por las características de este enfoque utilizado para el desarrollo de software a partir de modelos y transformaciones entre modelos.

En la **Parte III “Desarrollo de la propuesta”** se presenta el enfoque propuesto para la solución de los problemas mencionados en la primera parte del capítulo primero. El **capítulo 6, “Metodología de modelado por niveles para BPM”**, actúa a modo de presentación de la propuesta que se realiza en esta tesis: una metodología de modelado que se adapta al conocimiento del usuario y que puede utilizarse dentro de cualquier dominio. Dentro de esta misma parte se encuentra el **capítulo 7**, llamado **“Caso de estudio”**. El contenido de este capítulo trata sobre la aplicación de la propuesta al modelado de diversos tipos de procesos de negocio, haciendo un recorrido explicativo por los cambios que se producen dentro de los procesos con el paso de estos a través de cada nivel de la metodología propuesta.

La **Parte IV “Desarrollo de prototipos y evaluación”**, está integrada por la presentación de los prototipos que acompañan a la propuesta. El **capítulo 8, “Desarrollo de prototipos”**, incluye la presentación de las herramientas que acompañan a la metodología de modelado propuesta: una herramienta para el modelado de procesos de negocio y una herramienta que genera aplicaciones informáticas en base a los modelos que se le pasan. En el **capítulo 9, “Evaluación”**, se presentan los resultados obtenidos a partir de la evaluación a la que fue sometida la propuesta.

Por último, en la **Parte V “Conclusiones”** se incluyen las conclusiones que pueden extraerse del trabajo realizado durante esta investigación. El **capítulo 10, “Conclusiones”**

**y trabajo futuro**", presenta las conclusiones a través de: un resumen del trabajo realizado, la verificación del cumplimiento de los objetivos enumerados al principio de la investigación y una mención a las publicaciones más relevantes. Para finalizar este capítulo se incluyen las líneas de trabajo futuro que se derivan de la propuesta realizada.

El último fragmento de este documento se destina a la inclusión de la bibliografía y las referencias consultadas para la realización de las distintas partes de esta tesis doctoral.



# CAPÍTULO 2

## METODOLOGÍA Y DESARROLLO DE LA INVESTIGACIÓN

---

### *2.1 INTRODUCCIÓN*

En este capítulo se presenta la metodología utilizada para el desarrollo de esta tesis doctoral. Además, se incluye un pequeño resumen cronológico de la investigación donde se destacan los momentos más relevantes.

## **2.2 METODOLOGÍA DE TRABAJO**

Este trabajo de investigación ha sido realizado en varias fases, utilizando un enfoque iterativo e incremental. La figura 2.1, incluida a continuación del texto, muestra un pequeño resumen gráfico de la investigación, incluyendo tanto detalles del trabajo como referencias temporales.

El inicio de la investigación está marcado por el arranque del trabajo fin de master realizado para obtener la titulación del Máster en Ingeniería Web por la rama de investigación, realizado en la Escuela de Ingeniería Informática de la Universidad de Oviedo. Utilizando los conocimientos adquiridos durante los cursos académicos del máster (años 2009/2010 y 2010/2011) se realizó una primera propuesta donde se definía el nivel inicial de una metodología de modelado orientada a los expertos del dominio sin conocimientos técnicos o informáticos. Esta propuesta fue el centro de unas pruebas de uso para un caso real, donde se contó con la colaboración de la empresa asturiana Isastur **[Isastur]** para su utilización como lenguaje para el modelado de dos procesos de negocio existentes en la compañía. Los resultados obtenidos en las pruebas de usabilidad realizadas así como los comentarios realizados por los expertos del dominio que utilizaron la propuesta sirvieron de punto de partida para la expansión de la investigación. Además, los datos incluidos en otros trabajos relacionados con esta propuesta sirvieron de motivación y guía.

Una vez llegados a este punto se comenzó el trabajo de investigación, que consistió en realizar un estudio del estado del arte sobre los problemas actuales de las notaciones de modelado y sobre la utilización de modelos para definir el funcionamiento de las aplicaciones informáticas. A partir de este estado del arte y de las conclusiones que se extrajeron del mismo comenzó el desarrollo de esta tesis doctoral, comenzando por la identificación del problema a resolver y los objetivos e hipótesis de partida.

Para llevar a cabo la investigación se ha optado por un enfoque iterativo e incremental. Así en cada interacción, se desarrollaron prototipos y se modificó el modelo propuesto hasta llegar al finalmente expuesto. Se han realizado varias publicaciones sobre esta investigación con el objetivo de obtener retroalimentación por parte de otros investigadores, buscando obtener opiniones que permitan llevar el trabajo por el camino adecuado.

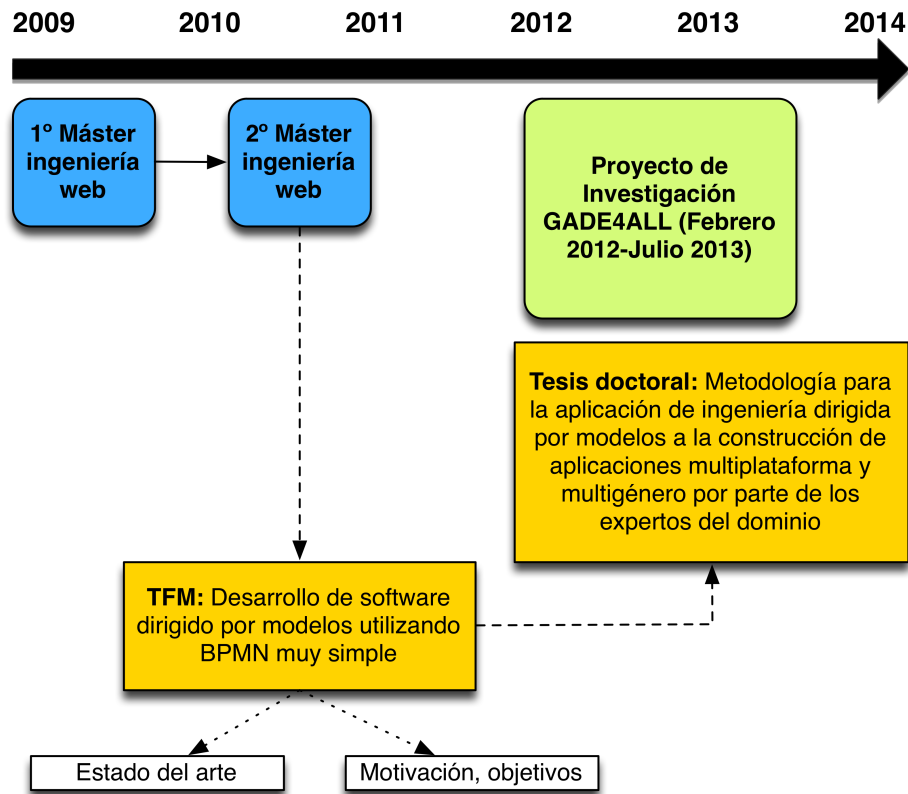


Figura 1: Metodología de investigación utilizada

### 2.3 DESARROLLO TEMPORAL DE LA INVESTIGACIÓN

A continuación se describen de forma detallada algunos de los aspectos de esta investigación en orden cronológico.

- **Septiembre 2009.** Comienzo del primer curso del Máster en Ingeniería Web. Durante el primer año se cursan una serie de asignaturas que permiten asentar y mejorar conocimientos técnicos adquiridos durante la realización de la Ingeniería Técnica en Informática de Sistemas, que serán de gran utilidad para completar el desarrollo de los prototipos que acompañan este trabajo.
- **Septiembre 2010.** Comienzo del segundo curso del Máster en Ingeniería Web. Durante el segundo año del máster se cursan las asignaturas de la rama de investigación: diseño y construcción de MDA, desarrollo e integración de aplicaciones en Internet, etc.
- **Diciembre 2010.** Comienzo del trabajo fin de máster. El objetivo es analizar plataformas con soporte para el modelado de procesos de negocio y realizar una propuesta de notación muy simplificada de modelado que pueda ser usada por personal no técnico para definir sus procesos de negocio y generar aplicaciones web a partir de los procesos que hayan definido.
- **Febrero 2011.** Se envían los artículos “BPMN MUSIM: Notación BPMN MUy Simplificada” e “Isastur Modeler: A tool for BPMN MUSIM” para su revisión y posible inclusión en la 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información. El primero de los artículos se envía al apartado del simposio doctoral mientras que el segundo se envía como *full paper*.
- **Marzo 2011.** Se recibe confirmación de la aceptación, en la 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información, de los dos artículos anteriormente mencionados.
- **Junio 2011.** Participación en la 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información que tuvo lugar entre los días 15 y 18 de Junio en Chaves (Portugal). Se acude para exponer y defender el trabajo presentado en los dos artículos aceptados.
- **Junio 2011.** Envío a revisión del artículo “BPMN MUSIM: Very simple BPMN for Application Domain Experts” a la revista Computer Standards & Interfaces de la editorial Elsevier, **con un factor de impacto en el índice JCR de 0.825 en el año 2010**. Este artículo se estructuró como un recorrido por los problemas e inconvenientes de las notaciones y lenguajes de modelado existentes para acto seguido proceder a la presentación del nivel inicial de la metodología de modelado, una notación de modelado muy simplificada. Dicha presentación se acompañó de la descripción de los prototipos que ofrecían soporte a la notación presentada y de los resultados obtenidos a partir de la aplicación de la propuesta al caso de uso real existente en la empresa asturiana Isastur, que cuenta con delegaciones en varios países del mundo.
- **Julio 2011.** Lectura del trabajo fin de máster en la modalidad de investigación “Desarrollo de software dirigido por modelos utilizando BPMN muy simple” con calificación de Sobresaliente. Este trabajo de investigación contiene el estado del arte, la definición de los objetivos y la especificación del nivel inicial de la metodología de modelado. Además se presenta la primera versión de dos



prototipos funcionales, uno para el modelado de los procesos de negocio por parte de los expertos del dominio y otro para la generación de aplicaciones web por parte de los expertos informáticos.

- **Diciembre 2011.** Participación en las primeras jornadas doctorales de la Universidad de Oviedo con el póster titulado “Metodología por pasos para la aplicación de BPMN 2.0”. El póster incluye un resumen de los objetivos de la investigación, incluyendo una introducción a la metodología de modelado y un ejemplo sobre el uso del nivel inicial de la metodología para la generación de aplicaciones web.
- **Febrero 2012.** Comienzo del proyecto de investigación “Gade4all: Plataforma genérica para facilitar el desarrollo de videojuegos y software de entretenimiento multiplataforma”, financiado por el Ministerio de Industria, Energía y Turismo, por el plan Avanza2 y por Fondo Europeo de Desarrollo Regional; código del proyecto: MITC-11-TSI-090302-2011-11. La función principal desarrollada durante el proyecto es la de desarrollador de software para las plataformas iOS, Android y HTML5. Durante la realización de este proyecto se realiza un estudio acerca de la importancia de los videojuegos en el mercado actual y cómo se podría aplicar la metodología de modelado por niveles para la descripción de la lógica de un videojuego. La duración del proyecto es de 18 meses.
- **Octubre 2012.** Realización del curso “Habilidades para la difusión y la comunicación de la investigación” encuadrado dentro del programa de formación transversal de doctorado organizado por la Universidad de Oviedo. El curso trata sobre la forma de elaborar y presentar tanto artículos como pósteres y comunicaciones científicas en inglés.
- **Noviembre 2012.** Participación en las segundas jornadas doctorales con el póster titulado “Generación automática de aplicaciones a partir de modelos de procesos de negocio”. El poster incluye detalles de la metodología de modelado y también un resumen gráfico del proceso que se sigue para la generación de las aplicaciones informáticas a partir de los modelos de proceso.
- **Febrero 2013.** Realización del curso “Habilidades para una inserción profesional de calidad” encuadrado dentro del plan de formación transversal de doctorado de la Universidad de Oviedo. Durante el curso se abordan contenidos importantes tanto para la realización de un trabajo de investigación como para el mundo laboral, entre los que destacan: trabajo en equipo, organización, gestión del tiempo y gestión del cambio organizativo.
- **Mayo 2013.** Envío a revisión del artículo “BPLOM: BPM Level-Oriented Methodology for Incremental Business Process Modeling and Code Generation on Mobile Platforms” a la revista International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI). El artículo incluye la presentación de todos los niveles de la metodología de modelado y su aplicación para la construcción de aplicaciones de catálogo para dispositivos móviles.
- **Mayo 2013.** Realización del curso “Orientación para el empleo” encuadrado dentro del plan de formación transversal de doctorado de la Universidad de Oviedo. El curso incluye formación acerca del proceso de solicitud de patentes así como de la creación de empresas y la obtención de financiación a partir de los trabajos de investigación. Con este curso se da por completado el plan de formación transversal de doctorado del año 2013.

- **Junio 2013.** Publicación del artículo “BPLOM: BPM Level-Oriented Methodology for Incremental Business Process Modeling and Code Generation on Mobile Platforms” en el volumen 2 nº2 de la revista *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*.
- **Julio 2013.** Publicación del cuaderno de investigación titulado “Metodología para la aplicación de BPM a procesos de negocio por parte de los expertos del dominio” en la editorial Servitec. ISBN: 978-84-695-8369-3.
- **Agosto 2013.** El artículo “BPMN MUSIM: Approach to improve the domain expert’s efficiency in business processes modeling for the generation of specific software applications” es aceptado para publicación en la revista *Expert Systems With Applications* de la editorial Elsevier, **con un factor de impacto en el índice JCR de 1.854 en el año 2012**. El artículo es una revisión y mejora del enviado a revisión al finalizar el trabajo fin de máster, incluyendo mejoras tanto en el estado del arte como en la evaluación de la propuesta.
- **Septiembre 2013.** Envío a revisión del artículo “Using Business Process Modeling for Videogame Modeling and Code Generation in Multiple Platforms” a la revista *Computer Standards & Interfaces* de la editorial Elsevier, **con un factor de impacto en el índice JCR de 0.978 en el año 2012**. El artículo presenta la aplicación de la metodología de modelado por niveles a la construcción de videojuegos para dispositivos móviles.
- **Octubre 2013 a Abril 2014.** Trabajo en la confección de la memoria que recoge el trabajo realizado durante esta investigación.

---

---

## PARTE II

# CONCEPTOS, SOLUCIONES Y TECNOLOGÍAS EXISTENTES

---

---



# CAPÍTULO 3

## BUSINESS PROCESS MODELING

---

---

### 3.1 INTRODUCCIÓN

BPM es el conjunto de técnicas que permiten realizar el modelado de los procesos para cualquier área de negocio, estableciendo mecanismos para el modelado, la gestión, la optimización y la ejecución de estos procesos. Para ofrecer soporte a estas técnicas BPM se apoya en las notaciones de modelado de procesos de negocio, que son lenguajes de modelado que se encuentran a medio camino entre las notaciones utilizadas por los expertos informáticos y las entidades que manejan los expertos de los dominios de negocio.

Debido a la utilización de estas notaciones, BPM logra involucrar de manera activa a los expertos del dominio, consiguiendo que sean los propios expertos quienes intenten modelar sus procesos en vez de optar por una captura de requisitos tradicional. Así, los procesos se encontrarán bien definidos desde el primer momento, evitando algunos de los problemas conocidos [**Kotonya and Sommerville 1996**] del proceso de captura de requisitos:

- La persona encargada del proceso de captura de requisitos desconoce el dominio sobre el que realiza esta tarea.
- La ambigüedad del lenguaje natural lleva a interpretaciones erróneas, idea que se ve reforzada por el estudio realizado por Laue y Gadatsch [**Laue and Gadatsch 2010**] donde se establece que la formulación de las preguntas sobre un proceso concreto influye de forma clara en la corrección de las respuestas dadas.

En este capítulo se realizará una exposición concisa de las fases en las que se aplica este concepto y de los beneficios que se pueden obtener. Además, se revisarán las principales notaciones o lenguajes de BPM según una serie de factores que se consideran importantes.

### **3.2 ESQUEMA DE FUNCIONAMIENTO DE BPM**

Según un el trabajo de Ryan K. L. Ko [Ko 2009], el esquema de funcionamiento de BPM se puede contemplar como un conjunto de cinco fases claramente diferenciadas que permiten construir un sistema de ejecución de modelos.

Durante la primera fase se produce la identificación de las necesidades del proceso de negocio en cuestión, delimitando su alcance y estableciendo todas las características que han de estar presentes en el mismo. Esta identificación la realiza el experto del dominio del problema.

Una vez realizada esta identificación se produce la definición de objetivos del proceso presentando una definición de requisitos junto con unos diagramas de alto nivel en los que se establecen las partes fundamentales del proceso, siendo toda esta documentación entregada al analista del negocio. Habitualmente esta segunda fase se realiza de forma manual y comprende el establecimiento de reuniones entre las partes para comprobar que el analista ha comprendido completamente todos los requisitos del proceso.

Al término de estas dos primeras fases, el analista del negocio está capacitado para modelar el proceso con una notación gráfica que pueda ser interpretada de forma sencilla. En esta fase se introducen detalles relacionados con las características operativas de la compañía como restricciones y condiciones especiales de los procesos. El siguiente paso será la traducción de los diagramas generados a código, realizada por el analista de negocio a través de alguna herramienta creada por los desarrolladores de software y que ofrece soporte para estas tareas.

La siguiente parte del proceso será la revisión del código por parte de un componente del equipo informático de la organización. En este momento se introducirán el resto de detalles lógicos que sean necesarios y también se realizarán los ajustes convenientes. El prototipo surgido será mostrado a los responsables y con el visto bueno dado por estos se producirá la publicación de la aplicación, que será el último paso que componga el esquema.

### 3.3 BENEFICIOS DERIVADOS DE LA APLICACIÓN BPM

La aplicación de las técnicas propuestas en BPM conlleva la existencia de una serie de beneficios que son explicados por Havey en su libro [Havey 2005]. Algunos de los beneficios más relevantes son los siguientes:

- **Formalización de los procesos de negocio existentes**, lo que puede producir la introducción de mejoras en los mismos. En muchos escenarios reales los procesos de negocio son conocidos en detalle sólo por sus responsables lo que dificulta su seguimiento; en este punto, la adopción de BPM conlleva el modelaje de los procesos de tal forma que es necesario repasarlos y entenderlos, dejando la puerta abierta a posibles modificaciones que permitan mejorarlos.
- **Automatización de los procesos**. Como cada proceso dispone de más de una actividad, la reducción del tiempo entre actividades supone una mejora a nivel organizativo. La ejecución de los procesos a través de las características propuestas por BPM permite que el tiempo de espera entre actividades sea el menor posible y por lo tanto el proceso se realiza de forma más rápida.
- **Incremento de la productividad con menor número de personal**. Según algunos de los ejemplos de aplicación de BPM en escenarios reales, las organizaciones consiguen reducir el número de personas necesarias para la realización de estas tareas. Ello implica que el tiempo de ejecución de sus procesos disminuye y aumenta la satisfacción de los usuarios.

### **3.4 NOTACIONES DE MODELADO RELACIONADAS CON BPM**

Una vez presentado el concepto de modelado de procesos de negocio se realizará un recorrido por algunas de las notaciones existentes. Es necesario señalar que existen otras aparte de las que aquí se presentan, aunque las incluidas en este documento han sido escogidas en base a las siguientes características:

- **Grado de difusión**, contemplando aquellas notaciones que disponen de un mayor índice de utilización.
- **Nivel de complejidad**, representando un rango amplio de opciones. En este recorrido se parte del ejemplo más complejo (representado por BPMN) hasta llegar a uno de los más sencillos (representado por SBPMN).
- **Implicación de los expertos**. Según Lu y Sadiq en su estudio [Lu and Sadiq 2007] las notaciones gráficas permiten representar la mayor parte de los procesos con una sintaxis más abstracta y una semántica más simple, lo que reduce la complejidad de representación y verificación del modelo. Además, en ese mismo estudio se establece que si bien las notaciones textuales son más poderosas y flexibles, su utilización requiere un grado de experiencia de carácter técnico que no se puede presuponer en el caso de expertos del dominio. Por lo tanto, como una parte fundamental de este trabajo es la implicación de los expertos, el conjunto de notaciones estudiadas se reduce a las notaciones de carácter gráfico.

#### **3.4.1 BPMN: notación estándar de modelado de procesos de negocio**

BPMN es la notación estándar propuesta por el OMG para el modelado, gestión y ejecución de procesos de negocio.

##### *3.4.1.1 Historia de BPMN*

El estándar BPMN surge en el año 2004 a partir del trabajo de un grupo surgido dentro del OMG [OMG] con el nombre de BPMI [BPMI]. Ante la situación existente por aquel entonces, este grupo del OMG propone la creación de una notación estándar de modelado bajo los siguientes objetivos: en primer lugar dotar a los expertos de los distintos dominios de una notación que puedan comprender y utilizar de forma inmediata y por otro lado intentar terminar con la existencia de una gran variedad de notaciones y herramientas de modelado, homogeneizando el ambiente del modelado de procesos existente en aquel entonces [White 2004].

Desde el año de su creación, BPMN ha ido sufriendo constantes cambios y evoluciones, hasta el punto de haberse publicado recientemente su versión 2.0 [BPMN]. Según datos ofrecidos por el OMG, BPMN es la notación de modelado de procesos de negocio más utilizada en la actualidad; en concreto se tienen datos de la existencia de más de 70 implementaciones de la misma (datos obtenidos a finales del año 2013), número que continúa aumentando [BPMN].



### 3.4.1.2 Elementos incluidos en BPMN


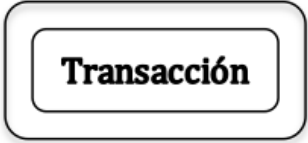




Debido a su carácter de estándar, BPMN ofrece una extensa colección de elementos repartidos en distintos grupos. Según las entidades ofrecidas por cada uno de los conjuntos existentes, se puede considerar que los elementos encuadrados dentro de los grupos actividades, compuertas y datos son el núcleo de BPMN mientras que las entidades pertenecientes a los demás conjuntos ofrecen las funcionalidades avanzadas.









A continuación se hará un pequeño recorrido por algunos de los conjuntos existentes en BPMN para conocer algunas de las entidades que están disponibles para el modelado de los procesos; las imágenes utilizadas han sido extraídas de la documentación existente sobre BPMN 2.0, que puede encontrarse en la web **[BPMN]**.

#### 3.4.1.2.1 Actividades

Como en cualquier notación de modelado de procesos de negocio, las actividades son la parte principal de BPMN. Estas entidades son las encargadas de representar las acciones que han de realizarse en un proceso desde que se inicia hasta que se concluye, de forma que la correlación de las mismas se corresponda con el orden requerido para alcanzar el cometido del proceso.

La tabla que se muestra a continuación se corresponde con los elementos que ofrece BPMN en el apartado de actividades. Tal y como se puede observar el conjunto ofrece, además de las actividades propiamente dichas, otra serie de elementos: marcadores para las actividades, tipos de tareas y tipos de flujos o conectores.

Nombre	Representación	Explicación
<b>Actividades</b>		
Tarea		Una tarea es una unidad de trabajo. Cuando se marca con el símbolo de subproceso se indica que puede ser refinada.
Transacción		Es un conjunto de actividades relacionadas entre si.
Subproceso de Evento		Situado en el interior de un (sub)proceso, se activa cuando ocurre el evento de inicio y se mantiene activo mientras el (sub)proceso esté activo también. Tiene la facultad de poder interrumpir al (sub)proceso que lo contiene.
Actividad de Llamada		Es una referencia a un subproceso o tarea definido de forma global y reutilizado en el proceso actual.
<b>Marcadores</b>		
Subproceso		Todos los marcadores tienen como función especificar el comportamiento particular de cada actividad durante la ejecución del modelo.
Ciclo		

Instancias múltiples en paralelo		
Instancias múltiples en secuencia		
Ad Hoc		
Compensación		
<b>Tipos de Tareas</b>		
Envío		Los tipos de las tareas sirven para marcar la naturaleza de la tarea que se va a realizar.
Recepción		
Tarea de Usuario		
Tarea Manual		







Regla de Negocio		
Invocación de Servicio		
Ejecución de Script		
<b>Flujos</b>		
Secuencia		Define el orden de ejecución entre dos actividades.
Por Defecto		Camino a seguir si las condiciones de los caminos alternativos evalúan a falso.
Condicional		Posee una condición asociada que permite saber si el camino será activado o no.

Tabla 1: Actividades y elementos relacionados con éstas incluidos en BPMN

### 3.4.1.2.2 Eventos

Los eventos son otro de los grupos de entidades existentes en BPMN. En este caso a partir del uso de estos símbolos se puede manifestar la aparición de un evento en un proceso. Los eventos están divididos en un serie de grupos entre los que se encuentran los siguientes: mensaje, condicional, error y temporal.

Además de esta división en grupos, tal y como se observa en la siguiente imagen también existe una división de los grupos en momentos donde aplicar los eventos; en este caso se

hace distinción entre eventos para el inicio de los procesos, para puntos intermedios y para el final de los procesos.

Eventos	Inicio			Intermedios			Fin
	Alto Nivel	Evento Interruptor de Subproceso	Evento No Interruptor de Subproceso	Captura	Adjunto Interruptor	Adjunto No Interruptor	Lanzamiento
<b>Simple:</b> Eventos sin especificar. Indican puntos de inicio, de fin y situaciones intermedias.							
<b>Mensaje:</b> Recepción y envío de mensajes.							
<b>Temporal:</b> Puntos en el tiempo, lapsos, límites (timeouts). Pueden ser eventos únicos o cíclicos.							
<b>Escalable:</b> Cambio a un nivel mas alto de responsabilidad.							
<b>Condicional:</b> Reacción a cambios en las condiciones de negocios o integración de reglas de negocio.							
<b>Enlace:</b> Conectores: fuera de página. Dos conectores de enlace equivalen a un flujo de secuencia.							
<b>Error:</b> Captura y lanzamiento de errores conocidos con nombre.							
<b>Cancelación:</b> Reacción a la cancelación de una transacción/ Solicitud de cancelación.							
<b>Compensación:</b> Manejo/ Solicitud de compensación.							
<b>Señal:</b> Intercambio de señales entre procesos. Una señal puede ser capturada varias veces.							
<b>Multiple:</b> Captura uno de un conjunto de eventos. Lanza todos los eventos definidos.							
<b>Paralela Multiple:</b> Captura todos los eventos de un conjunto de eventos en paralelo.							
<b>Terminación:</b> Terminación inmediata del proceso.							

Figura 2: Eventos incluidos en BPMN [BPMN]

### 3.4.1.2.3 Compuertas

El grupo denominado compuertas está compuesto por aquellas construcciones que permiten la toma de alternativas dentro del flujo de un proceso. La tabla incluida a continuación muestra los siete tipos de compuertas que existen en BPMN.








Nombre	Representación	Explicación
Exclusiva		<p>En un punto de bifurcación selecciona únicamente un flujo de secuencia de entre las alternativas. En la salida la compuerta espera a que el flujo de entrada termine para activar la salida.</p>
Basada en eventos		<p>Siempre está seguida por eventos o tareas de recepción y activará solamente un flujo saliente en respuesta al evento que ocurra en primer lugar.</p>
Paralela		<p>Todos los caminos de salida se activan simultáneamente. En el punto de convergencia se espera a la finalización de todos los caminos entrantes para activar la salida.</p>
Inclusiva		<p>A la llegada de esta compuerta se activa al menos un flujo de entrada. En la salida se espera a que finalicen todos los flujos activados en la entrada.</p>
Exclusiva basada en eventos		<p>Cuando ocurre uno de los eventos subsecuentes se crea una nueva instancia del proceso.</p>
Compleja		<p>Comportamiento complejo no capturado por el resto de compuertas.</p>
Paralela basada en eventos		<p>Cuando ocurren todos los eventos subsecuentes se crea una nueva instancia del proceso.</p>

Tabla 2: Compuertas de BPMN [BPMN]

3.4.1.2.4 Datos

Las entidades contenidas dentro del grupo de datos permiten especificar la participación de datos de entrada y salida en las tareas así como la utilización de documentos o las actividades de lectura y escritura de datos que se pueden dar en un proceso. La tabla que se incluye a continuación muestra los elementos de este grupo.

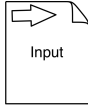
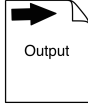



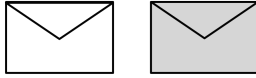
Nombre	Representación	Explicación
Input		Un dato de entrada o <i>input</i> es una entrada externa a todo el proceso. Puede ser leído por una actividad.
Output		Un dato de salida o <i>output</i> es una variable disponible como resultado del proceso.
Dato de tipo objeto		Un dato de tipo objeto representa información que fluye a través del proceso. Por ejemplo: documentos, e-mails o cartas.
Colección de objetos de datos		Una colección de objetos de datos representa una colección de información, p.e. una lista de artículos.
Data Store		Un almacén o <i>data store</i> es un lugar donde el proceso puede leer o escribir datos, p.e. una base de datos. La información en un almacén persiste más allá de la vida de la instancia del proceso.
Mensaje		Un mensaje es utilizado para representar el contenido de una comunicación entre dos participantes.

Tabla 3: Tipos de datos contemplados en BPMN

### 3.4.1.2.5 Contenedores

Los contenedores, conocidos como *pools* en inglés, de BPMN son los encargados de acoger a todos los tipos de entidades que componen la notación estándar de modelado de procesos de negocio. Dentro de BPMN los contenedores pueden dividirse en compartimentos, denominados *lanes*. Estos contenedores representan a las entidades responsables de las actividades existentes en un proceso; estas entidades responsables pueden ser organizaciones, roles o sistemas. Es necesario señalar que los compartimentos pueden anidarse en compartimentos y contenedores.

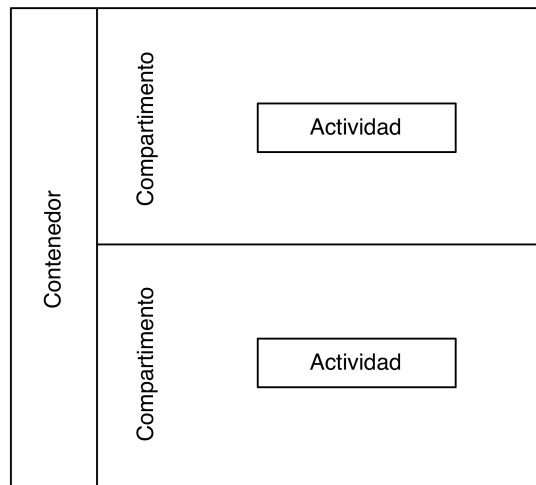


Figura 3: Estructura de contenedores y contenido

El grupo de los contenedores se ve cumplimentado por la existencia de una entidad denominada flujo de mensajes. A través de la utilización de esta entidad es posible representar el paso de mensajes en las organizaciones. En este caso el flujo de mensajes puede ser entre compartimentos, entre actividades o entre eventos de mensaje. La siguiente figura ilustra el flujo de mensajes, representado por una flecha con línea discontinua, entre los eventos de dos contenedores.

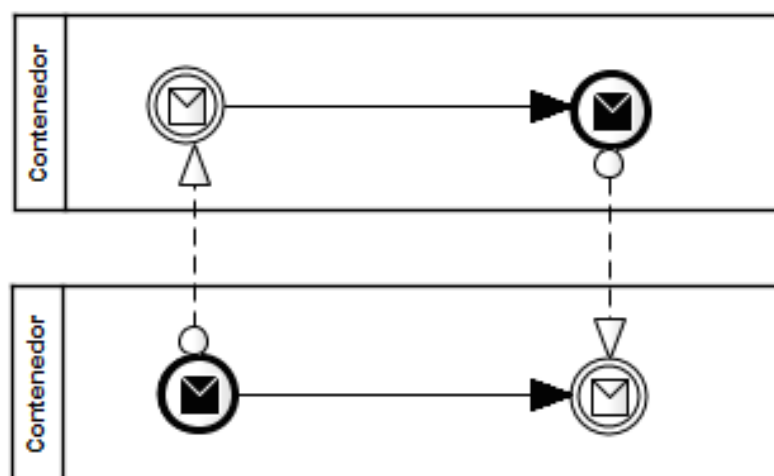





Figura 4: Flujo de mensaje entre contenedores



**3.4.1.2.6 Conversaciones**

Las conversaciones permiten representar el intercambio de mensajes entre compartimentos. Existen tres elementos dentro de este grupo, tal y como se muestra en la tabla incluida a continuación.

Nombre	Representación	Explicación
Comunicación		Define un conjunto de mensajes intercambiados y relacionados entre si. Cuando aparece con el símbolo de subprocesso indica una sub-conversación, que es un elemento compuesto de conversaciones.
Conector de conversación		Este elemento conecta las conversaciones con un participante.
Conector de conversación bifurcado		Este elemento conecta las conversaciones con sus múltiples participantes.

**Tabla 4: Conversaciones de BPMN**

La imagen que se muestra a continuación se corresponde con un ejemplo de utilización de las comunicaciones y los distintos tipos de conectores para realizar el intercambio de mensajes entre distintos compartimentos. Tal y como se puede observar, los compartimentos marcados como cerrados utilizan elementos de comunicación, representados por las formas hexagonales, para comunicarse. Además se puede observar como se utilizan los dos tipos de conector de conversación existentes.

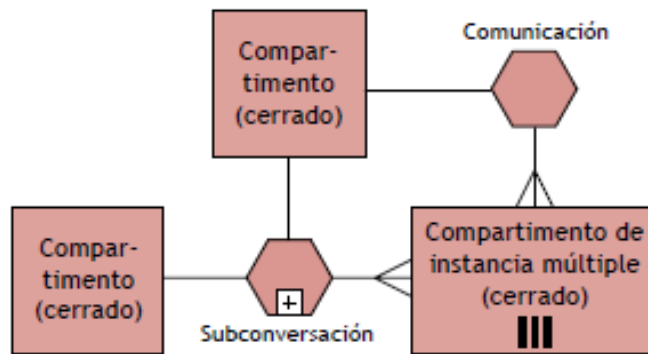


Figura 5: Diagrama de conversación [BPMN]

### 3.4.1.2.7 Coreografías

El apartado de coreografías está compuesto por tres elementos distintos. En primer lugar tenemos la tarea de coreografía (*choreography task*), que representa la interacción o intercambio de mensajes entre dos participantes, denominados *participants* en inglés.

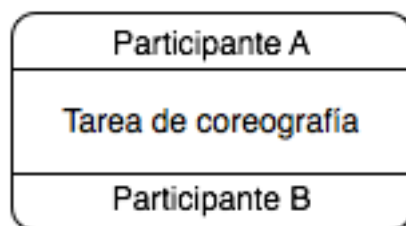


Figura 6: Representación de tarea de coreografía

Este elemento básico se puede ver cumplimentado por los otros dos elementos del grupo. La primera cumplimentación se obtiene por parte del indicador de múltiples participantes, que avisa de la existencia de más de un participante del mismo tipo. Esta condición se marca situando el símbolo de la siguiente imagen dentro de la representación del participante en la coreografía.



Figura 7: Indicador de múltiples participantes

El otro elemento que permite añadir connotaciones a la tarea de coreografía es el subproceso de coreografía, también conocido como *choreography sub-process*, que indica que esta contiene una coreografía refinada en múltiples interacciones. Esta característica se representa a través de la utilización del símbolo que representa al subproceso en combinación con el de la tarea de coreografía.

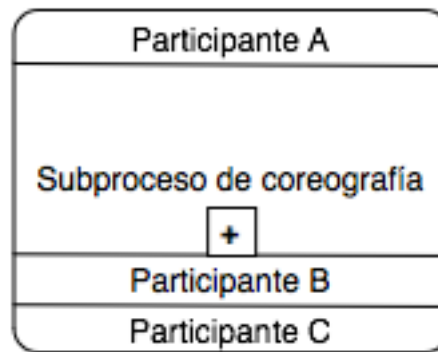


Figura 8: Subproceso de coreografía

A través de la utilización de estos elementos que componen el grupo que nos ocupa se pueden componer diagramas de coreografía como el del siguiente ejemplo, donde se encuentran representados dos de los elementos del grupo.

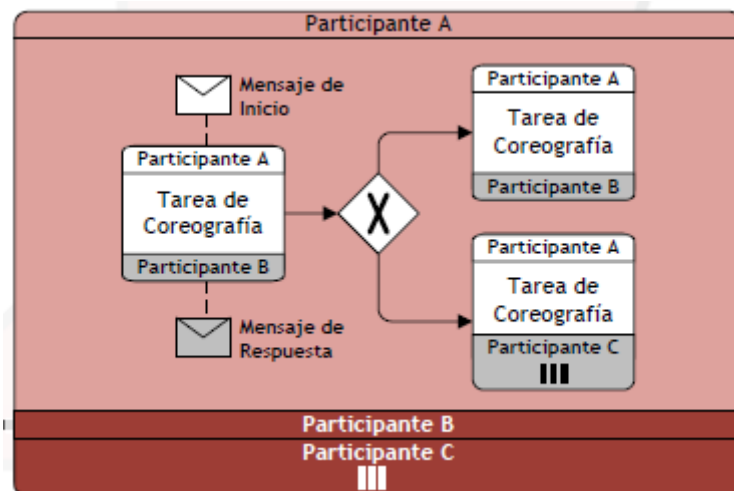


Figura 9: Diagrama de coreografía [BPMN]

### 3.4.1.3 Modelado de procesos de negocio con BPMN

BPMN define un elemento raíz denominado BPD (Business Process Diagram) que está compuesto por un conjunto de contenedores y compartimentos que albergan elementos de los grupos presentados anteriormente, permitiendo representar un proceso de negocio.

La ilustración que se incluye a continuación muestra un ejemplo de proceso de negocio modelado usando BPMN, haciendo uso de un conjunto amplio de los símbolos presentados anteriormente.

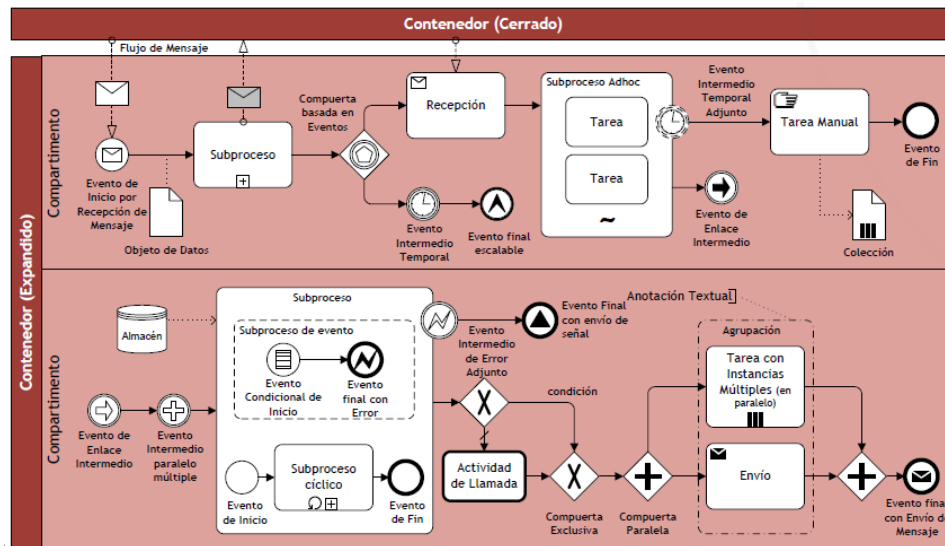


Figura 10: Ejemplo de diagrama BPMN [BPMN]

La parte central de la ilustración muestra un contenedor expandido para que se pueda comprobar como se relacionan algunos de los tipos de entidades existentes en BPMN. Por ejemplo, en el compartimento inferior del contenedor expandido se puede observar como se utiliza un almacén para introducir datos en un subproceso y como se utilizan los eventos en dicho subproceso para marcar la posibilidad de errores o el inicio y el fin de un subproceso cíclico. En la parte superior de la ilustración también se puede observar el uso del flujo de mensajes para establecer la comunicación entre el contenedor expandido y el contenedor que se encuentra cerrado, lo que permite realizar el paso de mensajes entre los contenedores para crear una colaboración entre las entidades incluidas en cada uno de ellos.

### 3.4.1.4 Desventajas de BPMN

El principal inconveniente de BPMN es su complejidad. Debido a la necesidad de ofrecer soporte para todo tipo de procesos por el hecho de tratarse de una notación estándar de modelado, el número de entidades que ofrece es demasiado amplio. Esto no supone un problema si el personal que trabaja con la notación dispone de unos conocimientos informáticos y de modelado suficientes; sin embargo, la utilización de esta notación por parte de expertos de un dominio concreto que no dispongan de unos conocimientos informáticos amplios acarrea problemas. Esta circunstancia queda expuesta en las conclusiones del artículo de Wahl y Sindre [Wahl and Sindre 2005] donde se establece que un usuario sin conocimientos técnicos necesitará un periodo de tiempo considerable para lograr aprender a utilizar la notación de forma correcta.

Además, los resultados de unos estudios realizados por Recker [Recker 2008] y por Chinosi y Trombetta [Chinosi and Trombetta 2012] establecen la falta de uso que experimentan algunos de los símbolos incluidos en esta notación. La gráfica incluida a continuación, que resume los resultados incluidos en el trabajo de Recker referenciado anteriormente, muestra una comparativa del uso de dichos símbolos en base a las respuestas ofrecidas por los usuarios en el estudio de Recker; el color azul (*unused*)

representa la falta de uso del símbolo mientras que el verde (*sometimes used*) establece que el símbolo es utilizado en algunas ocasiones y el color amarillo (*important*) indica que se trata de una entidad importante. En dicha gráfica se puede observar como algunos de los símbolos incluidos en BPMN obtienen mayor incidencia de los colores granate y violeta, lo que los identifica como símbolos sin uso o utilizados solamente en algunas ocasiones.

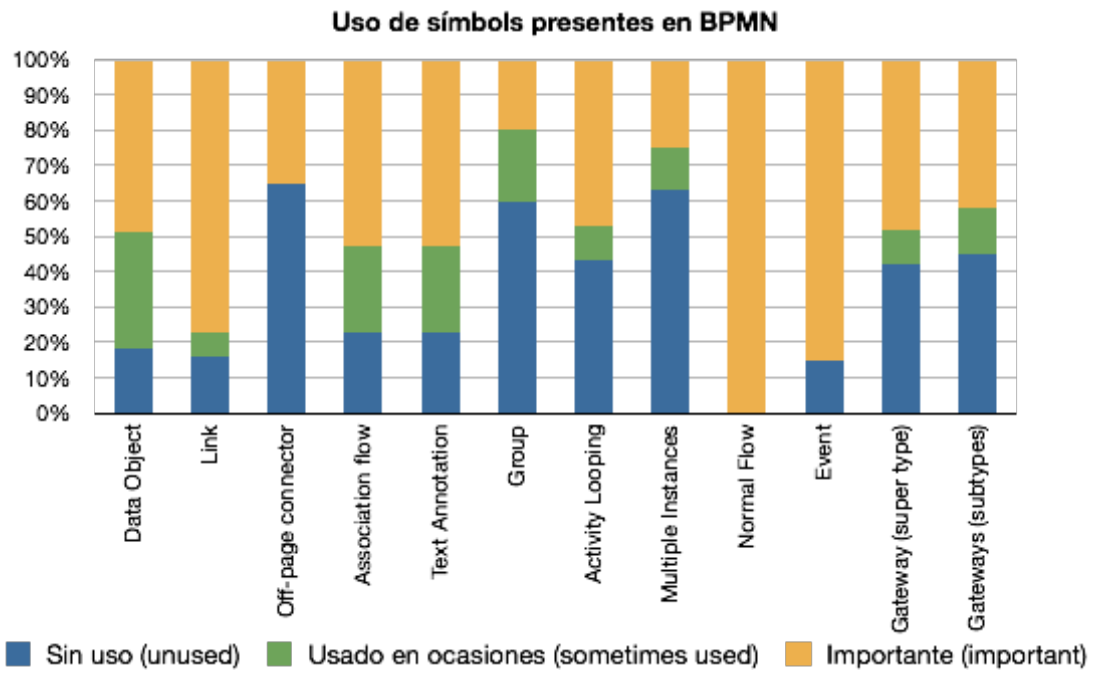


Figura 11: Gráfica de uso de símbolos de BPMN

Esta gráfica queda respaldada también por un fragmento del libro publicado por White y Miers [White and Miers 2008] en el que existe un apartado dedicado exclusivamente a la complejidad de BPMN donde se especifica que es improbable que el analista de negocio y el usuario final necesiten la mayor parte de los símbolos existentes en el estándar; según los autores, esta circunstancia es debida a la relación de estos símbolos con la semántica de ejecución del modelo. Este tipo de afirmaciones sobre la necesidad de formación específica para la utilización correcta de BPMN por parte de sus usuarios son compartidas también por Recker en su publicación del año 2010 [Recker 2010].

Aparte de la complejidad presentada por BPMN y acreditada en los trabajos referenciados, otro inconveniente existente son las herramientas de modelado que la soportan. Si bien en la publicación de presentación de BPMN [White 2004] se establece que uno de los objetivos del estándar es intentar terminar con las inconsistencias introducidas por la amplia variedad de herramientas de modelado existentes, la situación actual no es la esperada; tal y como refleja un estudio publicado en el año 2010 [Zhiqiang et al. 2010] la creciente popularidad de BPMN ha conseguido que aparezcan un mayor número de herramientas con soporte para ella pero las interpretaciones del estándar realizadas por sus autores así como las decisiones sobre qué elementos del estándar adoptar para su uso hacen que la diferencia de utilización entre unas herramientas y otras sea notable.

### 3.4.2 Diagramas de actividad de UML

Los diagramas de actividad de UML, también conocidos como UML Activity Diagrams o UML AD, son un tipo de diagramas recogidos dentro del lenguaje estándar de modelado UML [UML]. Según el artículo publicado en 2006 por Eloranta, Kallio y Terho [Eloranta et al. 2006] este tipo de diagramas son inferiores en lo que a expresividad se refiere en comparación con BPMN.

#### 3.4.2.1 Historia de los diagramas de actividad de UML

El lenguaje unificado de modelado, conocido también por su acrónimo UML, es un lenguaje de modelado de propósito general propuesto por el OMG. Desde su aparición en escena, hace más de 10 años, este lenguaje ha sido revisado y mejorado hasta alcanzar la versión actual: la 2.0. El alcance conseguido por este lenguaje es muy amplio, ofreciendo la posibilidad de modelar un amplio conjunto de dominios.

Los diagramas de actividad son los encargados de ofrecer soporte para el modelado de negocios dentro de UML, permitiendo también el modelado de lógica de aplicación para un escenario concreto. En la versión 1 de UML estos diagramas se caracterizaban por ser una modificación de los diagramas de estados mientras que en la segunda revisión de UML se optó por basarlos en la idea de las Redes de Petri.

#### 3.4.2.2 Elementos de los diagramas de actividad de UML

Al igual que ocurriera en el caso de la notación BPMN, los UML AD están divididos en una serie de conjuntos que agrupan entidades del mismo tipo. A continuación se realizará un pequeño recorrido por estos grupos, especificando los elementos contenidos en cada uno de ellos. Las imágenes incluidas a continuación han sido construidas utilizando la herramienta Enterprise Architect [Ent. Architect].

##### 3.4.2.2.1 Actividades

Las actividades dentro de UML AD son las entidades que representan una secuencia de comportamiento. Estas entidades se podrían definir como un conjunto de acciones, flujos de control y otros elementos existentes en la notación.

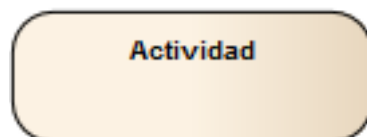


Figura 12: Representación de una actividad en UML AD

##### 3.4.2.2.2 Acciones

Las acciones son el núcleo principal de los diagramas de actividad UML. Una acción es una tarea que no puede ser descompuesta o dividida en varias, de tal forma que es la unidad mínima que se puede encontrar en un diagrama de actividad UML. Se podría considerar

que una acción es un paso dentro de una actividad, por lo que cada actividad contendrá múltiples acciones.

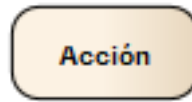


Figura 13: Acción de UML Activity Diagrams

Una de las características de estos diagramas es la posibilidad de incorporar restricciones a las distintas acciones que se utilicen en un proceso. En concreto, la forma más utilizada para definir estas restricciones son las pre-condiciones y post-condiciones, tal y como muestra la siguiente figura.

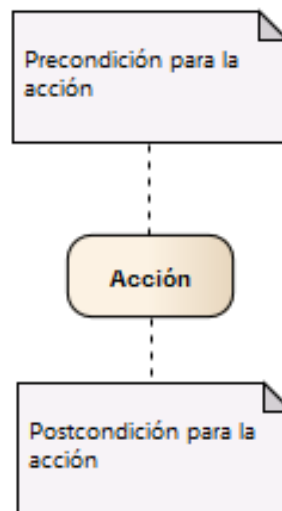


Figura 14: Restricciones aplicadas a una acción

#### 3.4.2.2.3 Puntos de inicio y fin

Los puntos de inicio y fin son los elementos que marcan el comienzo y el final de un proceso. En el caso de los UML AD el punto de inicio está representado por un círculo de color negro.



Figura 15: Punto de inicio

En lo referente a los puntos de fin es necesario señalar que UML AD disponen de dos tipos diferenciados de puntos de fin. El punto de fin para las actividades está representado por

un círculo blanco con un círculo negro en su interior mientras que el punto de fin de flujo está representado por un círculo blanco con un aspa en el interior.



Figura 16: Punto de fin para actividades



Figura 17: Punto de fin de flujo

#### 3.4.2.2.4 Entidades de control de flujo

Existen tres entidades dentro de los diagramas de actividad de UML que permiten realizar el control de flujo dentro de un proceso. En primer lugar encontramos el conector estándar de este tipo de notaciones, que permite conectar dos actividades de forma secuencial y establece la dirección del flujo utilizando una punta de flecha.



Figura 18: Conexión de dos acciones

El segundo elemento de control de flujo es la decisión. Estos elementos permiten la toma de alternativas en la ejecución de un proceso en base a una condición de entrada. Las decisiones se componen de dos elementos gráficos: el primero (el nodo decisión) realiza la bifurcación del flujo mientras que el segundo (el nodo *merge*) une los flujos para realizar la salida de forma correcta, si bien solamente uno de ellos habrá sido ejecutado.

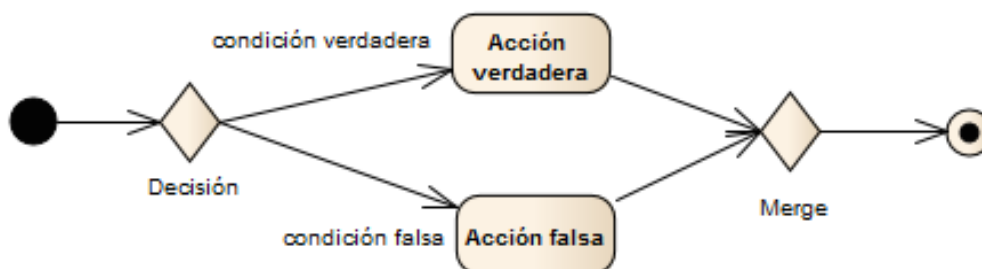


Figura 19: Entidad decisión de UML AD



La última entidad de control de flujo es el *fork*, que permite la ejecución de acciones en paralelo. Al igual que en el caso de las decisiones, este elemento está representado por dos elementos gráficos (el nodo *fork* en primer lugar y el nodo *join* en segundo); sin embargo, en este caso el segundo elemento, que compone un solo flujo a partir de varias entradas, solamente se realiza en el momento en el que las dos entradas han sido ejecutadas.

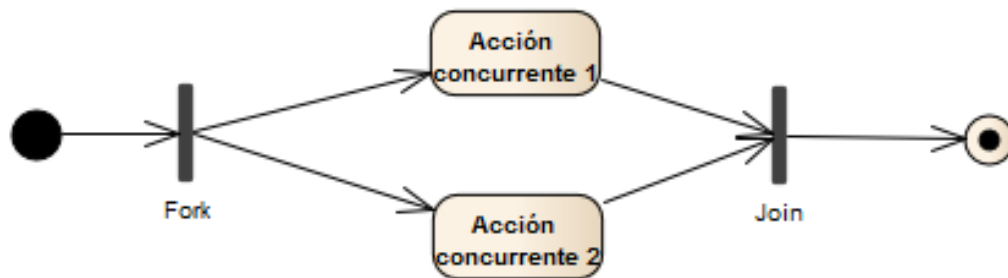


Figura 20: Entidad fork

#### 3.4.2.2.5 Los objetos y su flujo

Un flujo de objetos se establece como un camino o ruta del proceso que permite el paso de objetos entre las distintas acciones. Cada objeto estará representado por la figura incluida a continuación.

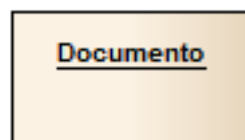


Figura 21: Representación de un objeto

El flujo de los objetos se denota a través de los conectores normales de UML AD, de tal forma que el objeto tenga un flujo de entrada y uno de salida.

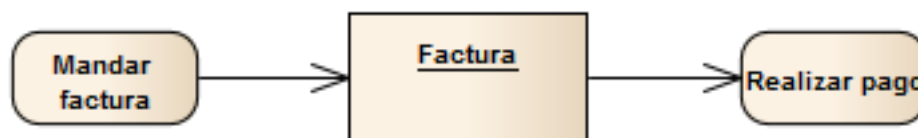


Figura 22: Flujo de un objeto factura

Además, este grupo contiene también una entidad que denota un almacén de datos. Esta entidad estará representada por un objeto identificado con la palabra clave “datastore”, que indicará que ese objeto realiza el almacenamiento de datos.



Figura 23: Objeto de tipo datastore

### 3.4.2.2.6 Particiones

Las particiones permiten representar el mismo tipo de situaciones que los compartimentos de BPMN. Por ejemplo, las particiones permiten establecer que la participación de varios departamentos o secciones de una empresa dentro de una misma actividad, tal y como muestra la imagen incluida a continuación.

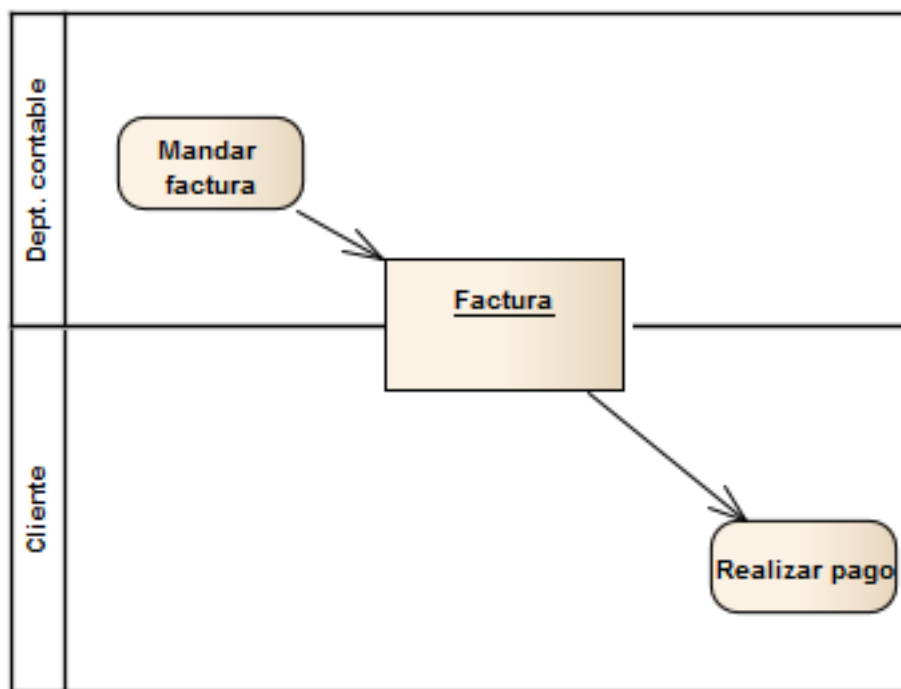


Figura 24: Representación de una partición

### 3.4.2.3 Modelado de procesos de negocio con diagramas de actividad UML

Estos diagramas se basan en la utilización de una entidad llamada Activity que representa una acción a realizar, ya sea una actividad que pueda realizarse físicamente o una actividad automática. Además de las actividades, los UML AD proporcionan otra serie de elementos que son comunes a la mayor parte de las notaciones para el modelado de procesos de negocio como las decisiones y las actividades en paralelo.

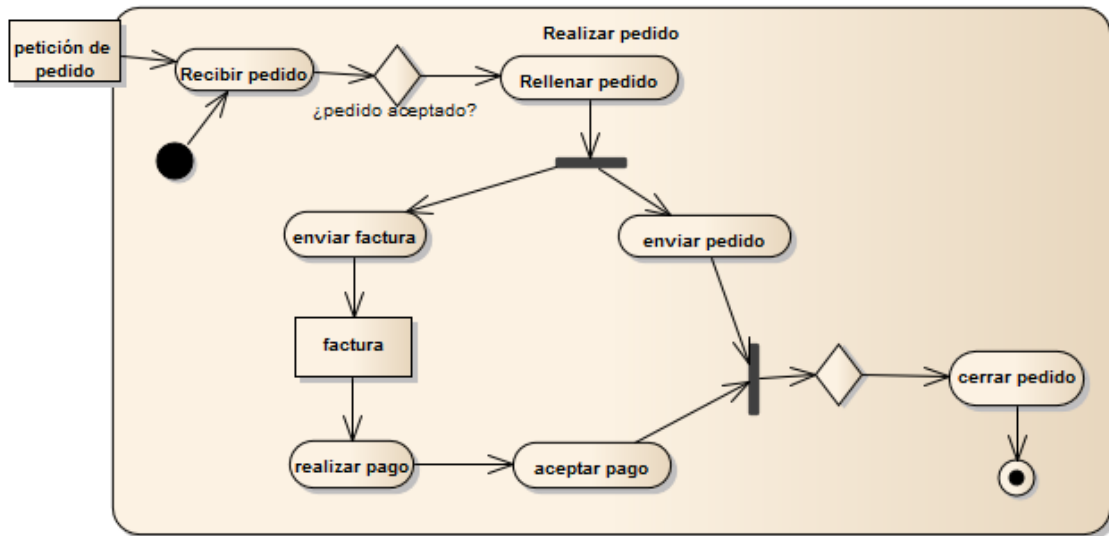


Figura 25: Ejemplo de diagrama de actividad UML

El diagrama que se muestra en la figura anterior representa el procesamiento de una orden de pedido. Cuando se recibe la orden lo que se produce es el procesamiento de la misma, que pasa por distintas tareas: la creación de la factura, la aceptación del pago o el envío del pedido. Tras todas las distintas actividades por las que pasa el pedido, la última actividad que se incluye en el diagrama es el cierre del pedido, tras lo cual se incluye el punto de fin del diagrama de actividad.

#### 3.4.2.4 Inconvenientes de los diagramas de actividad de UML

El primer contratiempo notable es el hecho de que existe más de una entidad con la misma representación gráfica, como en el caso de las entidades *decision* y *fork*. Estas entidades están compuestas por dos representaciones gráficas idénticas que tienen como única diferencia el sentido de las flechas que conectan con ellas. En el caso de introducir esta notación a expertos del dominio con pocos conocimientos en informática creemos que el hecho de que una misma representación gráfica represente dos comportamientos distintos puede ser un obstáculo a la hora de interpretar un diagrama para un proceso concreto.

Sin embargo, se considera como un mayor inconveniente el bajo nivel de abstracción ofrecido por este tipo de diagramas, denotado por el hecho de que UML es un lenguaje de modelado de propósito general. Esta característica de UML es una circunstancia que entra en conflicto con uno de los principales objetivos marcados al comienzo del proyecto: la usabilidad de la notación por parte de expertos del dominio con pocos conocimientos informáticos. En concreto esta circunstancia se hace notar, entre otros aspectos, en la inclusión de las restricciones de las actividades o en la posibilidad de incluir acceso a bases de datos a través de las entidades *datastore*.

Además, es necesario resaltar que el estudio publicado en 2006 por Russell, van der Aalst, ter Hofstede y Wohed [Russell et al. 2006] establece en el apartado de conclusiones que

si bien los UML Activity Diagrams son ampliamente utilizados para modelar procesos de negocio estos diagramas no son adecuados para capturar todos los aspectos directamente relacionados con este tipo de modelado.

### 3.4.3 jPDL: Java Process Definition Language

jPDL es el lenguaje gráfico de definición de procesos utilizado por las versiones iniciales de la herramienta JBoss jBPM [JBoss] [jBPM], una suite escrita en Java que permite utilizar BPM bajo esta plataforma.

jPDL es una adaptación de la notación BPMN donde se reduce el número de elementos ofrecidos para simplificar la tarea de la definición de los procesos. A través de esta simplificación, y cambiando la representación gráfica de algunos de los elementos para facilitar su comprensión, se consigue reducir la complejidad de BPMN y subir un poco el nivel de abstracción.

#### 3.4.3.1 Elementos del lenguaje jPDL

A diferencia de lo ocurrido en las notaciones anteriores, jPDL no ofrece una clasificación de sus entidades en tipos. En este caso, las entidades se presentan de forma conjunta en el entorno de trabajo. Sin embargo, se puede realizar una clasificación de las entidades que ofrece en tres grupos: puntos de inicio y fin, actividades y estados y, por último, elementos de control de flujo.

##### 3.4.3.1.1 Puntos de inicio y fin

jPDL ofrece un único tipo de punto de inicio, representado mediante un círculo de color verde. Sin embargo, a la hora de representar los puntos de fin de proceso ofrece tres alternativas (de izquierda a derecha en la Figura 27): un punto de fin normal, un punto de fin asociado a la cancelación del proceso y un punto de fin para casos en los que se produzcan errores.



Figura 26: Punto de inicio de proceso en jPDL



Figura 27: Puntos de fin de jPDL

### 3.4.3.1.2 Actividades y estados

Las actividades y estados son las entidades que permiten especificar las distintas acciones que se acometen durante el proceso que se está modelando. En el caso de la versión 4 de jPDL, este grupo está compuesto por ocho elementos: los estados, las actividades HQL, las actividades SQL, las actividades Java, las actividades de tipo script, las actividades comunes, los subprocesos y las actividades personalizadas.

Todas las entidades pertenecientes a este grupo están representadas por el símbolo de la siguiente imagen, lo que se establece como un inconveniente de esta notación en el análisis que se realiza más adelante. La única diferencia entre estas entidades es el contenido que se genera en el XML que representa al proceso, donde cada actividad o estado está representado por un contenido distinto que permite configurar sus distintas opciones.

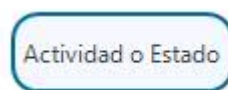


Figura 28: Actividad o estado del lenguaje jPDL

### 3.4.3.1.3 Elementos de control de flujo

Los elementos de control de flujo que ofrece jPDL son dos: las decisiones y los *fork*, compuestos por un elemento *fork* y otro elemento *join*. Las decisiones permiten tomar alternativas en la ejecución de un proceso en base a una condición de entrada mientras que los *fork* permiten la ejecución de actividades en paralelo.

Conforme a la representación gráfica de estas entidades es necesario señalar que la decisión está representada por un rombo con un aspa en su interior mientras que los *fork* tienen, al igual que ocurriera en UML AD, dos elementos gráficos iguales (uno a la entrada, llamado *fork*, y otro a la salida, llamado *join*) representados por un rombo con una cruz en su interior. La figura incluida a continuación muestra el elemento decisión a la izquierda y el elemento *fork* a la derecha.



Figura 29: Decisión y fork de jPDL

## 3.4.3.2 Modelado de procesos de negocio con jPDL

La imagen incluida a continuación muestra un modelo de proceso realizado con jPDL. Tal y como se puede observar, la inclusión de código de colores para los puntos de inicio y fin supone una mejora a la hora de comprender mejor el proceso que se ha modelado. Sin embargo, el hecho de que no podamos saber qué entidad representa cada una de las acciones que componen el proceso implica que no se pueda comprender el proceso de forma completa.

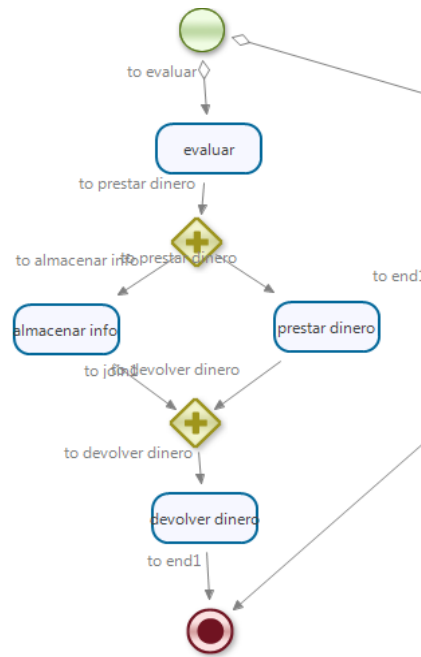


Figura 30: Proceso de negocio de ejemplo modelado con jPDL

El diagrama mostrado en la imagen anterior se corresponde con el proceso de negocio que representa la realización de un préstamo monetario. Al iniciarse el proceso se produce la evaluación de la propuesta, tras la que el proceso se bifurca en dos ramas: por un lado se produce un almacenamiento de la información y por el otro se produce el préstamo del dinero en si. Tras esta bifurcación se produce de nuevo la unión del proceso en una única rama, llegando a la actividad de devolución del dinero que marca el fin del proceso de negocio.

### 3.4.3.2 Inconvenientes de jPDL

A pesar de que los cambios realizados permiten subir un poco el nivel de abstracción de jPDL con respecto a BPMN, existen algunos aspectos del lenguaje que hace que no se adecúe a este proyecto. En primer lugar, el número de entidades ofrecidas para la realización de los modelados es demasiado amplio; por ejemplo, existen varias alternativas para representar las actividades o acciones (acciones HQL, acciones SQL, script, etc.) lo que hace complicado la selección de la tarea adecuada para cada punto del diagrama.

En segundo lugar es necesario señalar la dificultad que entraña utilizar esta notación en cuanto a la diferenciación entre los tipos de actividades se refiere. La representación gráfica de todas las entidades incluidas en el grupo de actividades y estados es la misma, por lo que es imposible distinguir una entidad de otra a simple vista; la única diferenciación existente es el nombre con el que aparecen en el diagrama del proceso en el momento en que son arrastradas al mismo, por lo que una vez cambiado el nombre no se pueden diferenciar. También en este sentido, la representación de las entidades de fork y join es la misma, al igual que ocurriera en el caso de los UML Activity Diagrams.

En último lugar es necesario señalar que el nivel de abstracción que presenta jPDL sigue sin ser suficientemente alto como para cumplir el nivel de usabilidad que exige este

proyecto. La utilización de algunas de las entidades comprendidas en el lenguaje implica la necesidad de tocar el código generado en el fichero XML para configurar algunos parámetros de las mismas, por lo que el nivel de abstracción disminuye y se dificulta el uso del lenguaje.

### 3.4.4 SBPMN: Simple BPMN

SBPMN [Fernández 2008] [Fernández et al. 2009] [Fernández et al. 2010] es una reducción y simplificación de BPMN realizada por un grupo de personas de la Universidad de Oviedo. SBPMN surge a partir del uso arbitrario de algunos de los símbolos de BPMN y el nivel de conocimiento técnico necesario para comprender el cometido de alguno de sus símbolos.

#### 3.4.4.1 Elementos de SBPMN

Debido a su carácter de simplificación de BPMN, esta notación propone una separación de las entidades en grupos al igual que el estándar. Así, los grupos en los que se dividen las entidades de SBPMN son los siguientes: decisiones, eventos, tareas, objetos de datos y subprocesos.

##### 3.4.4.1.1 Tareas

El grupo de tareas de SBPMN está compuesto por tres tipos de tareas diferentes: las tareas humanas, las tareas simples y las tareas automáticas. En este caso este número de entidades supone un incremento con respecto a BPMN, donde solamente existe una entidad de tipo tarea, justificando esta elección por la mayor información semántica que aportan estas entidades con respecto al tipo de proceso.

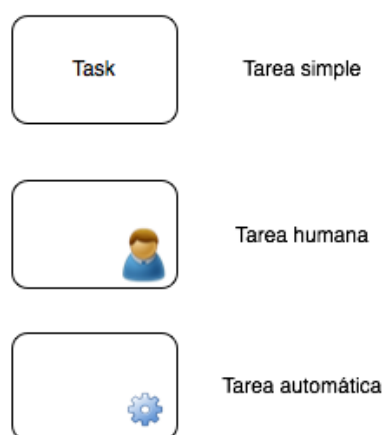


Figura 31: Representación de las tareas en SBPMN

##### 3.4.4.1.2 Eventos

SBPMN propone la eliminación de los subconjuntos de inicio, intermedio y fin que propone BPMN para la clasificación de sus eventos. Así, SBPMN ofrece distintos tipos de eventos que pueden ser utilizados en cualquier parte de un proceso. Además, los eventos incluidos en SBPMN han sido rediseñados de tal forma que su representación gráfica suponga una mejora de la comprensión de su significado para los expertos del dominio.






Eventos SBPMN	
	Evento de Inicio
	Evento de Fin
	Evento de Cancelación
	Evento de Rol
	Evento de Temporización
	Evento de Mensaje
	Evento de Excepción

Figura 32: Conjunto de eventos definidos en SBPMN [Fernández 2008]

#### 3.4.4.1.3 Decisiones

En el caso de las decisiones, SBPMN propone una reducción drástica conforme a BPMN. Tal y como se observó en el recorrido por BPMN, el estándar propone la existencia de siete tipos de decisiones mientras que SBPMN propone la existencia de un único tipo de esta entidad, cuyo comportamiento varía en función de las condiciones establecidas por el experto. En lo que a la representación gráfica de esta entidad se refiere, SBPMN apuesta por mantener el símbolo con forma de rombo utilizado por la mayor parte de las notaciones gráficas de modelado de procesos de negocio, contando con la eliminación de la decoración interior que este símbolo presentaba en BPMN.

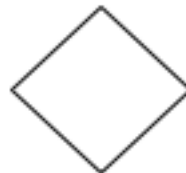


Figura 33: Representación de la decisión en SBPMN

#### 3.4.4.1.4 Objetos de datos



SBPMN también supone una reducción en el número de objetos de datos ofrecidos. En concreto SBPMN ofrece tres objetos de datos: un objeto de datos XSD, un objeto para representar servicios web y un objeto de base de datos. Según su autor, esto simplifica la inclusión de estos elementos en los procesos diseñados por los expertos al tiempo que se facilitan al usuario elementos que se ajustan a tareas específicas.



Figura 34: Conjunto de objetos de datos propuestos en SBPMN [Fernández 2008]

#### 3.4.4.1.5 Subprocesos

En el grupo de subprocesos, SBPMN propone una nueva simplificación drástica ya que solamente se contempla la existencia de un solo tipo de subproceso: el subproceso simple. En este caso, el subproceso indicará la existencia de un sub-diagrama para la representación del subproceso. De esta forma se consigue reducir el punto de complejidad de elección de subprocesos en BPMN.

#### 3.4.4.1.5 Otros elementos de SBPMN

Existe un grupo de elementos dentro de la especificación de SBPMN que no han sufrido cambios con respecto a BPMN debido a que no suponen un aumento de complejidad ni una carga adicional de contenidos técnicos para los expertos del dominio. Estos elementos son los siguientes: contenedores, compartimentos, flujos de mensajes y anotaciones.

#### 3.4.4.2 Cambios propuestos por SBPMN

Los cambios realizados en SBPMN, tal y como se pudo comprobar en el recorrido por sus elementos, se corresponden con las siguientes entidades de BPMN: los eventos, las decisiones, los artefactos, las tareas y las subtareas. Estas modificaciones fueron introducidas por las dificultades que encontraban los usuarios para identificar y comprender el cometido de las entidades.

Además, la representación gráfica de algunos elementos fue modificada para facilitar la comprensión de los modelados realizados con esta reducción de BPMN. Un ejemplo de un modelado realizado con SBPMN se puede observar en la figura incluida a continuación, donde se observan tanto la simplificación de algunos elementos como las decisiones y las tareas como la inclusión de las nuevas representaciones gráficas de algunas de las

entidades. El diagrama en cuestión representa el proceso de negocio asociado a la reserva de un viaje.

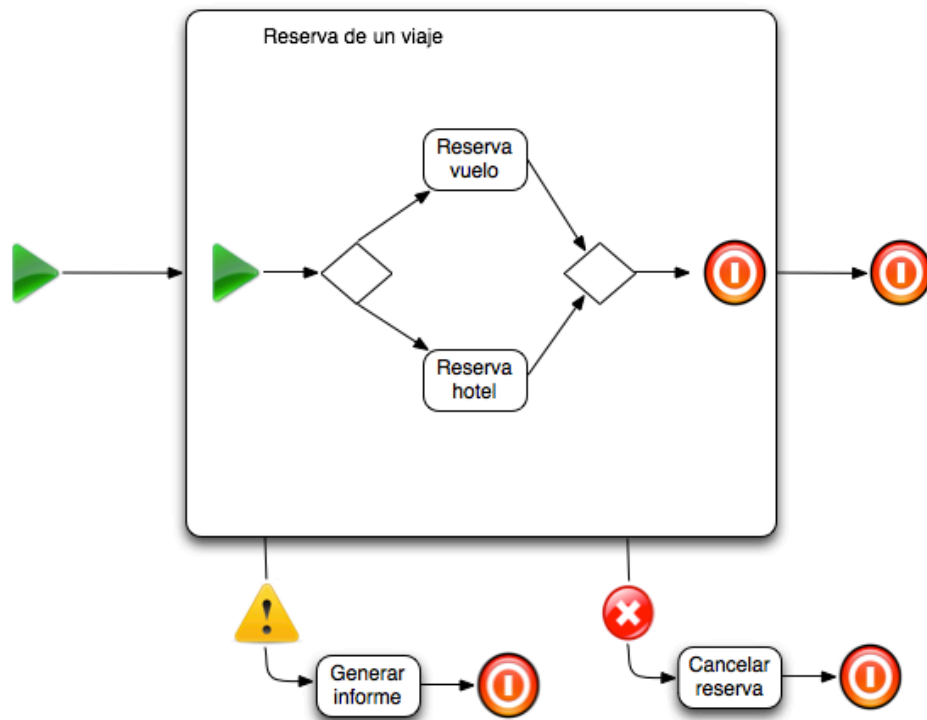


Figura 35: Ejemplo de proceso de negocio modelado con SBPMN

### 3.4.4.3 Inconvenientes de SBPMN

Al igual que ocurriera con algunas de las notaciones presentadas anteriormente, se considera que el número de entidades de SBPMN es elevado para el tipo de usuarios a los que se orienta el modelado de procesos de negocio: expertos del dominio con poco o ningún conocimiento de informática.

Por ejemplo, SBPMN ofrece tres representaciones distintas para la entidad actividad o tarea (tarea humana, tarea simple y tarea automática); así, igual que ocurriera en el caso de jPDL, el usuario podría experimentar dificultad en el modelado de un proceso al no saber qué tipo de actividad corresponde en cada momento. Bajo este mismo caso se encuentran los objetos de datos, donde se añaden una serie de detalles de implementación como son el acceso a bases de datos o a servicios web que se consideran de alto nivel de detalle con respecto al nivel pretendido para este tipo de usuarios.

### 3.5 Redes de Petri: Otra alternativa para el modelado de procesos

Si bien las redes de Petri no son una notación de modelado de procesos de negocio propiamente dicha, éstas son utilizadas de forma extendida para este cometido. A continuación se realizará un pequeño recorrido por ellas, estableciendo sus bases teóricas y dando a conocer tanto las cualidades que permiten utilizarlas para esta labor como aquellos inconvenientes que nos hacen descartarlas para el trabajo que nos ocupa.

#### 3.5.1 Historia de las Redes de Petri

Las redes de Petri surgen en la década de 1960 a partir del trabajo del matemático Carl Adam Petri [Petri 1962]. Son una representación matemática para el diseño de sistemas distribuidos discretos que conforman una generalización de la teoría de autómatas para la representación de eventos concurrentes. Las redes de Petri son utilizadas para el modelado de procesos de negocio debido a que ofrecen soporte para modelar un sistema en el que intervienen varios procesos en paralelo en pos de un objetivo común.

Una red de Petri estará formada por un conjunto de lugares, representados por círculos, y transiciones, representadas por rectángulos, que están acompañados por unas fichas (llamadas *tokens*), representadas por puntos negros, y conectados mediante arcos. No es posible que dos nodos iguales (por ejemplo dos lugares) estén conectados directamente.

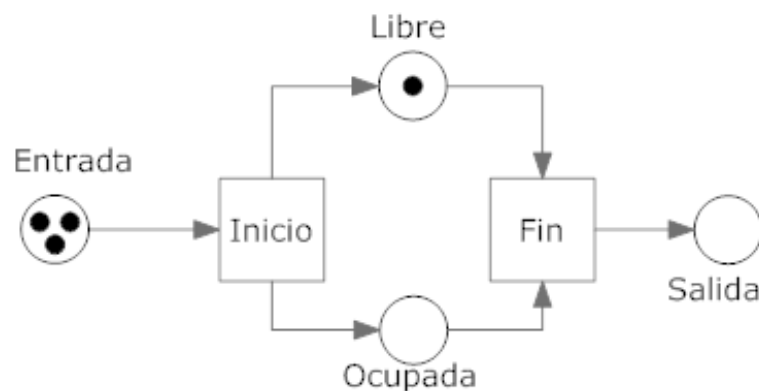


Figura 36: Red de Petri representando una máquina

Existen dos clases diferenciadas de redes de Petri: las redes de Petri de bajo nivel y las redes de Petri de alto nivel [Jensen and Kristensen 2009]. Las primeras se caracterizan por poseer unas características que las hacen adecuadas para el modelado teórico de la concurrencia, si bien algunas son usadas para el modelado y verificación de sistemas hardware. Por otro lado, las redes de Petri de alto nivel están más orientadas al uso práctico debido a la posibilidad de definir modelos compactos y parametrizados; se caracterizan por aunar las características de las redes de Petri con los lenguajes de programación.

### 3.5.2 Teoría sobre redes de Petri

Debido a que suponen una generalización de la teoría de autómatas, las redes de Petri están descritas [van der Aalst et al. 1998] por una tripleta  $(P, T, F)$  donde:

- $P$  es un conjunto finito de lugares.
- $T$  es un conjunto finito de transiciones de tal forma que  $(P \cap T = \emptyset)$ .
- $F$  es un conjunto de arcos de tal forma que  $F \subseteq (P \times T) \cup (T \times P)$ .

#### 3.5.2.1 Consideraciones sobre lugares

- Se dice que  $p$  es un lugar de entrada para una transición  $t$  si existe un arco que vaya de  $p$  a  $t$ . Por otro lado, se dice que  $p$  es un lugar de salida para una transición  $t$  si existe un arco que vaya de  $t$  a  $p$ .
- En todo momento un lugar podrá contener cero o más *tokens*.

#### 3.5.2.2 Consideraciones sobre transiciones

- Se denota  $p \bullet$  como el conjunto de transiciones que tienen al lugar  $p$  como lugar de entrada y  $\bullet p$  como el grupo de transiciones que tienen al lugar  $p$  como punto de salida.
- Se utiliza la representación  $\bullet t$  para denotar el conjunto de lugares de entrada para una transición  $t$  y  $t \bullet$  para representar al conjunto de lugares de salida para una transición  $t$ .
- Se considera que una transición  $t$  está activa si cada lugar de entrada  $p$  de  $t$  posee al menos una *token*. Una transición activa  $t$  puede dispararse, en cuyo caso consumirá una *token* de cada lugar de entrada  $p$  de  $t$  y producirá una nueva *token* en cada lugar de salida  $p$  de  $t$ .

#### 3.5.2.3 Estados de las redes de Petri

El estado de una red de Petri, representado por  $M$ , viene denotado por la distribución de las *tokens* en los distintos estados de la red. Por ejemplo, el estado de un red de Petri con tres lugares sería  $1p_1 + 2p_2 + 0p_3$  si el lugar  $p_1$  contuviera una *token*, el lugar  $p_2$  contuviera dos *tokens* y el lugar  $p_3$  no contuviera *tokens*; este mismo estado podría ser descrito también mediante la siguiente fórmula:  $p_1 + 2p_2$ .

##### 3.5.2.3.1 Comparación de estados

La comparación de los distintos estados de una misma red de Petri es posible. Para dos estados cualesquiera  $M_1$  y  $M_2$  de una red de Petri, se dice que  $M_1 \leq M_2$  si:

$$\forall p \in P, M_1(p) \leq M_2(p)$$

### 3.5.2.3.2 Notaciones sobre estados

Dada una red de Petri  $(P,T,F)$  y un estado  $M_1$ , existen las siguientes notaciones:

- $M_1 \xrightarrow{t} M_2$ : la transición  $t$  está activa y realizarla implica un cambio de estado de  $M_1$  a  $M_2$
- $M_1 \rightarrow M_2$ : existe una transición  $t$  tal que  $M_1 \xrightarrow{t} M_2$
- $M_1 \xrightarrow{\sigma} M_n$ : la secuencia  $\sigma = t_1 t_2 t_3 \dots t_{n-1}$  lleva a la red del estado  $M_1$  al estado  $M_n$
- $M_1 \xrightarrow{*} M_n$  (**estado  $M_n$  alcanzable desde  $M_1$** ): existe una secuencia  $\sigma = t_1 t_2 t_3 \dots t_{n-1}$  tal que  $M_1 \xrightarrow{\sigma} M_n$ . Es necesario señalar que en este caso se contempla la secuencia vacía ( $M_1 \xrightarrow{*} M_1$ ).

### 3.5.2.4 Propiedades de las redes de Petri

Usando la definición  $(PN, M)$  para hacer referencia a una red de Petri  $PN$  con estado inicial  $M$  y teniendo en cuenta que  $M'$  es un estado alcanzable de  $(PN, M)$ , se pueden definir las siguientes propiedades de  $(PN, M)$ :

- **Red de Petri viva:**  $(PN, M)$  es viva si para cada estado alcanzable  $M'$  y cada transición  $t$  existe un estado  $M''$  alcanzable desde  $M'$  y que habilita  $t$
- **Red de Petri limitada y segura:**  $(PN, M)$  es una red limitada si para cada lugar  $p$  hay un número natural  $n$  tal que para cada estado alcanzable el número de tokens en  $p$  es menor a  $n$ . Esta misma red se denomina segura si para cada lugar el número máximo de tokens no es superior a 1.
- **Camino:** Siendo  $(PN, M)$  una red y tomando dos de sus nodos  $n_1$  y  $n_k$  se considera que  $C$  es un camino si existe una secuencia  $(n_1, n_2, \dots, n_k)$ .
- **Alfabeto:** Tomando el camino anterior  $C$ , se dice que  $\alpha(C) = \{n_1, n_2, \dots, n_k\}$  es el alfabeto de  $C$  si  $(n_i, n_{i+1}) \in F$  para  $1 \leq i \leq k-1$
- **Camino Elemental:** Se dice que un camino  $C$  es elemental si para dos nodos cualesquiera  $n_i, n_j \in C, i \neq j \Rightarrow n_i \neq n_j$

### 3.5.3 Redes de Petri de alto nivel

Tal y como se comentó en uno de los apartados anteriores, una de las clases de redes de Petri que existen son las de alto nivel. Esta extensión, cercana a los lenguajes de programación, surge como respuesta a las limitaciones de las redes de Petri convencionales así como al tamaño y la complejidad que éstas adquieren a la hora de representar los procesos actuales [van der Aalst 1994] [van der Aalst et al. 1998]. Esta extensión permite la ejecución de las redes que se construyen [Jensen and Kristensen 2009] y utiliza algunos mecanismos existentes en los lenguajes de programación para definir condiciones en las transiciones.

Las redes de Petri de alto nivel se basan en la inclusión de tres propiedades dentro de las redes convencionales: el color, que permite añadir atributos a las *tokens* utilizadas en las redes de Petri para modelar de forma más ajustada a la realidad los objetos que fluyen a través de un proceso de negocio; el tiempo, para incluir los aspectos temporales relacionados con el comportamiento del sistema; y la jerarquía, que habilita la posibilidad de descomponer un sistema complejo en partes [van der Aalst 1994]. Es necesario señalar que estas tres propiedades constituyen de forma individual tres extensiones reconocidas de las redes de Petri convencionales [van der Aalst et al. 1998].

### 3.5.4 Cualidades de las redes de Petri para BPM

Según el artículo publicado por Van der Aalst et al. [van der Aalst et al. 1998] en el año 1998 sobre la aplicación de las redes de Petri para la gestión de procesos existen una serie de características que habilitan la utilización de las mismas en esta disciplina. Estas características son las siguientes: semántica definida formalmente, representación gráfica, expresividad, técnicas de análisis disponibles e independencia de la plataforma.

- **Semántica:** En primer lugar destaca el hecho de que un proceso especificado con las redes de Petri tendrá una definición clara y precisa. Esto es debido a que tanto las primitivas redes de Petri como las extensiones propuestas a posteriori poseen una semántica definida formalmente.
- **Representación Gráfica:** Las redes de Petri son un lenguaje gráfico que permite representar mediante nodos y transiciones un proceso. Esta representación gráfica habilita un aprendizaje sencillo de su utilización así como un flujo de comunicación directo con el usuario final.
- **Expresividad:** Todas las primitivas necesarias para el modelado de procesos están presentes en las redes de Petri. Además, el hecho de que los estados se representen de forma explícita habilita la utilización de hitos y alternativas.
- **Técnicas de Análisis:** Existen una gran cantidad de técnicas de análisis aplicables a las redes de Petri. A partir de estas técnicas es posible medir diversas características de las mismas como algunas de las propiedades incluidas en ellas (seguridad, invariantes) o el rendimiento. De esta forma es posible realizar una

evaluación de distintas versiones de un mismo proceso para conocer cuál se ajusta mejor a las necesidades.

- **Independencia de la Plataforma:** Las redes de Petri proveen un framework independiente de la plataforma para su modelado y análisis y no están basadas en ningún tipo de software propietario.

### 3.5.5 Usando redes de Petri para modelar procesos de negocio

Tal y como se ha mencionado anteriormente existen una serie de atributos o cualidades de las redes de Petri que las hacen oportunas para utilizarlas en el modelado de procesos de negocio. Sin embargo, a la hora de utilizarlas será necesario establecer una serie de correspondencias y restricciones que ajusten sus propiedades de forma estricta al modelado de procesos de negocio. A partir de la aplicación de estas restricciones nacen las Workflow Nets o WF-Nets, redes de Petri de flujo de trabajo.

#### 3.5.5.1 Correspondencias entre modelado de procesos de negocio y redes de Petri

Existen una serie de correspondencias que es necesario conocer a la hora de adaptar el modelado de procesos de negocio al escenario de las redes de Petri, redefiniendo ciertos conceptos clave del modelado de procesos de negocio.

- En primer lugar los procesos o actividades serán modelados como transiciones, que contendrán las tareas a ejecutar.
- Todos los tipos de condiciones (incluyendo las precondiciones y postcondiciones) serán representadas como lugares, especificando estos la habilitación de las distintas transiciones.
- Los casos o escenarios de uso particulares de cada proceso que no se puedan encuadrar en las categorías anteriores serán modelados a través del uso de Tokens.

#### 3.5.5.2 Descripción de una Workflow Net (WF-Net)

Una Workflow Petri Net o red de Petri de flujo de trabajo es una adaptación de las redes de Petri convencionales al modelado de procesos de negocio o flujos de trabajo. Existen dos restricciones que es necesario aplicar a una red de Petri para que esta pueda considerarse Workflow Net [**van der Aalst et al. 1998**]:

1. Deberá existir un lugar inicial denominado “i” y un lugar final denominado “o”. Una token en “I” se corresponderá con la existencia de un caso a tratar mientras que una token en “o” representará a un caso que ya ha sido tratado.

2. Todos los lugares y transiciones de una WPN deben contribuir al procesado de los *tokens*, por lo que toda transición  $t$  hacia un lugar  $p$  debe estar contenida en un camino desde “i” hasta “o”.

Así, una red de Petri  $PN = (P, T, F)$  será una red de Petri de flujo de trabajo o Workflow Net (WF-Net) si cumple que:

- PN tiene dos lugares especiales “i” y “o” tales que  $i$  es un punto de partida ( $\bullet i = \emptyset$ ) y  $o$  es un sumidero ( $o \bullet = \emptyset$ ).
- Si se añadiera una transición  $t^*$  a PN conectando  $o$  con  $i$  ( $\bullet t^* = \{o\}$  y  $t^* \bullet = \{i\}$ ) la red PN resultante estuviera fuertemente conectada.

### 3.5.5.2.1 Construcciones de control de flujo de las WF-Nets

Además de necesitar de las propiedades citadas anteriormente de manera formal, una Workflow Net dispone de una serie de construcciones especiales relacionadas con el control del flujo de los procesos de negocio. En concreto las WF-Nets disponen de una serie de construcciones que permiten las siguientes formas de ejecución del proceso: ejecución secuencial, ejecución paralela (*AND*), ejecución condicional (*implicit OR*) y ejecución condicional complementaria (*explicit OR o XOR*).

En este punto es necesario señalar que la utilización la representación de estas construcciones en las WF-Nets es considerado azúcar sintáctico, ya que las herramientas con soporte para este tipo de redes de Petri construyen una traducción de las mismas a una red de Petri normal de forma transparente para el usuario. Así, la apariencia de la WF-Net permanece simple a ojos del usuario mientras que internamente se asocia a una red de Petri más complicada. La representación gráfica de las construcciones AND, OR y XOR es la que se muestra en la siguiente figura.

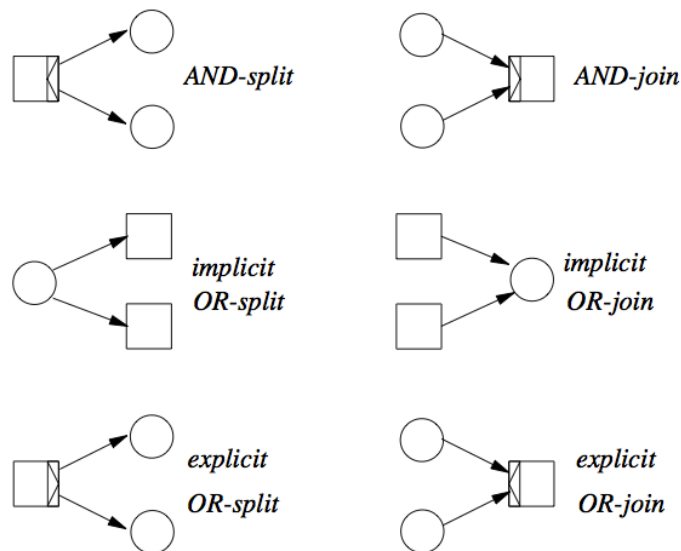


Figura 37: Representación de las condiciones AND, OR y XOR



- La ejecución en secuencia o secuencial de una WF-Net es el modo normal de ejecución de este tipo de redes. La ejecución secuencial se realiza de tal forma que si un elemento se ejecuta es porque todos los elementos anteriores a él han sido ejecutados en el orden indicado por los arcos que los unen.
- La ejecución paralela, posibilitada por la utilización de los componentes *AND-join* y *AND-split*, permite la ejecución de más de un camino de la WF-Net de forma simultánea. El componente *split* es el encargado de producir la separación de los caminos de ejecución de la red, provocando que de un misma transición surjan dos arcos de salida. A partir de ese momento se ejecutan los dos caminos del proceso de forma simultánea hasta que se utiliza la construcción *AND-join* para devolver a la red a su estado de ejecución secuencial.
- La ejecución condicional, también denominada OR por su comportamiento idéntico al de la operación lógica OR, permite la toma de una decisión en la ejecución del proceso. Mediante la utilización de la construcción *OR-split* se toma una decisión en base a una condición en la transición y se activa el camino correspondiente a la evaluación verdadera de la condición. Una vez ejecutado el camino que se ha activado, la construcción *OR-join* devuelve la red a su estado de ejecución secuencial.
- La ejecución condicional complementaria o XOR tiene un comportamiento parecido a la ejecución condicional ya que permite la toma de una decisión en base a una condición de entrada. La diferencia existente en este caso radica en la activación de los caminos ya que si bien la ejecución condicional solamente permitía la activación de un único camino en ese caso es posible que se active más de uno; en concreto la ejecución condicional complementaria permite tres opciones: que se active un camino, que se active el otro camino o que se activen los dos al mismo tiempo.

Tal y como se puede observar a partir de las construcciones de control de flujo presentadas anteriormente, existen un componente de apertura y un componente de cierre para cada uno de estos elementos. Existe una restricción respecto al uso de estas estructuras de control de flujo que si bien no se corresponde con una limitación especificada en la definición de este tipo de redes de Petri está relacionada con un correcto diseño de este tipo de redes: por cada conector *split* de cualquier tipo ha de utilizarse un conector *join* del mismo tipo. A través de la aplicación de esta pequeña norma se consigue evitar la aparición de marcadores muertos y *tokens* innecesarios.

#### 3.5.5.2.2 Disparadores

El otro elemento típico de cualquier sistema de modelado de procesos de negocio es el disparador. A la hora de modelar cualquier proceso de negocio es necesario ofrecer un mecanismo que permita que una actividad o tarea se ejecute ante un acontecimiento dado, funcionalidad conseguida a través del uso de disparadores.

Las Workflow Nets disponen de cuatro tipos de separadores que permiten establecer las condiciones especiales de ejecución de una tarea. Estos disparadores son los siguientes:

- **Sin disparador:** Una tarea que no esté etiquetada de ninguna forma se considera libre de disparadores y será ejecutada de forma automática.
- **Disparador de recurso o usuario:** Una tarea etiquetada con este disparador será ejecutada en caso de que exista un cierto recurso material o humano a su disposición.
- **Disparador de mensaje:** Una tarea marcada con el disparador de mensaje se ejecutará en el momento en el que se reciba un mensaje o señal externa; por ejemplo, esperar a la recepción de un correo electrónico.
- **Disparador temporal:** Un disparador temporal impondrá a una tarea una condición temporal para su ejecución; por ejemplo, enviar un correo electrónico a una hora determinada.

Todas las tareas que dispongan de un disparador serán identificadas mediante la existencia de uno de los siguientes iconos: una flecha entrante para los disparadores de recurso o usuario, un sobre para un disparador de mensaje y un reloj para un disparador temporal. Estas representaciones gráficas se pueden observar en la figura incluida a continuación.

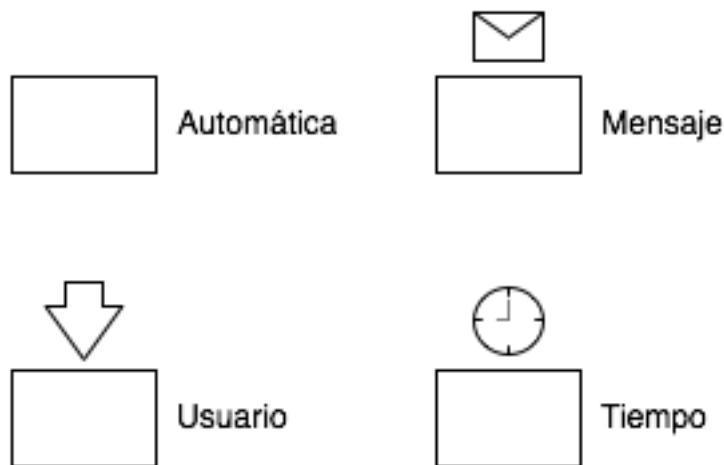


Figura 38: Disparadores de las WF-Nets

### 3.5.5.3 Modelado de procesos de negocio con Workflow Nets

Un trabajo recientemente presentado en la Universidad de Oviedo [Fernández González 2012] propone la utilización de las redes de Petri de flujo de trabajo o WPN para la definición de procesos de negocio que puedan ser traducidos a código y posteriormente ejecutados.

Este trabajo toma como punto de partida la herramienta Workflow Petri Net Designer (WoPeD) [WoPeD], una herramienta open-source para el modelado, la simulación y el

análisis de flujos de trabajo representados mediante este tipo especial de redes de Petri. A partir de esta herramienta gráfica de modelado se construye una WF-Net que representa un proceso de negocio utilizando las construcciones presentadas anteriormente.

Una vez construida la red a partir de WoPeD, se procede a transformarla a partir de la herramienta propuesta en este trabajo; de esta forma se consigue obtener un proyecto ejecutable de Windows Workflow Foundation (WWF) [WWF], una plataforma que se presentará más adelante. Esta plataforma ofrece soporte para la publicación y ejecución de los procesos de negocio así como para la creación de servicios web que los referencien; de esta forma, los servicios pueden ser utilizados como base para la construcción de aplicaciones web que recojan la interacción del usuario e introduzcan en el proceso los datos que este proporcione.

La herramienta de transformación propuesta en este trabajo recibe dos parámetros para que la transformación de la WF-Net en un proceso compatible de Windows Workflow Foundation sea realizada: por un lado se debe introducir el fichero XML resultante de la creación de la WF-Net con WoPeD y, por el otro, se ha de especificar la ruta o directorio en el que se desea que la herramienta almacene el contenido generado. A partir de estos datos la herramienta realiza el análisis pertinente de la WF-Net y genera los artefactos equivalentes bajo la plataforma WWF.

#### 3.5.5.4 Inconvenientes de las redes de Petri

Es indudable la potencia que otorgan las redes de Petri a la hora de modelar cualquier tipo de sistema, lo que es debido a su naturaleza de grafo. Además, tal y como se muestra en el apartado anterior, es posible utilizar las redes de Petri para el modelado de procesos de negocio de forma gráfica a través de las propiedades declaradas en la definición de las WF-Nets, la adaptación de las redes de Petri al modelado de procesos de negocio.

Sin embargo, existen un par de cuestiones que nos permiten descartar la utilización de las WF-Nets, y por extensión de las redes de Petri, para el modelado de procesos de negocio por parte de expertos del dominio con limitados conocimientos tecnológicos. En primer lugar es necesario señalar que el nivel de abstracción que se obtiene a partir de la utilización de las WF-Nets es demasiado bajo. Tal y como se observa en la figura 3.44 la definición de un proceso de negocio completo (incluyendo las restricciones y condiciones pertinentes) conlleva la utilización de sentencias de estructura muy cercana a los lenguajes de programación; en concreto en la imagen se pueden observar sentencias de tipo if/else y también accesos a posiciones de vectores. Este tipo de contenidos hacen que las WF-Nets estén más indicadas para los expertos informáticos que para los expertos del dominio del problema.

El otro inconveniente observado es la usabilidad de las WF-Nets. Además del nivel bajo de abstracción, las WF-Nets utilizan conceptos demasiado difíciles de comprender y asimilar por parte de los expertos del dominio. Por ejemplo, las WF-Nets definen las actividades del proceso como transiciones o arcos y los distintos estados por los que pasa el proceso durante su ejecución como lugares. Sin embargo, un experto ve su proceso como un conjunto de actividades a ejecutar y una serie de transiciones que conectan las actividades

según el flujo deseado para el proceso. Este tipo de cambios conceptuales provocan una confusión innecesaria en el experto, dificultando el modelado de los procesos de forma innecesaria y pudiendo provocar un clima de rechazo hacia la construcción de modelos de proceso utilizando esta técnica.

Vistos los dos principales inconvenientes, que atacan directamente a dos de los objetivos principales de este trabajo, es posible señalar también que la notación gráfica ofrecida por las WF-Nets puede resultar confusa. Las construcciones gráficas asociadas a las construcciones AND y XOR son demasiado parecidas como para que un proceso pueda ser comprendido de forma completa a simple vista. Además, la carencia de un código de color también merma la comprensión de este tipo de diagramas en comparación a alguna de las alternativas presentadas anteriormente.

### 3.6 CONCLUSIONES

Tal y como se puede observar en los apartados anteriores, todos los lenguajes y notaciones mencionados son capaces de modelar los procesos de negocios de distintos ámbitos de manera completa a través de los conjuntos de entidades que ofrecen. Sin embargo, el grado de complejidad y el nivel de abstracción que los expertos del dominio experimentan al utilizarlos se alejan de los niveles considerados como ideales para los objetivos marcados en esta investigación.

Una vez presentadas algunas de las notaciones más importantes y utilizadas para el modelado de procesos de negocio se realizará un estudio sobre herramientas y plataformas que soportan el uso de este tipo de notaciones para la creación de modelos de procesos de negocio.



# CAPÍTULO 4

## HERRAMIENTAS Y PLATAFORMAS CON SOPORTE PARA EL MODELADO DE PROCESOS DE NEGOCIO

---

### *4.1 INTRODUCCIÓN*

Una vez presentados los contenidos teóricos del modelado de procesos de negocio y de los lenguajes o notaciones que permiten la realización de estos modelados, es necesario realizar un estudio de las plataformas que ofrecen soporte para el desarrollo de aplicaciones utilizando BPM.

Debido a la expansión del modelado de procesos de negocio existen un gran número de herramientas y plataformas destinadas a esta actividad, pudiendo dividir este conjunto en dos: aquellas herramientas que permiten solamente el modelado de los procesos y las plataformas que permiten tanto el modelado como la ejecución de los procesos. A continuación se realiza un recorrido por algunas de las herramientas y plataformas existentes, haciendo hincapié en dos plataformas de modelado y ejecución de procesos que fueron seleccionadas como las principales alternativas para la realización de los prototipos que acompañan a esta investigación: jBPM y WWF.

Uno de los objetivos establecidos al inicio de la realización de este trabajo de investigación es la ejecución de los procesos de negocio. Por lo tanto, aquellas alternativas que permiten el modelado y la ejecución de los procesos serán las que mayor importancia cobren en este capítulo.

## 4.2 HERRAMIENTAS CON SOPORTE PARA EL MODELADO DE PROCESOS

Tal y como se explicó anteriormente, el grupo de herramientas y plataformas con soporte para el modelado de procesos está compuesto por aquellos programas que permiten únicamente la especificación de los modelos de los procesos de negocio a partir de la utilización de notaciones gráficas. Así, estas herramientas sirven para establecer un repositorio digital de procesos de negocio para el usuario, permitiéndole almacenar en soporte digital sus procesos de negocio para su visualización o modificación.

Según se podrá observar en las figuras que acompañan a la descripción textual de cada una de las herramientas, la mayor parte de ellas se basan en una disposición de elementos y una funcionalidad muy similar. Así, la mayoría de las herramientas disponen de una zona a la izquierda del interfaz donde se sitúan las entidades que pueden ser utilizadas para la composición de los modelos y reservan la zona central de la ventana para el área donde se realiza el modelado de los procesos.

### 4.2.1 Microsoft Visio

Microsoft Visio [**Visio**] es un producto software que se encuentra enmarcado dentro de la suite Microsoft Office. Se trata de una herramienta de dibujo vectorial que permite a los usuarios, a través de la utilización de un plugin de la notación BPMN, modelar los procesos de negocio de forma sencilla.

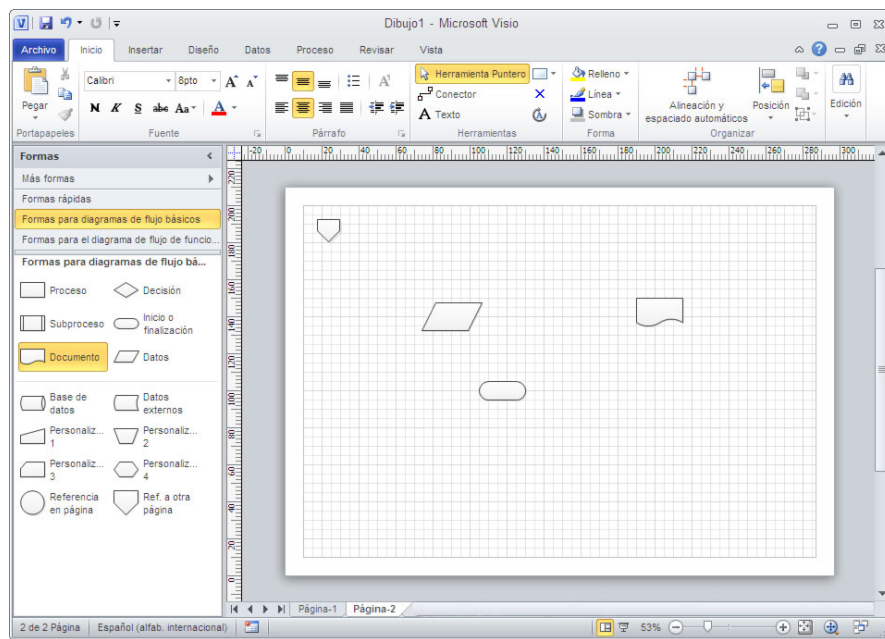


Figura 39: Interfaz de usuario de la herramienta Microsoft Visio



### 4.2.2 Enterprise Architect

Enterprise Architect **[Ent. Architect]**, la herramienta de modelado ofrecida por Sparx Systems, permite realizar el modelado de procesos de negocio a través de la versión 2.0 de la notación estándar de modelado de procesos de negocio BPMN. Además, permite al usuario generar ficheros en el lenguaje de ejecución de procesos BPEL (Business Process Execution Language) a partir de los modelos realizados con BPMN 2.0, si bien no ofrece soporte para la ejecución de estos ficheros.

En la figura incluida a continuación se puede observar un proceso modelado a partir de la notación estándar de modelado de procesos de negocio y también es posible identificar dos áreas adicionales dentro del interfaz de la herramienta: una zona donde se ven todas las entidades presentes en el modelo actual y una zona donde se observa una miniatura de todo el área dedicada al diseño del proceso.

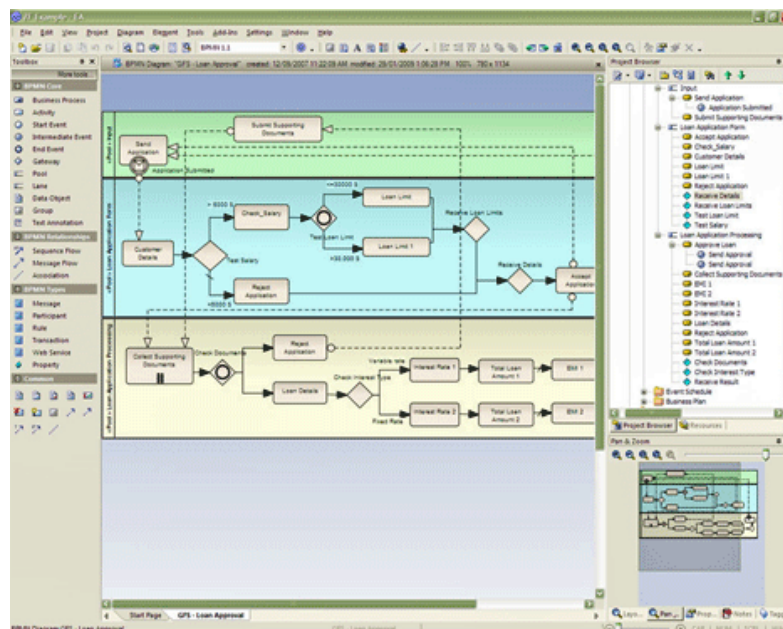


Figura 40: Interfaz de usuario de Enterprise Architect

### 4.2.3 Magic Draw

Magic Draw **[M. Draw]** es una herramienta de modelado desarrollada por la empresa No Magic Inc. Ofrece soporte de manera nativa para BPMN 2.0 así como una opción de exportar los modelos realizados a XPD 2.2, el lenguaje XML de definición de procesos de negocio. Además, permite a los usuarios transformar los diagramas de actividad de UML a modelos BPMN y viceversa.

En la imagen incluida a continuación se muestra el interfaz gráfico de la herramienta Magic Draw albergando un proceso de negocio en la zona de diseño. En dicha figura es posible observar como ofrece, al igual que en el caso de la herramienta Enterprise Architect, las zonas que albergan las entidades del modelo actual y la vista general en miniatura del proceso que se está modelando. Sin embargo, la disposición de estas zonas

en el interfaz es diametralmente opuesta a la de la herramienta de la compañía Sparx Systems, ya que se encuentran en el margen izquierdo del interfaz.

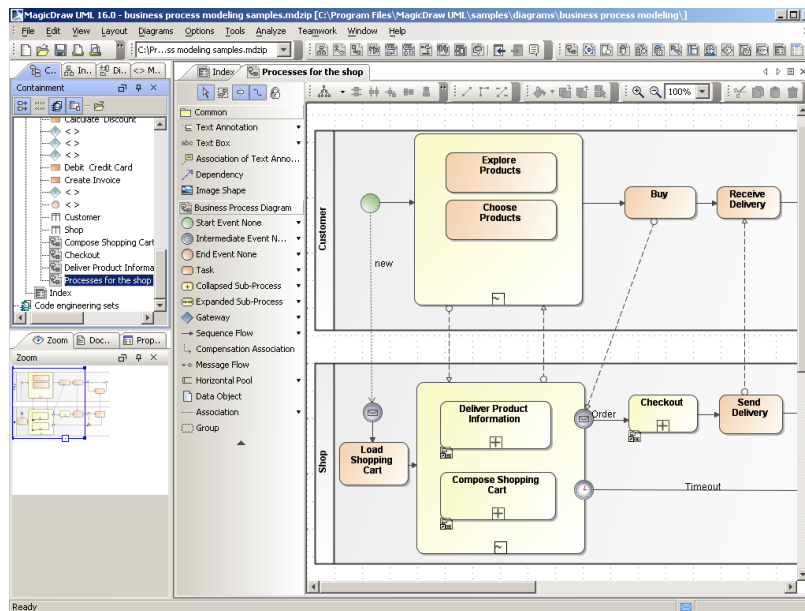


Figura 41: Interfaz gráfico de Magic Draw

#### 4.2.4 Altova UModel

La empresa Altova ofrece una herramienta denominada UModel [UModel] que permite al usuario modelar procesos de negocio a través de dos notaciones de las estudiadas anteriormente en este documento: UML Activity Diagrams y BPMN. Con respecto a BPMN, esta herramienta permite utilizar tanto la versión 1.0 como la 2.0 al tiempo que ofrece al usuario la posibilidad de convertir diagramas antiguos a la versión actual.

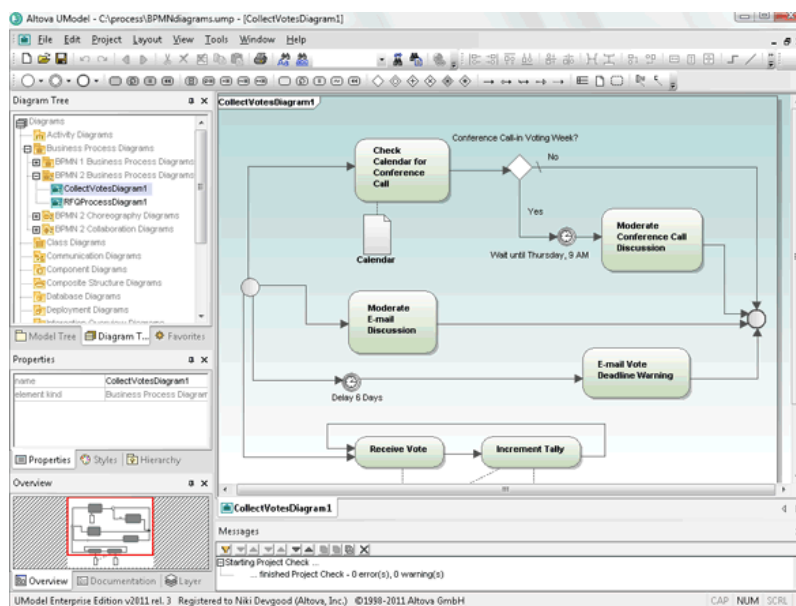


Figura 42: Interfaz de Altova UModel

### 4.2.5 Yaoqiang BPMN Editor

Yaoqiang BPMN Editor [**BPMN Editor**] es un editor de diagramas BPMN que sigue el estándar de modelado de procesos en su versión 2.0. Está desarrollado bajo licencia GNU General Public License versión 3.0. Una de las características que incluye esta herramienta es la posibilidad de importar archivos de la herramienta Microsoft Visio y convertirlos a diagramas BPMN.

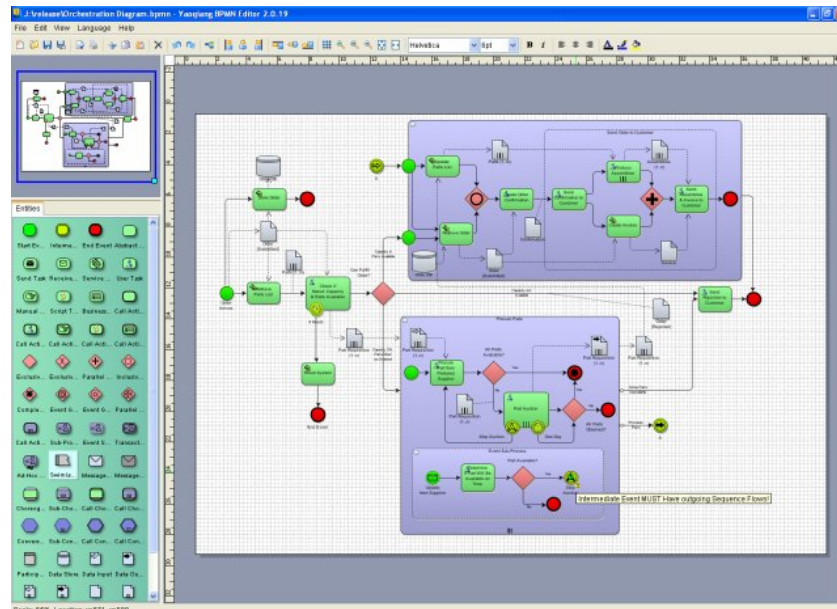


Figura 43: Yaoqiang BPMN Editor mostrando un proceso de negocio

### 4.2.6 Business Process Visual Architect

Business Process Visual Architect [**BP-VA**], también conocido por sus siglas BP-VA, es una herramienta de modelado de procesos de negocio desarrollada por la empresa Visual Paradigm. Aparte de permitir el modelado de procesos de negocio con BPMN esta herramienta incorpora un motor de definición de reglas de negocio así como un simulador para realizar simulaciones con los distintos procesos y generar tablas con los resultados de la simulación. Si bien BP-VA no permite la ejecución de los procesos diseñados sí que habilita la transformación de los modelos a lenguajes de ejecución de procesos como BPEL, XPDL y JPDL con el fin de poder ejecutarlos bajo otra plataforma.

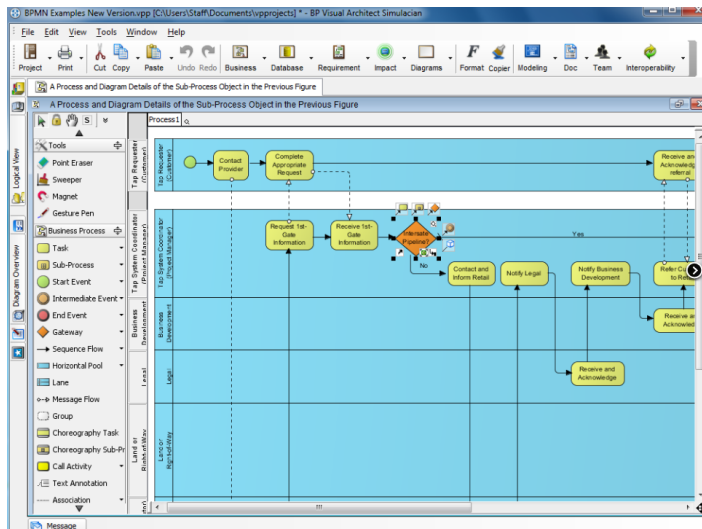


Figura 44: Herramienta BP-VA

#### 4.2.7 Facilis BPMN

Facilis BPMN [Facilis BPMN] es una herramienta de modelado de procesos de negocio de la compañía Statum. Ofrece soporte para las entidades de BPMN y permite exportar los procesos de negocio a formato XPDL para fomentar el intercambio de los procesos entre distintas herramientas.

Al igual que la mayor parte de las herramientas analizadas hasta el momento, Facilis BPMN cuenta con un interfaz gráfico dividido en zonas: la zona central está destinada al modelado de los procesos mientras que el área izquierda contiene las entidades que es posible utilizar en los modelos y la zona derecha las propiedades de la entidad que se encuentra seleccionada en el modelo de proceso actual.

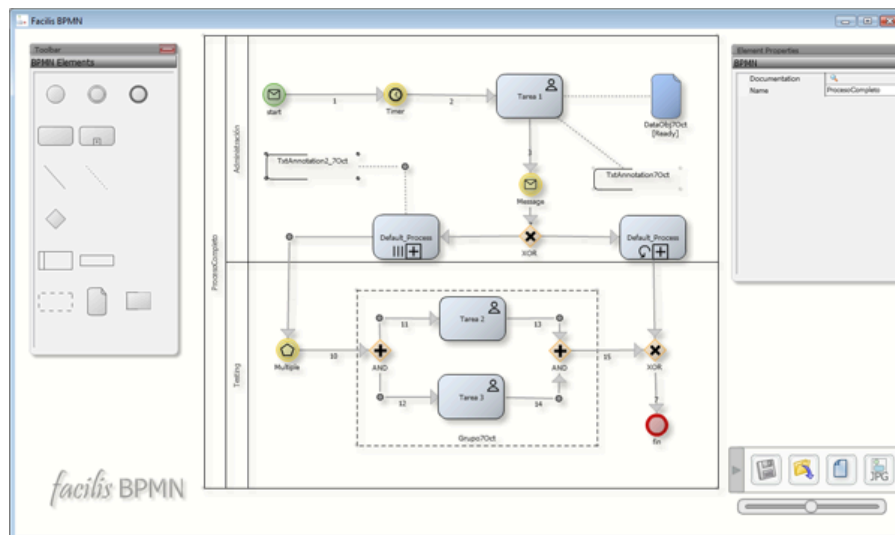


Figura 45: Interfaz de usuario de Facilis BPMN

### 4.2.8 Resumen

Todas las herramientas vistas en este recorrido permiten el modelado de los procesos de negocio a través del uso de algunas de las notaciones vistas en el [capítulo 3](#).

Sin embargo, teniendo en cuenta los objetivos marcados al comienzo de esta tesis y el escenario de trabajo asociado a esta investigación, donde se va a tratar con expertos del dominio sin conocimientos informáticos, no se considera que ninguna de las herramientas anteriores permita alcanzar las metas marcadas para este trabajo.

### **4.3 PLATAFORMAS CON SOPORTE PARA EL MODELADO Y LA EJECUCIÓN DE PROCESOS**

Una vez realizado el recorrido por algunas de las principales herramientas que permiten a los usuarios construir modelos de proceso a partir de notaciones gráficas de modelado, es necesario hacer un recorrido similar por las alternativas que ofrecen tanto modelado como ejecución de procesos de negocio.

El conjunto de plataformas de modelado y ejecución de procesos está formado por productos software que permiten al usuario tanto modelar sus procesos a través de cualquier notación gráfica como ejecutar los procesos a través de un motor destinado a tal efecto, haciendo que los procesos diseñados puedan ser utilizados como base para las aplicaciones informáticas que los gestionan.

El gran inconveniente de algunas de las plataformas o soluciones que aquí se presentan es el establecimiento de un funcionamiento de tipo caja negra, donde el usuario no tiene acceso, de forma directa, a modificar la biblioteca de funcionalidades incluida en la herramienta, por lo que los modelos estarían limitados a esta funcionalidad. Así, la adaptación de la herramienta de modelado al dominio del problema estaría limitada en estos casos a la funcionalidad cubierta en dicha biblioteca, limitándose la aplicación de la propuesta realizada en este trabajo a los dominios y funcionalidades incluidos en la herramienta.

Así, teniendo en cuenta las objetivos de este trabajo de investigación e intentando evitar la adopción de una solución que obligue a atarse a un producto concreto, se ha decidido tomar como mejores alternativas para la realización de este trabajo dos de las plataformas presentadas en esta sección: jBPM [**JBPM**] y Windows Workflow Foundation [**WWF**].

#### **4.3.1 ARIS Platform**

La empresa Software AG ha desarrollado ARIS Platform [**ARIS**] como su solución para la gestión de procesos de negocio. Este producto está compuesto por una serie de herramientas: ARIS Strategy, que permite el análisis y la optimización de los procesos; ARIS Design, para el diseño de los procesos de negocio; ARIS Implementation, para el despliegue de los procesos; y ARIS Controlling, para realizar mediciones sobre los procesos desplegados.

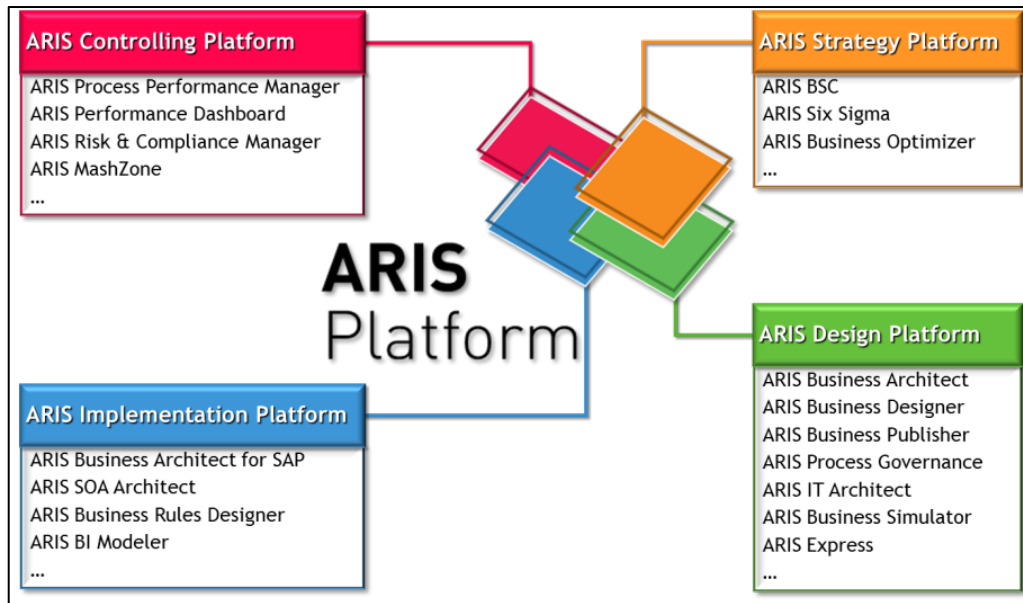


Figura 46: Esquema de componentes de ARIS Platform

La imagen anterior muestra, de forma resumida, el esquema de componentes de la propuesta de Software AG para el modelado y la ejecución de los procesos. En dicha figura se puede observar el nombre de los subcomponentes de cada una de las piezas centrales de ARIS Platform.

### 4.3.2 BPM Software Suite

BPM Software Suite [**BPM Soft. Suite**] es una suite de programas desarrollados por la empresa Ultimus para la gestión de procesos de negocio. En un acercamiento similar al ofrecido por ARIS Platform, BPM Software Suite se encuentra dividida en cuatro áreas asociadas a los procesos de negocio: modelado, automatización, gestión y optimización.

A diferencia de las herramientas vistas en la sección anterior BPM Software Suite habilita la ejecución de los procesos dentro del apartado de automatización, donde habilita la captura de información a través de formularios web y realiza el intercambio de información con las herramientas utilizadas en la organización.

### 4.3.3 Intalio|BPMS

Intalio|BPMS [**Intalio**] es un conjunto de herramientas para el modelado y la ejecución de procesos de negocio desarrollada por la empresa Intalio. Esta suite está construida tomando como base la plataforma Eclipse por lo que puede ser utilizada en los mismos entornos en los que esta es compatible.



### 4.3.4 Appian BPM Suite

Appian BPM Suite [**Appian BPM**] es un conjunto de herramientas para la gestión de procesos de negocio desarrollada por la empresa Appian. Algunos de los productos incluidos dentro de esta suite son: Tempo, Process, Rules, Portal, Forms y Analytics.

Dentro de los productos ofrecidos es Process el encargado de ofrecer soporte para el diseño, la ejecución, la gestión y la optimización de los procesos de negocio a través de una plataforma BPMN.

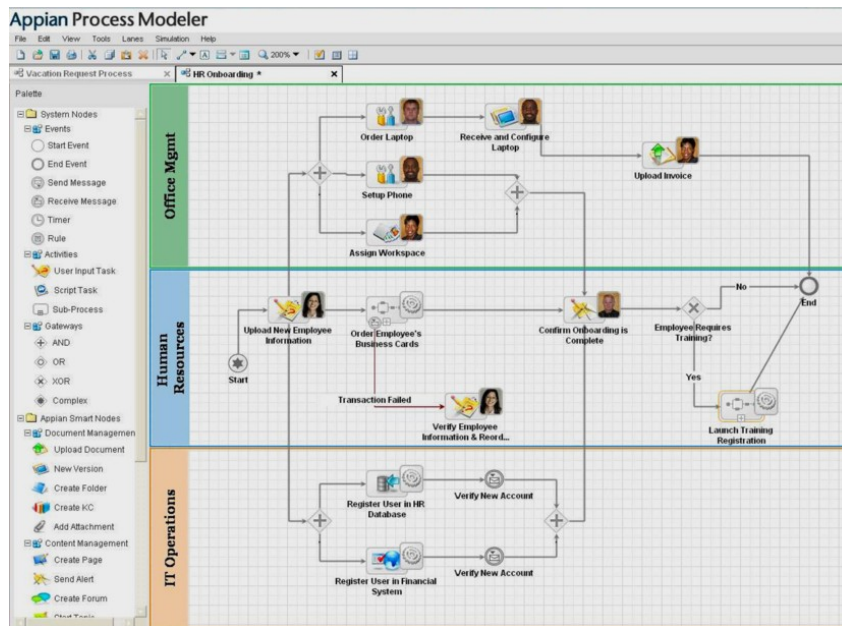


Figura 47: Interfaz de Appian Process Modeler

### 4.3.5 Bonita Open Solution

Bonita Open Solution [**Bonita Open Sol.**] es una solución de software libre para la gestión de procesos de negocio. Este conjunto de aplicaciones se basa en tres secciones: Bonita Studio, para el diseño de los procesos de negocio a partir de notaciones gráficas; un motor para BPM y Workflow, que permite la ejecución de los procesos; y un entorno de generación de interfaces gráficas para la creación de aplicaciones que ejecuten los procesos diseñados.



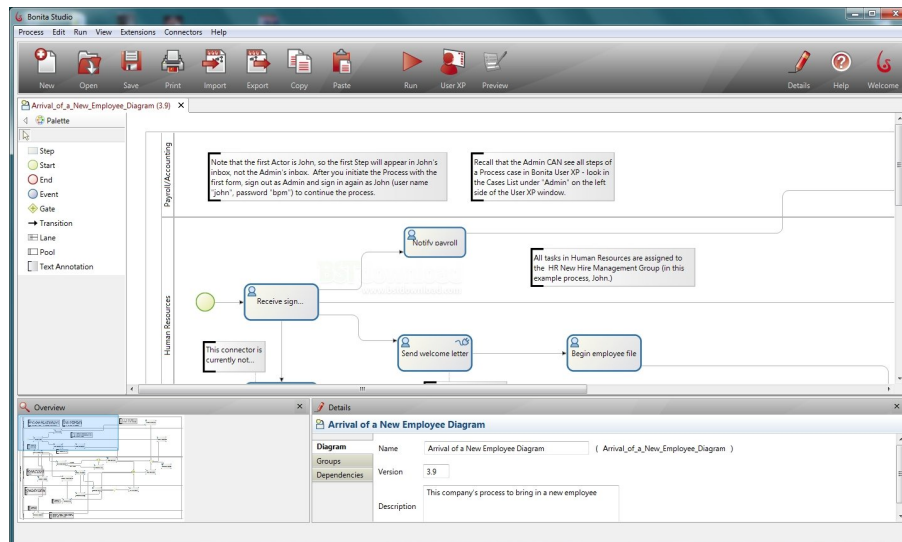


Figura 48: Interfaz de usuario de Bonita Studio

Como nota diferenciadora de las otras herramientas vistas hasta el momento está el hecho de que se proporcionan mecanismos de conexión de las aplicaciones desarrolladas con esta solución a dos tipos de sistemas: aplicaciones comerciales (Paypal, Google y Amazon entre otros) y bases de datos open-source (por ejemplo: SQL, SQLite y MySQL).

#### 4.3.6 jBPM: Java Business Process Modeling

jBPM [jBPM] es una aproximación a la disciplina BPM que ha sido promovida por la JBoss Community [JBoss]. Esta herramienta, basada en el software libre, se encuadra dentro de la plataforma Java si bien no posee una integración total dentro de la misma (su utilización se realiza a través de plugins en los entornos de desarrollo). Es necesario señalar también que la notación gráfica utilizada en las primeras versiones es muy cercana, si no idéntica, a la propuesta en BPMN y actualmente existe una última versión que utiliza BPMN 2.0.

La utilización de esta herramienta permite disponer de una serie de actividades prediseñadas al tiempo que ofrece la posibilidad al usuario de definir actividades propias que hagan tareas específicas del proceso que se pretende modelar. Sin embargo, la herramienta de modelado se encuentra dentro del entorno de desarrollo lo que la hace complicada de utilizar para todas aquellas personas ajenas al mundo de la informática y hace que el nivel de abstracción del usuario descienda.

##### 4.3.6.1 Entorno de trabajo

El entorno de trabajo principal de jBPM se encuentra enmarcado dentro del entorno de desarrollo Eclipse, si bien es posible crear un entorno independiente. La imagen que se incluye a continuación muestra una captura de pantalla del entorno de trabajo que se puede obtener dentro de Eclipse.

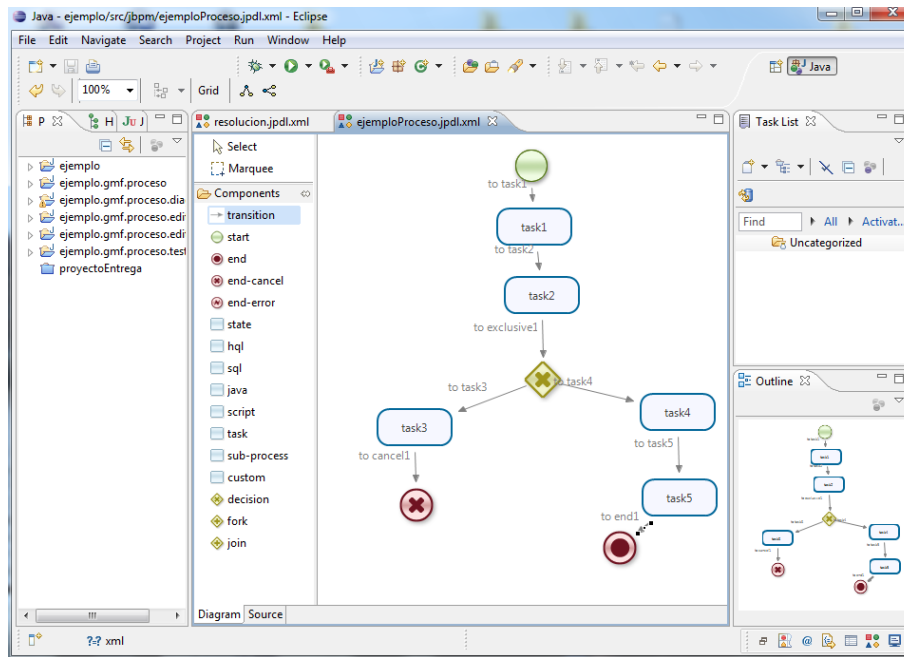


Figura 49: Entorno de trabajo de jBPM dentro de Eclipse

Tal y como se observa en la captura anterior el entorno de trabajo se encuentra dividido en tres secciones: la parte izquierda representa la estructura de directorios del proyecto en el que se está trabajando, la parte central representa el área de trabajo donde se realiza el modelado de los procesos y la parte derecha está constituida por una serie de paneles con propiedades.

La parte más importante de este entorno de trabajo es la parte central, donde el usuario puede modelar sus procesos. La estructuración de esta sección es sencilla, disponiendo únicamente de una paleta de herramientas situada a la izquierda y dejando el resto del espacio a la zona de diseño, siendo ambas zonas delimitadas por un separador. Para utilizar los componentes de la paleta en la zona de diseño el usuario debe seleccionar uno a uno los componentes que desea utilizar haciendo clic sobre ellos y después hacer clic sobre la zona de diseño para depositarlos allí; una vez que los elementos están ya sobre la zona de diseño el usuario puede cambiar el nombre de los mismos y moverlos libremente por la zona.

Observando la captura anterior se puede comprobar que en la parte inferior de la misma existen dos pestañas: *Diagram* y *Source*. La vista que se muestra en la captura es la correspondiente a la pestaña *Diagram* mientras que seleccionando la pestaña *Source* el usuario tiene acceso a ver y modificar el fichero XML asociado al proceso que se está diseñando. Es necesario señalar que la utilización de algunos de los componentes de la paleta de herramientas conlleva la necesidad de editar el contenido de este fichero a mano, siendo esto un inconveniente para las características del proyecto que nos ocupa.

### 4.3.6.2 Ejemplos de procesos realizados con jBPM

A continuación se incluyen unas imágenes de los procesos tal y como fueron modelados utilizando el plugin de Eclipse para jBPM, siendo el lenguaje de modelado jPDL. Ambos modelados se corresponden con modelos generados por los expertos utilizando una herramienta de dibujo que luego fueron transportados a la herramienta jBPM.

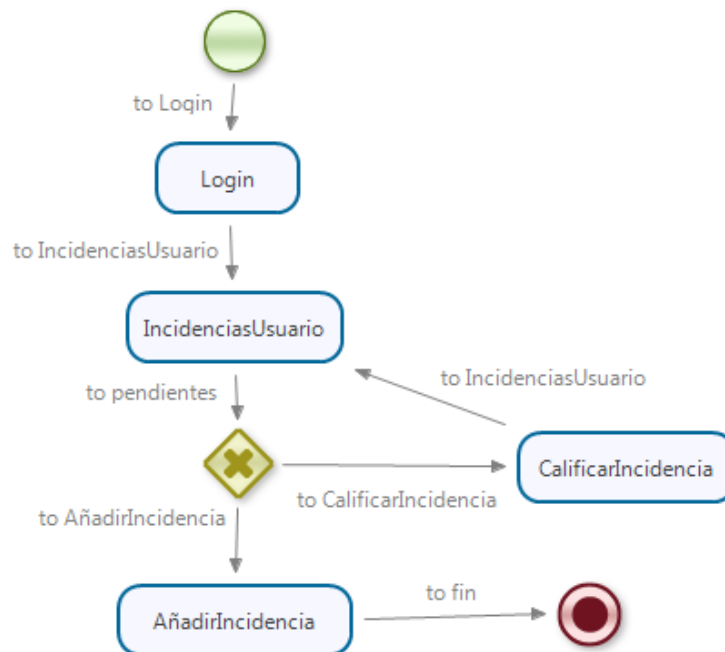


Figura 50: Proceso de inserción de incidencias en jBPM

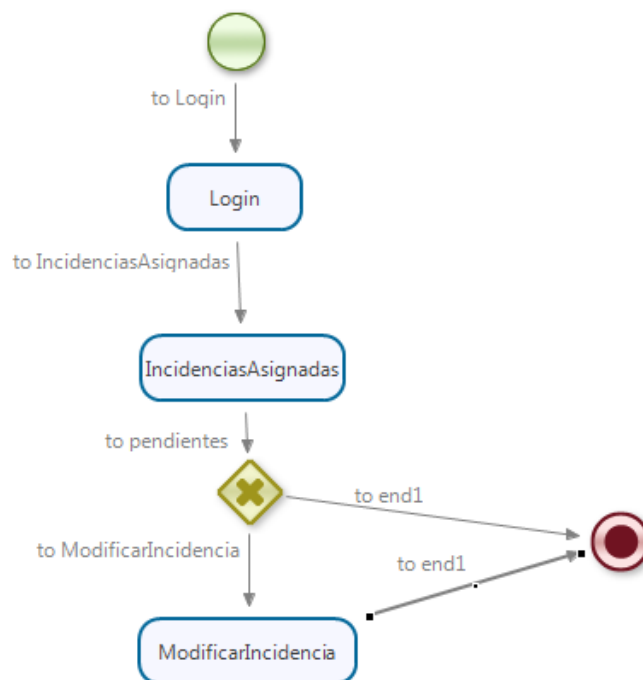


Figura 51: Proceso de resolución de incidencias en jBPM

#### *4.3.6.3 Ventajas de jBPM*

- La primera ventaja, y la más obvia, de jBPM es su gratuidad. Debido a estar sujeta a los términos del software libre, la utilización de jBPM para el desarrollo de aplicaciones utilizando modelos no tiene coste alguno.
- La cercanía de la representación gráfica de las entidades a la representación propuesta por el estándar, lo que hace que la comprensión de los modelados sea inmediata para aquellas personas que conozcan BPMN.
- La inclusión de soporte para el estándar BPMN 2.0 en las últimas versiones

#### *4.3.6.4 Inconvenientes de jBPM*

- Dificultad de creación de una herramienta de modelado para los expertos del dominio que sea independiente del entorno de trabajo disponible en la herramienta Eclipse.
- Bajo nivel de abstracción que experimenta el usuario, debido a la necesidad de realizar cambios a nivel de código para configurar algunas de las entidades. En algunos casos es necesario realizar cambios a nivel del fichero XML para configurar el modelo de forma correcta.
- Necesidad de utilizar software adicional para realizar el despliegue de las aplicaciones, como es el caso de los servidores de bases de datos y el servidor de aplicaciones. Esto requiere un proceso de configuración y orquestación de todas las herramientas para que el funcionamiento sea óptimo.

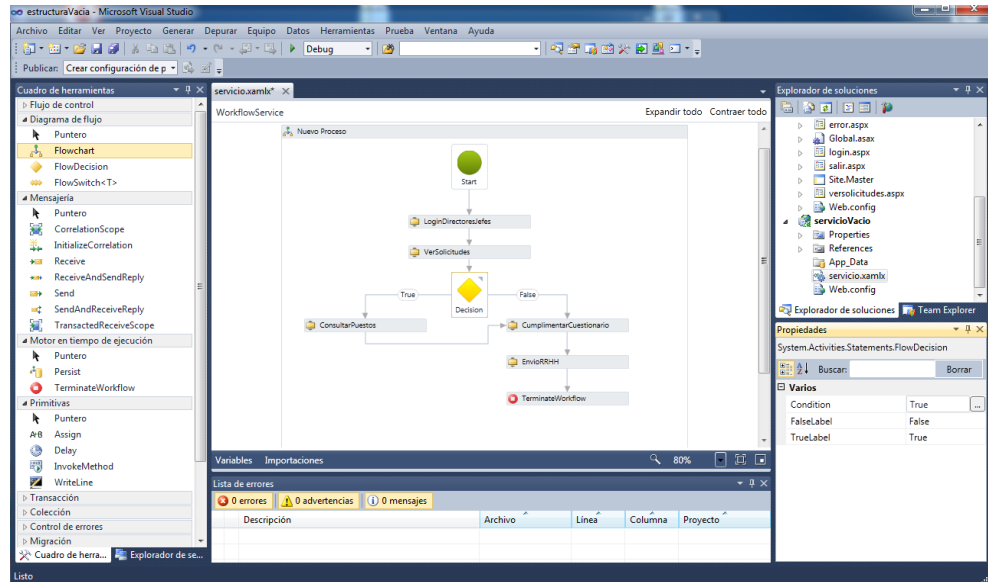
### **4.3.7 WWF: Windows Workflow Foundation**

Windows Workflow Foundation [WWF] es la aproximación de Microsoft a la disciplina BPM. Se encuentra integrado completamente dentro de la plataforma .NET desde el año 2006 y sigue en proceso de mejora para continuar incluyéndola en versiones posteriores de la plataforma. A diferencia de la herramienta anterior la representación gráfica de los elementos no se acerca tanto al estándar propuesto por BPMN.

Al igual que en el caso de jBPM, la herramienta ofrece la posibilidad de utilizar dos tipos de actividades: unas prediseñadas y otras definibles por el usuario para acometer tareas específicas del proceso a modelar. A pesar de tener este punto en común con jBPM, existe una diferencia muy notable en otra característica: WWF ofrece la posibilidad de utilizar una herramienta de modelado ajena al entorno de desarrollo y configurable para cada tipo de usuario; de esta forma se consigue adecuar el entorno de modelado a las necesidades del experto al tiempo que el nivel de abstracción aumenta.

### 4.3.7.1 Entorno de trabajo

La imagen incluida acto seguido de estas líneas es una captura del entorno de trabajo de Windows Workflow Foundation, que se encuentra incluido dentro del entorno de desarrollo integrado Visual Studio 2010.



**Figura 52: Entorno de trabajo de Windows Workflow Foundation**

De manera similar a lo que ocurriera con el entorno de trabajo de jBPM, el entorno de WWF se encuentra dividido en tres secciones; sin embargo es necesario establecer que la sección central dispone de un área de notificación de errores y que la colocación de las distintas secciones difiere con respecto al caso anterior.

La sección izquierda es la paleta de herramientas que contiene las entidades que pueden ser utilizadas en el modelado de los procesos; esta área es independiente en el entorno de trabajo de WWF y contiene un número mayor de elementos que en el caso anterior, existiendo incluso la posibilidad de añadir a esta sección las entidades personalizadas que se pueden definir (tal y como se explicará más adelante). La sección central está dividida en dos: el área de mayor tamaño se corresponde con la zona de modelado del proceso mientras que la zona inferior contiene el área de notificación de errores y avisos; a diferencia del caso anterior, la utilización de las entidades de la paleta de herramientas se hace a través de la selección de la entidad en la paleta y el arrastre de la misma a la zona de diseño. La zona de la derecha se encuentra distribuida en dos secciones: la sección superior muestra la estructura de ficheros del proyecto o solución que está siendo utilizado y la parte inferior es la zona donde se muestran las propiedades de los distintos elementos; a través de la utilización de esta sección inferior es posible realizar cambios en el modelo que se está diseñando sin necesidad de tocar el fichero XAMLX asociado al proceso, siendo esto un punto de diferencia con respecto al entorno de trabajo de jBPM.

La otra gran diferencia de este entorno de trabajo es la posibilidad de crear una herramienta de modelado completamente ajena al entorno de desarrollo Visual Studio 2010. Esta herramienta permite modelar los procesos de forma idéntica a lo ofrecido a través del entorno presentado anteriormente y ofrece unas posibilidades de configuración

que posibilitan la creación de una herramienta de modelado sencilla, ideal para usuarios expertos de los procesos de negocio pero que dispongan de pocos conocimientos informáticos.

#### 4.3.7.2 Ejemplos de procesos realizados con WWF

A continuación se incluyen dos imágenes de los procesos de gestión de incidencias, anteriormente mostrados con la propuesta de jBPM, modelados con la plataforma Windows Workflow Foundation. Tal y como se puede observar en las imágenes, la forma en la que WWF representa alguna de las entidades difiere de la propuesta de la plataforma anterior; sin embargo, los modelos de proceso resultantes tienen correspondencia directa con los mostrados anteriormente.

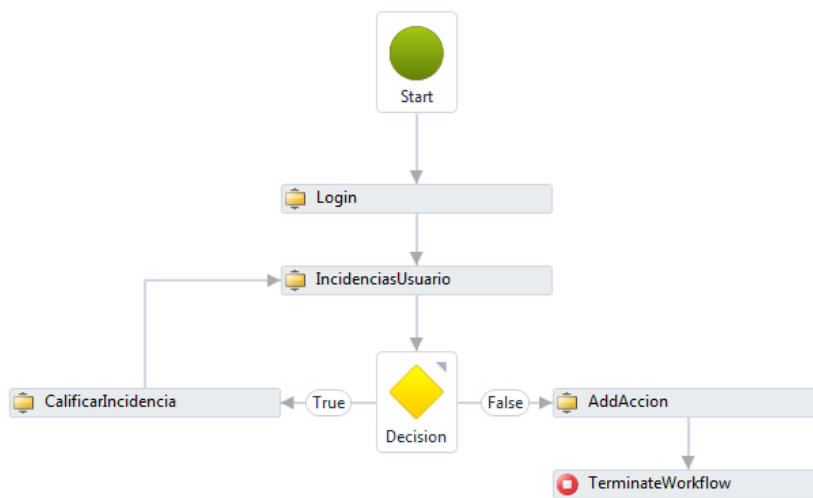


Figura 53: Proceso de inserción de incidencias según WWF

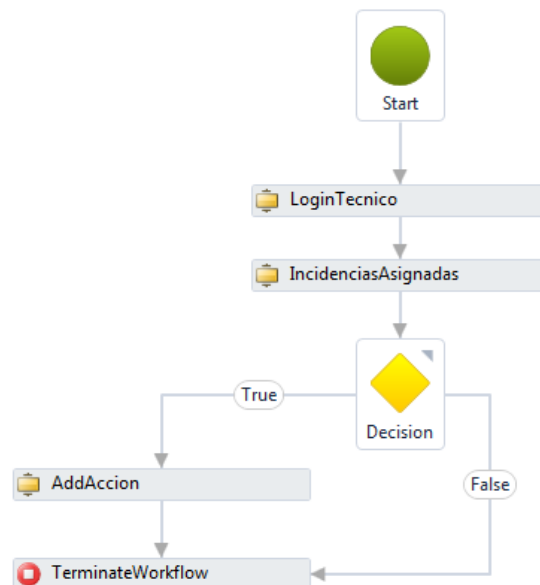


Figura 54: Proceso de resolución de incidencias según WWF

#### 4.3.7.3 Ventajas de Windows Workflow Foundation

- Integración total dentro de la plataforma .NET y de las herramientas disponibles en el sistema operativo Windows.
- Documentación y soporte oficial.
- Posibilidad de construir una herramienta de modelado altamente configurable para dar soporte al modelado de los procesos, con el consiguiente aumento del nivel de abstracción de los usuarios.
- La reutilización del código que se puede conseguir para las actividades que sean programadas.
- La integración sencilla con otras aplicaciones y herramientas de Microsoft, como los servidores de aplicaciones y de bases de datos. Esto reduce la configuración necesaria para la puesta en marcha.

#### 4.3.7.4 Inconvenientes de Windows Workflow Foundation

- Coste de la licencia necesaria para poder utilizar la plataforma.
- Relación con los estándares tanto en el apartado de representación gráfica de las entidades como en el de los formatos de almacenamiento de los modelos.

### 4.3.8 Resumen

Las plataformas con soporte para el modelado y la ejecución de procesos de negocio presentadas en los apartados anteriores ofrecen un soporte completo para la iniciativa BPM. Sin embargo, las plataformas jBPM y WWF son las que reúnen, en mayor medida, las condiciones necesarias para la consecución de los objetivos marcados en esta tesis. Las alternativas de carácter comercial mencionadas anteriormente disponen de una capacidad parecida a la hora de modelar y ejecutar modelos. Sin embargo su orientación comercial hace que estas alternativas no sean adecuadas para la realización de un trabajo como el que se pretende desarrollar en esta investigación.

En concreto, tal y como se mencionó anteriormente, la principal limitación que se puede identificar en estas plataformas es su carácter de caja negra, ya que las entidades que comercializan este tipo de herramientas y plataformas incluyen unos desarrollos cerrados para responder a las necesidades de su negocio. Es por ello que la inclusión de funcionalidades personalizadas para responder a las necesidades de cada dominio requiere un proceso costoso de definición e implementación. Esta circunstancia provoca que no sea factible realizar desarrollos personalizados de forma sencilla y dificulta, de manera importante, el proceso de construcción de los prototipos que permitan realizar las pruebas necesarias para extraer las conclusiones de este trabajo de investigación.

El **capítulo 8** de este documento, donde se realiza una descripción de los prototipos que acompañan a este trabajo, incluye un apartado en el que se razona la selección de la plataforma escogida.



#### 4.4 CONCLUSIONES

En la actualidad existen varias alternativas en lo que a herramientas y plataformas con soporte para BPM se refiere. A pesar del número de alternativas existentes, solamente se consideran como alternativas viables para la realización del trabajo las dos últimas alternativas estudiadas: jBPM y WWF.

A pesar de que las características del resto de herramientas y plataformas no se corresponden con lo requerido para la realización de este trabajo de investigación, la funcionalidad y disposición de los controles que ofrecen las herramientas de modelado estudiadas serán tomadas en consideración a la hora de establecer las características de los prototipos.

Una vez vistas las notaciones y las herramientas más destacadas para el modelado de procesos de negocio se realizará una introducción a la ingeniería dirigida por modelos o MDE [Kent 2002], para presentar las circunstancias que promueven el uso de modelos para la construcción de artefactos software.



# CAPÍTULO 5

## MDE: INGENIERÍA DIRIGIDA POR MODELOS

---

---

### 5.1 INTRODUCCIÓN

La ingeniería dirigida por modelos, también conocida por sus siglas en inglés MDE, es una aproximación al desarrollo de software propuesta a comienzos de los años 2000 [Kent 2002]. Dicha propuesta vino derivada por la falta de resultados obtenidos por las metodologías creadas para la reducción de los problemas que surgen en el desarrollo de un producto software; a estos problemas se los conoce como crisis del software, un término introducido a principios de los años 1970 [Dijkstra 1972].

En la Universidad de Oviedo existe un grupo de investigación centrado en la ingeniería dirigida por modelos [MDE-RG], donde se han realizado investigaciones importantes dentro de este campo. Entre ellas destacan algunos trabajos como Talismán MDE [García-Díaz et al. 2010] o MCTest [García-Díaz et al. 2011].

Durante el desarrollo de este capítulo se realiza una introducción a la ingeniería dirigida por modelos a través de: los términos o conceptos que introduce, sus ámbitos de aplicación y las iniciativas que promueven el desarrollo de software a partir de sus directrices.

## 5.2 PROBLEMAS DEL DESARROLLO DE SOFTWARE TRADICIONAL

La existencia de problemas en el desarrollo de las aplicaciones empresariales es una situación surgida en la segunda mitad del siglo XX, identificada por Dijkstra en uno de sus trabajos [Dijkstra 1972]. A continuación se incluye una lista que representa algunos de los problemas existentes en el proceso de desarrollo de aplicaciones que se mantienen vigentes desde la referencia temporal anterior:

- Nivel muy bajo de calidad en el desarrollo en líneas generales.
- El software no cumple con las especificaciones requeridas y su funcionalidad no es adecuada.
- Incumplimiento de la planificación prevista, tanto a nivel temporal como a nivel presupuestario.
- Incremento del coste de mantenimiento a medida que el proyecto crece.

Algunas de las posibles causas relacionadas con la lista de problemas mostrada anteriormente son las identificadas en uno de los trabajos de Greenfield [Greenfield et al. 2004]:

- Desarrollo de software aislado. No se tienen en cuenta desarrollos anteriores de similares características que pudieran tomarse como ejemplo para el desarrollo actual.
- Desarrollo de software monolítico, con componentes muy interconectados.
- Utilización de lenguajes de bajo nivel que, si bien son mucho más flexibles, reducen el nivel de productividad del desarrollo porque hacen que el desarrollador tenga que encargarse de un número elevado de aspectos durante el desarrollo.
- Debido al grado de juventud de la informática, los procesos de desarrollo de software no son tan maduros como otros procesos (construcción, automóvil, etc.).
- El alto nivel de demanda de software existente en la actualidad, que provoca que el proceso de desarrollo de un software concreto se deba abordar de la forma más rápida posible.

### 5.2.1 Necesidad de automatización del desarrollo de software

La automatización del proceso de desarrollo del software permite evitar los errores o problemas mencionados anteriormente, además de otros problemas que no fueron incluidos en el resumen anterior. Si bien se han producido una serie de avances a partir de la aparición de patrones de diseño, especificaciones, estándares y lenguajes de programación con mayor nivel de abstracción, la automatización del desarrollo de software no se ha producido de manera completa. Los beneficios obtenidos a partir de los avances mencionados incluyen la automatización parcial del desarrollo de software, el descubrimiento de la mejor forma para la solución de los problemas con los que trabajan

de forma habitual los desarrolladores y la homogeneidad en la realización de algunas tareas; esta última característica permite alcanzar un mejor nivel de mantenimiento e interoperabilidad entre las aplicaciones.

Aún viniendo de lejos, los problemas descritos anteriormente no parecen ser, aparentemente, demasiado complejos. El origen de un proyecto software viene determinado por la aparición de un problema en el espacio de trabajo del cliente para el que se quiere buscar una solución. La forma tradicional de abordar la resolución del problema es la descripción del mismo y de las necesidades para solucionarlo por parte del cliente y el traslado de dichos detalles a un lenguaje no formal por parte del personal técnico encargado de recogerlas. Sin embargo, la forma más conveniente para abordar una situación de este tipo sería la utilización de un lenguaje formal para recoger las especificaciones indicadas por el cliente, de tal manera que esta especificación formal permitiera automatizar, en la medida de lo posible, el desarrollo. La especificación formal se realizaría a través de la utilización de modelos software.

### **5.2.2 Nivel de abstracción del proceso de desarrollo de software**

El desarrollo de software ha experimentado cambios según los lenguajes de programación predominantes en cada momento y el nivel de abstracción de dichos lenguajes. Los tipos de lenguajes de programación que han sido utilizados para el desarrollo de software y sus características son los siguientes:

- Primera generación o lenguaje máquina. Utilizados durante los primeros desarrollos de software, estos lenguajes requerían que el programador introdujera directamente los bits (unos y ceros lógicos) para crear su software. Debido a esta circunstancia, el nivel de complejidad existente en los desarrollos con lenguaje máquina era muy elevado.
- Segunda generación o lenguaje ensamblador. Estos lenguajes supusieron la elevación del nivel de abstracción con respecto a la primera generación de lenguajes. Esta circunstancia trajo consigo la mejor del software tanto a nivel cualitativo como a nivel cuantitativo.
- Tercera generación o lenguajes procedimentales. La principal característica de estos lenguajes es su similitud con la forma de comunicación humana, por lo que son mucho más fáciles de utilizar que los lenguajes mencionados anteriormente. Sin embargo, suponen una pérdida considerable de rendimiento, sobre todo en la rapidez y la utilización de memoria. Algunos de los lenguajes más conocidos de este tipo son C, Pascal y Cobol.
- Cuarta generación o lenguajes orientados a objetos. Estos lenguajes son los que utilizan por vez primera los objetos, a través de las clases y sus instancias, apareciendo en este momento términos como la encapsulación, el polimorfismo y la herencia. La productividad en el desarrollo del software se ve incrementada debido a la reutilización de código y la similitud de la sintaxis con el lenguaje humano. C++, C# o Java son algunos de los lenguajes de este tipo.

- Quinta generación o lenguajes orientados a aspectos [Elrad et al. 2001]. Existe cierta discrepancia a la hora de aceptar esta categoría como válida debido a la falta de acuerdo sobre los lenguajes que pertenecen a este grupo. La idea relacionada con la programación orientada a aspectos es la de otorgar a la aplicación una modularización adecuada e introducir una mejora en la separación de conceptos dentro de esta; por ello, el objetivo principal es la separación de funcionalidades dentro del código de la aplicación. AspectJ y PHPAspect, extensiones de los lenguajes Java y PHP, son dos ejemplos de lenguajes de este tipo.

Tomando como referencia el concepto de nivel de abstracción, la clasificación anterior podría reformularse de la siguiente manera:

- **Lenguajes de bajo nivel**, que incluyen tanto al lenguaje máquina como al lenguaje ensamblador. Al trabajar directamente con el hardware existente en el equipo, están muy próximos al funcionamiento del ordenador.
- **Lenguajes de nivel medio**, que se sitúan entre los lenguajes de bajo nivel y los lenguajes de alto nivel. Se basan en la idea de permitir la realización de operaciones de alto y bajo nivel dentro del lenguaje. Por ejemplo, el lenguaje C permite realizar operaciones con los registros del sistema al tiempo que ofrece soporte para otras operaciones de alto nivel.
- **Lenguajes de alto nivel**. Los lenguajes de alto nivel se caracterizan por su independencia del hardware del ordenador, por lo que pueden ser migrados a otra máquina de forma sencilla a través del uso de traductores. Así, el uso de estos lenguajes abstrae al usuario de los detalles del hardware con el que está trabajando en cada momento. Algunos de los lenguajes más utilizados hoy en día, como C# o Java, son lenguajes de este tipo.

Según se produce un aumento del nivel de abstracción se consigue un aumento de la productividad, siendo esto debido a la utilización de términos más cercanos al conocimiento humano y al uso de funcionalidades más sofisticadas. El último gran avance en lo referente al aumento de la productividad y la calidad del desarrollo de productos software es la aparición de la ingeniería dirigida por modelos (Model-Driven Engineering o MDE, en inglés) [Kent 2002].

MDE está considerado un nuevo paradigma dentro del campo de la ingeniería del software. Está basado en la separación de la funcionalidad del sistema que se pretende desarrollar y la implementación de dicho sistema para una plataforma concreta; en otras palabras, se busca separar de forma clara el análisis y el diseño de la implementación. Para conseguir esto se han de utilizar diferentes modelos software.

Según Selic [Selic 2008], existen dos complejidades a la hora de realizar un desarrollo de software: la complejidad esencial, que no se puede evitar y es propia del problema que se pretende resolver, y la complejidad arbitraria, que se introduce según el tipo de herramienta y métodos que se utilizan para acometer el desarrollo. MDE está destinado a paliar la aparición de la complejidad arbitraria a través de un aumento del nivel de abstracción que permite evitar problemas léxicos, sintácticos y semánticos con los lenguajes de programación existentes en la actualidad y aquellos que puedan existir en el

futuro. Un aspecto clave de MDE es el resultado obtenido a través del uso de modelos para el aumento del nivel de abstracción, que permite conseguir una especificación del sistema por parte de los clientes a partir de un lenguaje formal.

### 5.2.2.1 Uso de modelos y diagramas para la construcción de software

Los modelos han sido utilizados a través de la historia para servir de representación y validación de un sistema cualquiera antes de acometer el proceso de realización o construcción del sistema al completo. Por ejemplo, unos planos de un edificio o la maqueta para la construcción de un puente son modelos utilizados de forma habitual. Según Selic [Selic 2003], un modelo debería poseer las siguientes características:

- **Coste bajo.** Al tratarse de representaciones anteriores a la construcción del sistema completo es lógico establecer que una de las características deseables de un modelo sea un coste muy inferior al del desarrollo completo del sistema. Al hablar de un coste bajo se hace referencia tanto al coste económico como al coste temporal de la creación del modelo.
- **Precisión.** Los modelos han de suponer la representación completa y precisa del sistema real al que representan para resultar útiles.
- **Comprensión.** Los modelos deben de ser fáciles de comprender y analizar por parte de las personas que los utilicen, ya que en caso contrario no servirían para simplificar la construcción del sistema al que representan.

Haciendo referencia más directa al mundo del desarrollo software normalmente se confunden los términos diagrama y modelo, utilizándose de forma indistinta. Sin embargo, la diferencia entre ambos es bastante notable.

- Un *modelo* es una abstracción de un sistema del mundo real que captura una vista, pudiendo existir varias vistas distintas dentro de un mismo sistema. Así, un modelo describe de forma detallada los elementos de un sistema desde su punto de vista.
- Un *diagrama* es una representación gráfica de una colección de elementos de modelado. De forma habitual, un diagrama se representa en forma de grafo. El tipo de diagrama más conocido es el diagrama de clases que se puede realizar con UML (Unified Modeling Language) [OMG 2007], que permite representar de forma gráfica los conceptos del modelo de clases de un sistema software.

El objetivo principal que establece MDE es la utilización de modelos para el desarrollo de software [Seidewitz 2003].

### 5.3 CICLO DE VIDA DEL DESARROLLO DE SOFTWARE

Se conoce como ciclo de vida del desarrollo del software al proceso de desarrollo del software desde su fase inicial hasta su fase final. Dicho periodo comprende las siguientes fases: obtención de requisitos, análisis, diseño, implementación, pruebas y despliegue.

Con la propuesta de MDE el ciclo de vida del software se modifica. Esto es debido al aumento del nivel de abstracción para generar las aplicaciones y a la automatización total o parcial de algunos de las fases mencionadas anteriormente. Un resumen gráfico de esta circunstancia puede observarse en la siguiente imagen, donde se muestra una comparación entre el ciclo de vida tradicional y el que se desprende de la propuesta de la ingeniería dirigida por modelos o MDE.

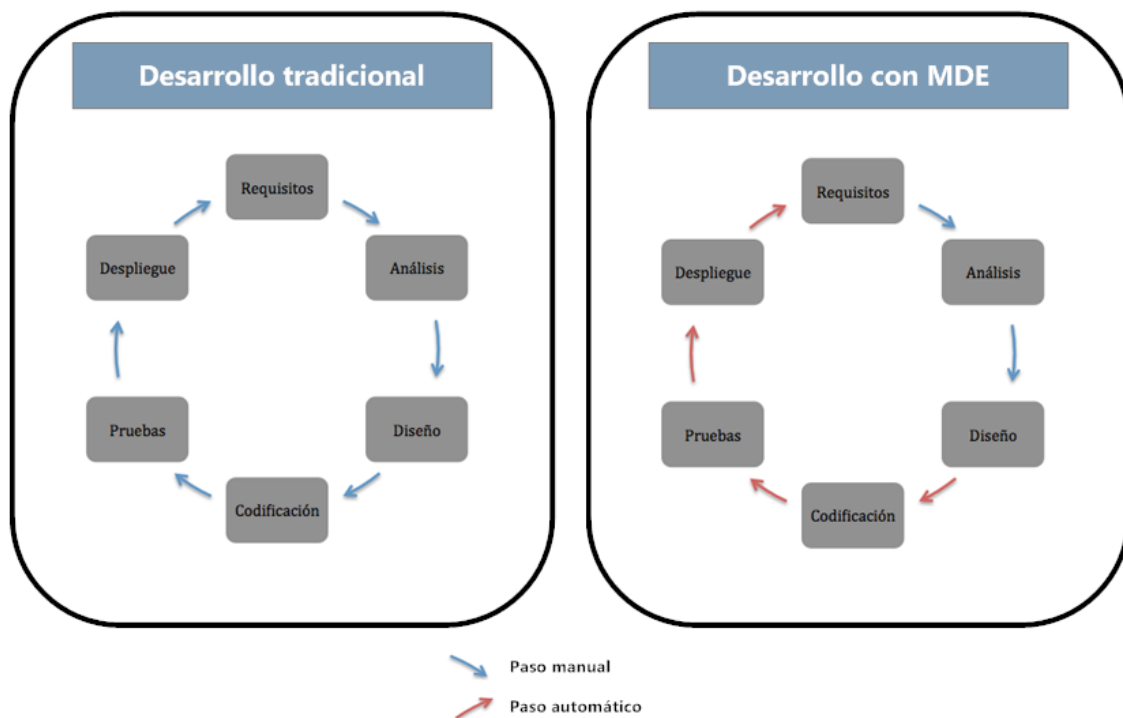


Figura 55: Comparativa de ciclos de vida

Tal y como se puede observar en dicha imagen el punto de inflexión es la fase de análisis de dicho ciclo, ya que a partir del artefacto resultante de la fase de análisis se puede automatizar la generación del código de la aplicación de forma total o parcial. Además, se puede considerar también que parte de las pruebas que serían necesarias durante un desarrollo tradicional ya no lo son y otras de las pruebas se podrían realizar de forma automática.



### 5.4 CONCEPTOS GENERALES DE LA INGENIERÍA DIRIGIDA POR MODELOS

La figura que se incluye a continuación, incluida dentro del trabajo de Völter y Stahl [Völter and Stahl 2006], representa los conceptos más importantes y básicos de MDE. Estos conceptos son independientes de cualquier iniciativa de MDE que se emplee (consultar apartado 5.4.1).

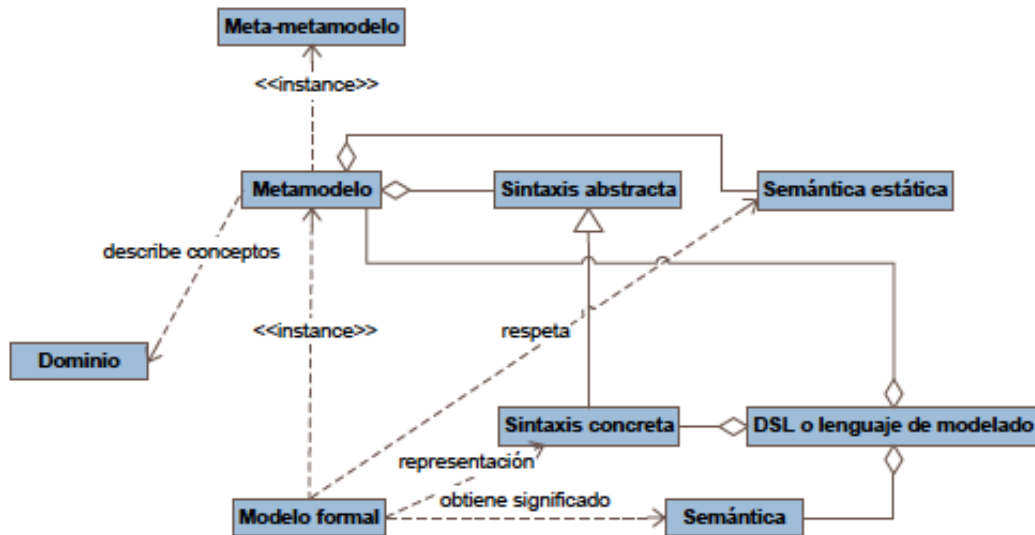


Figura 56: Diagrama de conceptos básicos de MDE

El punto inicial de MDE es siempre el **dominio**, que delimita el campo de conocimiento actual. Existen dos tipos de dominio: los tecnológicos, referentes a la tecnología de desarrollo de software, y los profesionales, que hacen referencia a los conceptos que van a ser parte de la aplicación.

El **metamodelo** sirve para describir, de una manera formal, los conceptos importantes del dominio sobre el que se trabaja. Es una parte fundamental a la hora de alcanzar la automatización del desarrollo de software.

La reutilización, interoperabilidad y portabilidad de los metamodelos dependen de la existencia de otro metamodelo en un nivel superior de abstracción. Este será el encargado de describir de forma única los conceptos que sirven para representar cualquier metamodelo de cualquier dominio. El **meta-meta-modelo** es el encargado de acometer esta tarea.

Los metamodelos tienen una **sintaxis abstracta** y una **sintaxis concreta**. La sintaxis abstracta está enfocada a los elementos a nivel conceptual y la sintaxis concreta está centrada en la representación de los conceptos. Así, la sintaxis abstracta de un metamodelo será invariable mientras que podrán existir varias sintaxis concretas para representar los mismos conceptos.

- La sintaxis abstracta de un lenguaje sirve para especificar su estructura: construcciones, propiedades y conectores. Es práctica común especificar alguno de estos detalles en el metamodelo, ya que la detección de errores en fases tempranas permite una resolución más sencilla de los mismos.

- La sintaxis concreta de un lenguaje es necesaria para especificar la notación que los usuarios van a utilizar. La situación ideal sería aquella en la que cada concepto del dominio y del lenguaje tuviera correspondencia con una representación en la notación.

La **semántica abstracta** de los metamodelos está basada en la sintaxis abstracta de éstos y su principal cometido es la realización de comprobaciones en los modelos a nivel semántico. De esta forma se puede asegurar que los modelos están contruidos de forma correcta.

Los **lenguajes específicos de dominio** [van Deursen 1997] [van Deursen et al. 2000], también conocidos como Domain-Specific Languages o DSLs, son lenguajes que permiten expresar los conceptos de un dominio específico. Un DSL está formado por uno o varios metamodelos, una o varias sintaxis concretas y una herramienta que ofrece soporte al lenguaje y facilita la usabilidad de éste. Un lenguaje de este tipo está diseñado para resolver un problema concreto dentro de un dominio específico y se lo considera el elemento principal de cualquier solución de dominio específico.

Tomando en consideración todos los elementos que han sido mencionados hasta este momento se puede hablar de modelos formales. Un **modelo formal** es el punto inicial desde el que se automatizan la transformación a entidades de menor nivel de abstracción. Estos modelos formales son instancias de los metamodelos y son representados a partir de una sintaxis concreta. Además, es necesario que respeten la semántica incluida en el metamodelo para que las construcciones que se realicen dentro del dominio sean coherentes.

La **semántica** de un lenguaje específico de dominio está relacionada con el significado de cada uno de los conceptos del modelo, por lo que al añadir un elemento al modelo lo que se está añadiendo en realidad es significado. De forma contraria a lo que ocurre con los lenguajes de propósito general, el uso de los lenguajes de dominio específico permite que los conceptos del lenguaje se mapeen directamente a los conceptos del dominio que se está modelando, eliminando la posibilidad de interpretaciones equivocadas.

#### 5.4.1 Iniciativas MDE

MDE se ha beneficiado de la aparición de varias iniciativas de cierta importancia que han hecho realidad sus directrices. Si bien es posible que los creadores de una iniciativa no hagan mención de otras iniciativas relacionadas, la utilización de conceptos similares aunque desde perspectivas distintas permite discernir aquellas iniciativas que establezcan como trabajar con MDE. Las iniciativas consideradas como más relevantes son las siguientes, si bien esta lista puede sufrir modificaciones al tratarse de un campo cambiante:

- Arquitectura dirigida por modelos, también conocida como Model-Driven Architecture o MDA. Esta iniciativa, propuesta por el OMG, ha adquirido un grado de importancia tan elevado que es frecuente que sea asociada directamente con MDE.

- Factorías software (*Software Factories* en inglés). A través del apoyo recibido por parte de Microsoft, las factorías software están aumentando su cuota en el desarrollo de software.
- Desarrollo de Software Dirigido por Modelos y centrado en la Arquitectura, más conocido por el acrónimo anglosajón AC-MDSD. AC-MDSD [Völter and Stahl 2006] es una iniciativa MDE centrada en la creación de un framework en el que se incluyen los conceptos arquitectónicos más importantes de una familia de software. Así, los modelos resultantes de la aplicación de esta iniciativa serían más sencillos, ya que únicamente incluirían aspectos relacionados con un dominio específico. Las aplicaciones creadas con el uso de esta iniciativa son aplicaciones completas, realizándose una transformación directa de los modelos a código mediante el uso de una plantilla.
- Programación generativa, también conocida como GP. La programación generativa [Czarnecki and Eisenecker 2000] es una iniciativa MDE que sus autores definen como “*un paradigma de la ingeniería del software basado en modelar familias de software de modo que dada una especificación de requisitos se pueda desarrollar, automáticamente y bajo demanda, un producto final o casi final ajustado a las necesidades y permitiendo la reusabilidad de componentes gracias a la configuración del conocimiento*”. Así, GP consigue generar aplicaciones completas a partir de componentes atómicos, que han sido optimizados para aspectos específicos gracias a la configuración del conocimiento.
- Computación Integrada con Modelos o MIC. MIC [Sztipanovits and Karsai 1997] es una iniciativa de la ingeniería dirigida por modelos surgida en el campo de trabajo de los sistemas distribuidos en tiempo real que propone la utilización de DSLs y varios modelos para especificar todos los aspectos que pueden estar contenidos en un sistema. El uso de esta iniciativa convierte a los modelos en el centro de todo el ciclo de vida de los sistemas, por lo que son necesarias las transformaciones entre los distintos modelos para poder realizar el análisis, las verificaciones o las simulaciones.
- Language-Oriented Programming o Programación Orientada al Lenguaje [Ward 1994]. El software Meta Programming System (MPS) permite definir lenguajes propios, a través de un metamodelo, integrados en el entorno de desarrollo de MPS. Al definir un lenguaje también se definen los siguientes aspectos: su editor, su compilador, las transformaciones que se realizan y el soporte para las operaciones de depuración.

## 5.5 EL METAMODELO: NEXO DE UNIÓN CON EL MODELADO DE PROCESOS

Una vez introducidos los conceptos generales de la ingeniería dirigida por modelos, mostrados de forma gráfica a través del diagrama de conceptos incluido en el [apartado 5.4](#), es necesario establecer la relación existente entre la ingeniería dirigida por modelos y el modelado de procesos.

A pesar de que todos los conceptos incluidos en dicho diagrama tienen su cuota de importancia a la hora de hablar de la ingeniería dirigida por modelos, uno de los conceptos sobresale por ser clave para este enfoque: el metamodelo. El metamodelo es un elemento fundamental dentro de la ingeniería dirigida por modelos porque es el que habilita la automatización del proceso de desarrollo del software. Esto es debido a que el metamodelo describe los conceptos del dominio sobre el que se está trabajando y permite crear una instancia de un modelo formal que representa los elementos de dicho dominio. A partir de este modelo formal es posible realizar las transformaciones con las que automatizar algunas de las fases del proceso de desarrollo de software.

Así, el metamodelo se convierte en el nexo de unión existente entre las notaciones o lenguajes de modelado de procesos de negocio y la ingeniería dirigida por modelos, permitiendo que los modelos de proceso construidos con las notaciones puedan ser transformados para conseguir la generación de artefactos software.

### 5.5.1 Los metamodelos de la notación estándar BPMN

BPMN [BPMN], la notación estándar de modelado promovida por el Object Management Group [OMG], ha visto evolucionar sus metamodelos a medida que ella misma iba cambiando para ajustarse a las necesidades de los expertos que la utilizaban. A continuación se realizará un recorrido por los metamodelos utilizados por las dos versiones que ha tenido.

#### 5.5.5.1 BPDM: el primer metamodelo para procesos de negocio

La primera versión de la notación de modelado BPMN no contó con la definición de un metamodelo propio. Sin embargo ésta se vio cumplimentada un tiempo después, entre los años 2007 y 2008, por la definición de un metamodelo denominado Business Process Definition Metamodel, al que también se puede hacer referencia a través de las siglas BPDM [BPDM].

El metamodelo BPDM, promovido también por el OMG, surge con la idea de proporcionar un lenguaje abstracto para la definición de procesos de negocio ejecutables, siendo independiente de la plataforma con respecto a los lenguajes de definición de los procesos de negocio. Los procesos de negocio que pueden definirse con este metamodelo pueden requerir la participación de personas o pueden ser totalmente automáticos. Además, existe la posibilidad de habilitar la colaboración de procesos considerados independientes entre sí, pertenecientes a distintas áreas de negocio o empresas.

Según los documentos donde se recoge el pliego de condiciones del metamodelo, proporcionados por el OMG [OMG], los objetivos que se pretendían conseguir con la definición de éste eran, entre otros, los siguientes:

- Crear un metamodelo común que permitiera la unificación de los distintos lenguajes de definición de procesos de negocio, tanto de carácter gráfico como de carácter textual.
- Definir un metamodelo que complemente a los metamodelos UML existentes en aquel momento, de tal forma que los procesos de negocio especificados puedan formar parte de especificaciones de sistemas completos. De esta forma se intentaba fomentar la consistencia y completitud.
- Introducir el uso de la coreografía como elemento de comunicación entre procesos, favoreciendo la colaboración a través de mecanismos de colaboración como los servicios web.

Además de todos los puntos mencionados anteriormente, la definición del metamodelo BPDM tenía como otro punto importante posibilitar el intercambio de ficheros entre distintas herramientas de modelado de procesos y también entre herramientas de modelado y entornos de ejecución. Este intercambio sería posible por la utilización de XMI, un estándar de intercambio de metadatos a través de ficheros XML propuesto también por el OMG.

A partir de las condiciones mencionadas anteriormente, se establece que el metamodelo BPDM se compone de un conjunto de modelos que puede observarse en la figura que se incluye a continuación en forma de diagrama de paquetes: el modelo de composición, el modelo de proceso o procedimiento, el modelo de actividad de proceso y el modelo de interacción de proceso. En la representación de cada modelo las entidades marcadas en verde hacen referencia a entidades existentes en otros modelos.

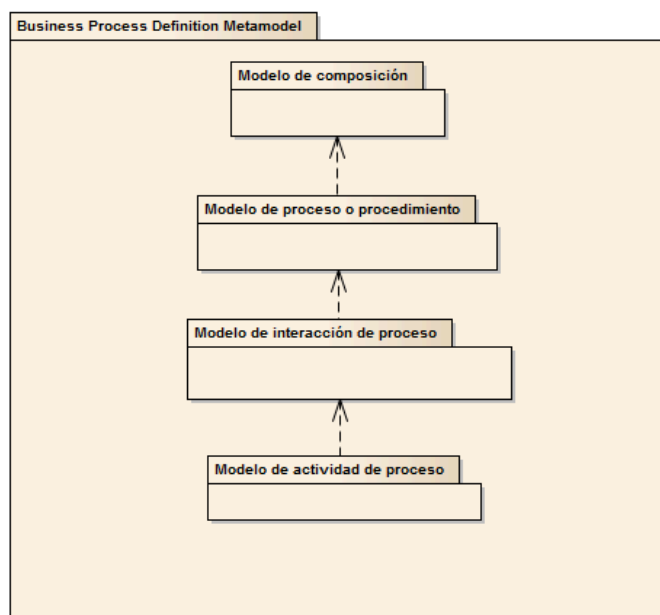


Figura 57: Diagrama de paquetes de BPDM

El modelo de composición es el que incluye los mecanismos de reutilización y descomposición de los elementos. Está basado en el uso de dos tipos de elementos: un elemento reutilizable denominado *Typeable Parts* (partes tipadas) que es reutilizado a través de objetos intermedios llamados *Parts* (partes).

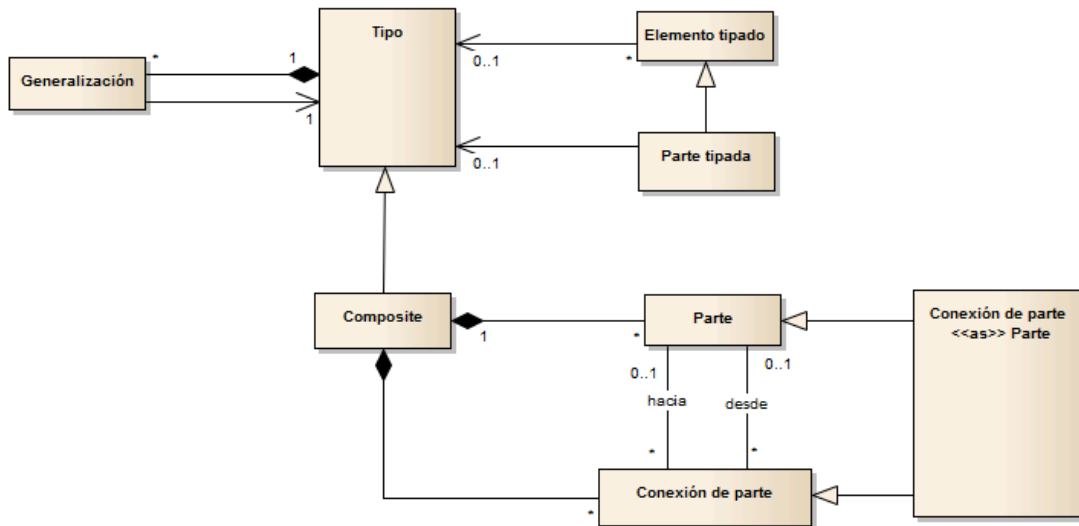


Figura 58: Modelo de composición de BPDM

El modelo de proceso o procedimiento describe los conceptos comunes utilizados en los modelos orientados a procesos. Describe los elementos de tipo *Parts* que componen un objeto representable en el proceso así como las conexiones que pueden utilizarse, llamadas *Part Connections*. Se puede observar como estos conceptos se acercan a términos de los procesos como actividades o conexiones.

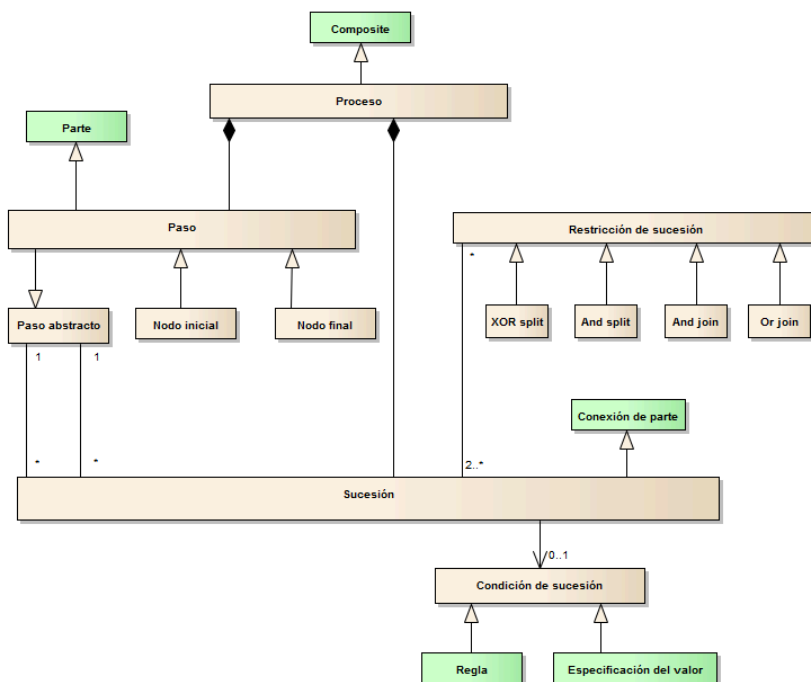


Figura 59: Modelo de proceso o procedimiento de BPDM

El modelo de actividad de proceso está centrado en la transformación de los procesos. Como este modelo dispone de un gran número de elementos se ha dividido en dos modelos (núcleo y flujos de acción) para facilitar la comprensión y visualización del mismo.

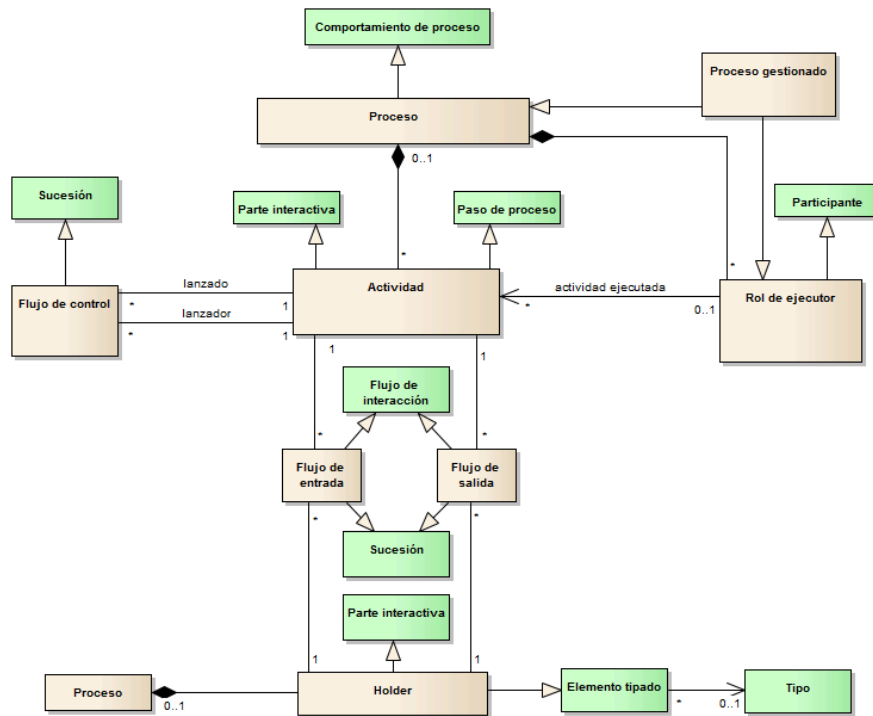


Figura 60: Núcleo del modelo de actividad de proceso de BPMN

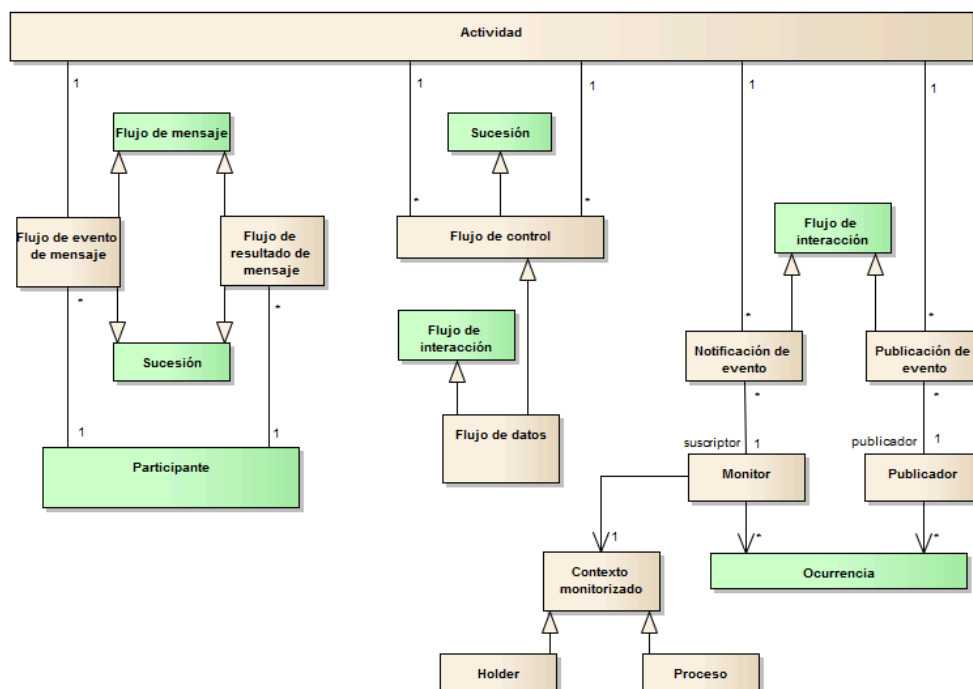


Figura 61: Flujos de acción del modelo de actividad de proceso de BPMN

El modelo de interacción de proceso es el encargado de recoger los elementos relacionados con las interacciones que pueden darse entre distintos procesos. Para facilitar la comprensión de este modelo se opta por dividir su explicación en tres secciones: paso de proceso, flujo de mensaje e interacción.

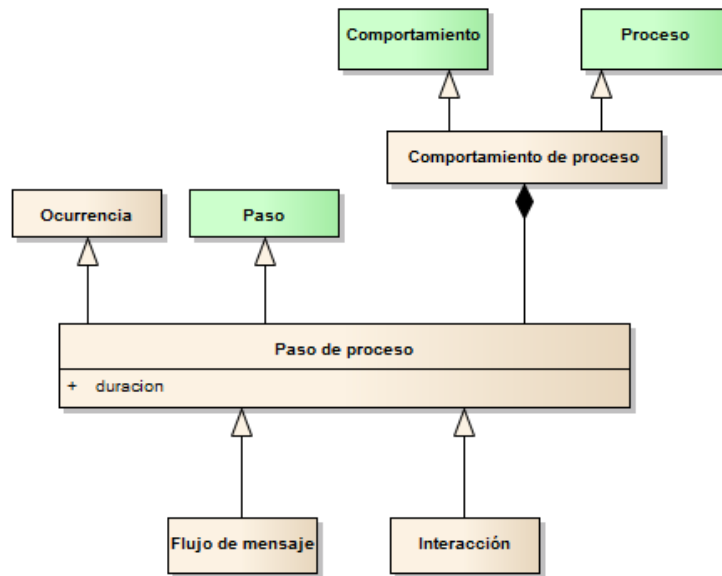


Figura 62: Paso de proceso del modelo de interacción de proceso de BPDM

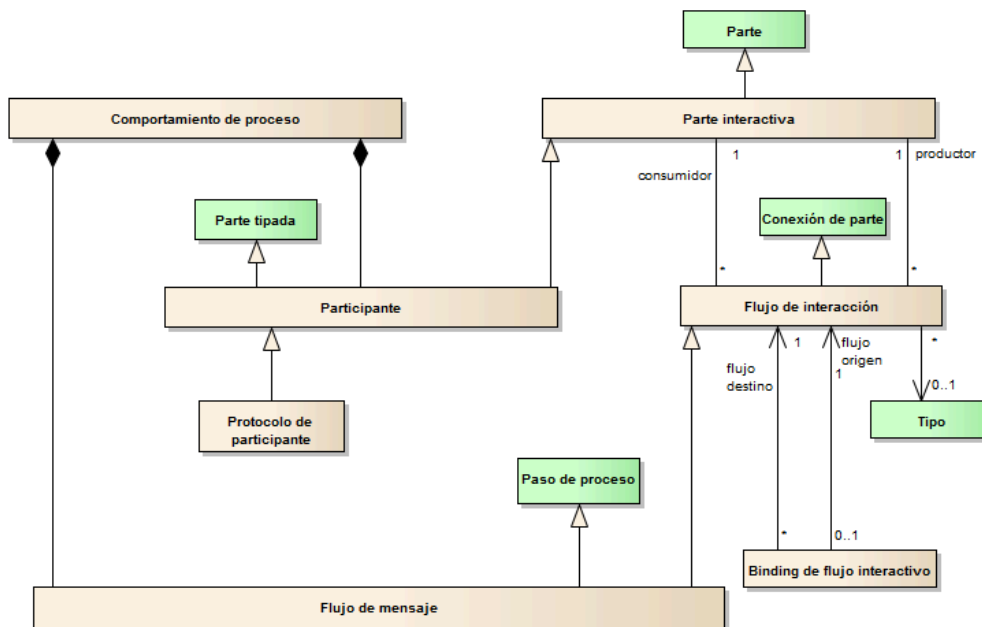


Figura 63: Flujo de mensaje del modelo de interacción de proceso de BPDM



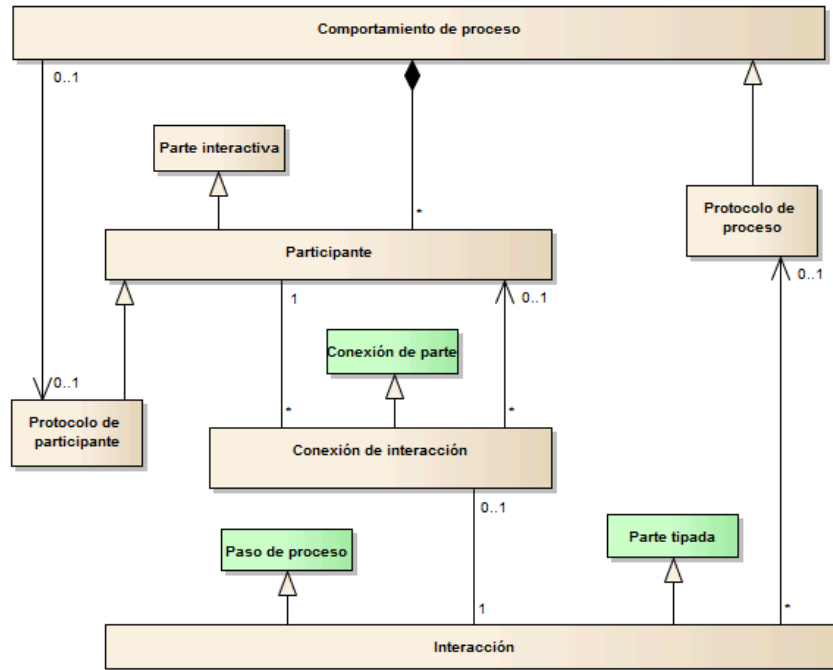


Figura 64: Interacción del modelo de interacción de proceso de BPDM

### 5.5.5.2 El metamodelo de BPMN 2.0

La aparición de la versión 2.0 de la notación BPMN a finales del año 2010 y comienzos del 2011 supuso un cambio en cuanto al metamodelo utilizado por el estándar de modelado de procesos de negocio. El alto nivel de abstracción presente en el metamodelo BPDM, que puede observarse en los diagramas de los distintos modelos que lo componen, favoreció el cambio hacia un metamodelo propio para la notación.

Para ello, el OMG afrontó la creación de esta nueva versión de BPMN como un momento de cambio en el que responder a las necesidades del momento con la especificación de un único lenguaje de modelado con metamodelo, notación gráfica y formato de intercambio de modelos. Así, el pliego de condiciones relacionado con la especificación de la versión 2.0 de BPMN incluye los siguientes objetivos entre otros:

- Creación de una única especificación denominada Business Process Model and Notation (BPMN 2.0) que defina la notación, el metamodelo y el formato de intercambio de los modelos.
- Realización de los cambios necesarios para conciliar BPMN y BPDM bajo un lenguaje único y coherente.
- Dotar a la especificación de la capacidad para el intercambio de los modelos de procesos de negocio y su representación gráfica entre herramientas de modelado distintas, al tiempo que se preserve la integridad semántica de los modelos.
- Mejorar la capacidad actual de BPMN con respecto a la orquestación de modelos y coreografías como modelos independientes o integrados.

A través de estas condiciones, el Object Management Group quería mejorar la capacidad de los expertos del dominio para desarrollar, transmitir y entender los procesos de negocio correspondientes. Además, también se buscaba la forma de proporcionar una mejor selección de herramientas de modelado y entornos de ejecución y promover la creación de herramientas de análisis y diseño de procesos con mayor grado de especialización.

La figura que se incluye a continuación muestra un esquema del metamodelo de la notación BPMN 2.0 donde se muestran algunos de los elementos más importantes de la notación. Este esquema incluye un código de colores que permite diferenciar los distintos componentes de la notación BPMN en base a los grupos en los que se presentaron en el [capítulo 3](#) de este documento.

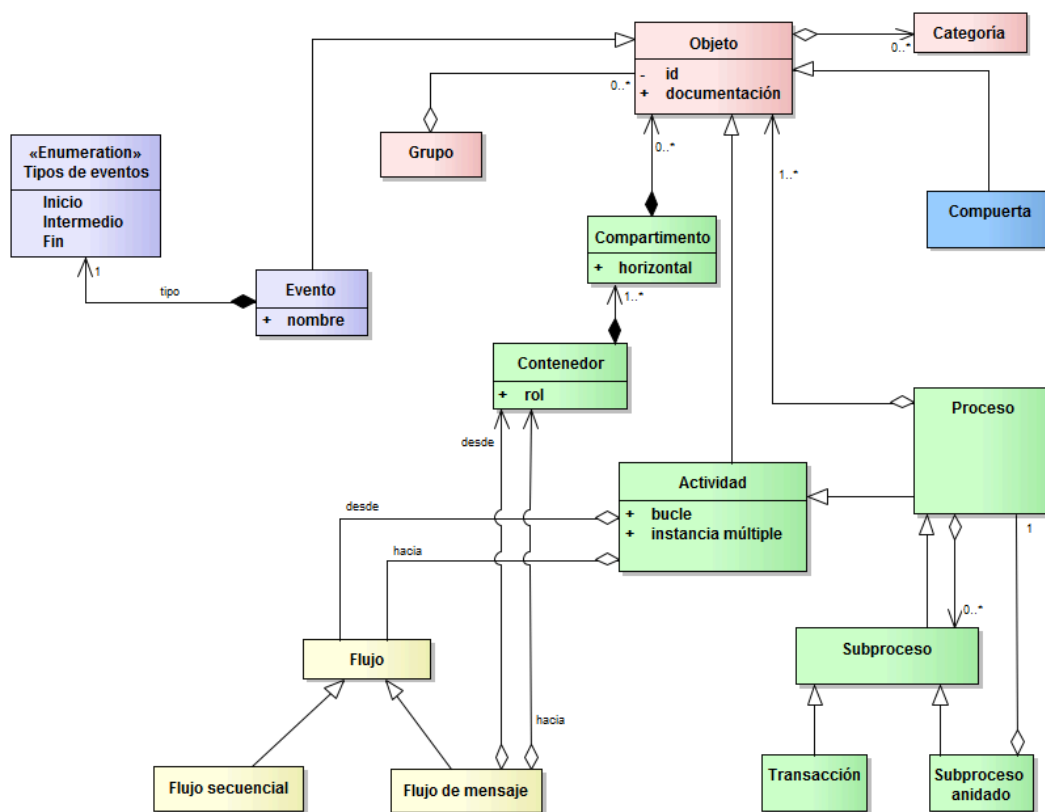


Figura 65: Esquema del metamodelo de BPMN 2.0

Así, los grupos principales de elementos que se incluyen en la especificación de la notación BPMN 2.0 están representados con el siguiente código de color: las compuertas están representadas en color azul, los eventos y sus tipos están representados en color morado, los flujos en color amarillo y el grupo de actividades y elementos relacionados con estas está representado en color verde. En la parte superior del metamodelo se puede observar un grupo de color rojo que representa entidades de carácter general como el objeto y la categoría.

Es posible entrar en más detalle sobre algunos de los elementos incluidos en el metamodelo anterior, mostrando una jerarquía de especialización de los mismos. Por ejemplo, es posible jerarquizar los elementos de tipo compuerta y flujo secuencial para

mostrar mayor nivel de detalle. Las figuras incluidas a continuación muestran la jerarquización de estos elementos.

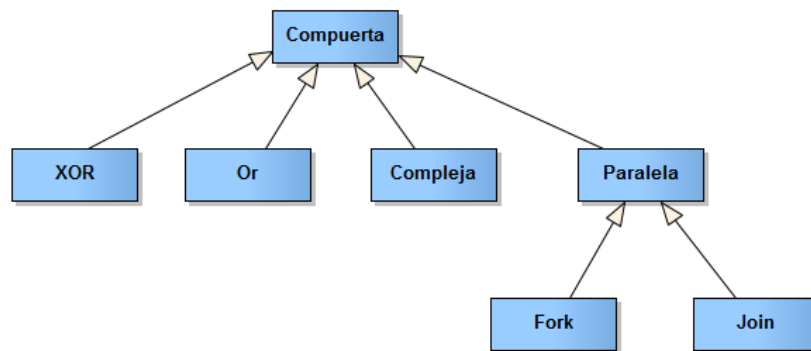


Figura 66: Jerarquización del elemento compuerta

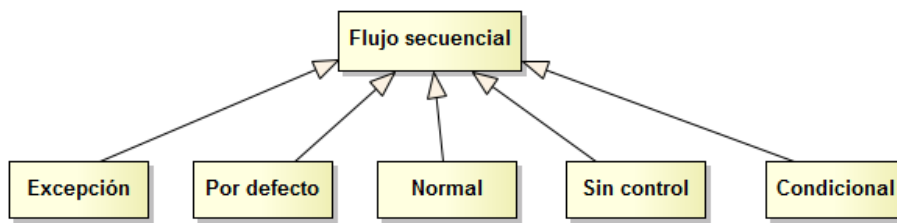


Figura 67: Jerarquización del elemento flujo secuencial

Se puede observar como estos grupos se corresponden, de forma prácticamente directa, con los elementos de la notación BPMN 2.0, ofreciendo esa unicidad en la especificación que buscaba el OMG al redactar el pliego de condiciones mencionado anteriormente.

En base al diagrama anterior se pueden construir una serie de ejemplos que permiten mostrar esta relación directa entre el metamodelo y los elementos de la notación. Si tomamos como muestra los grupos de colores amarillo, verde y azul, que se corresponden con las compuertas, los flujos y el grupo de actividades y elementos relacionados, es posible obtener la siguiente información: un contenedor estará compuesto por uno o más compartimentos; un compartimento es capaz de acoger muchos objetos, entre los que se encuentran procesos, compuertas y eventos, por ejemplo; una actividad dispone de un flujo de entrada y un flujo de salida; y un evento puede ser de varios tipos: inicial, intermedio o final.

Tomando como referencia la información descrita anteriormente es posible construir un modelo de proceso como el que se muestra en la imagen incluida a continuación. En este modelo de proceso se puede observar un contenedor principal con dos compartimentos que tienen en su interior eventos, compuertas y actividades. En concreto el diagrama muestra como sería la parte inicial del proceso de identificación de un usuario en un

sistema con arquitectura cliente-servidor, siendo uno de los compartimentos el dedicado a la parte cliente y el otro a la parte servidor. En el diagrama se puede observar cómo se realiza el procesamiento de la información en la parte servidora y cómo se realiza toda la parte de tratamiento de los datos en el apartado del servidor. En caso de que los datos de identificación sean incorrectos se lanza un evento de error y se pasa, a través de un flujo de mensaje, la información a la pantalla de identificación del usuario; por el contrario, si la identificación del usuario es correcta se muestra la interfaz privada.

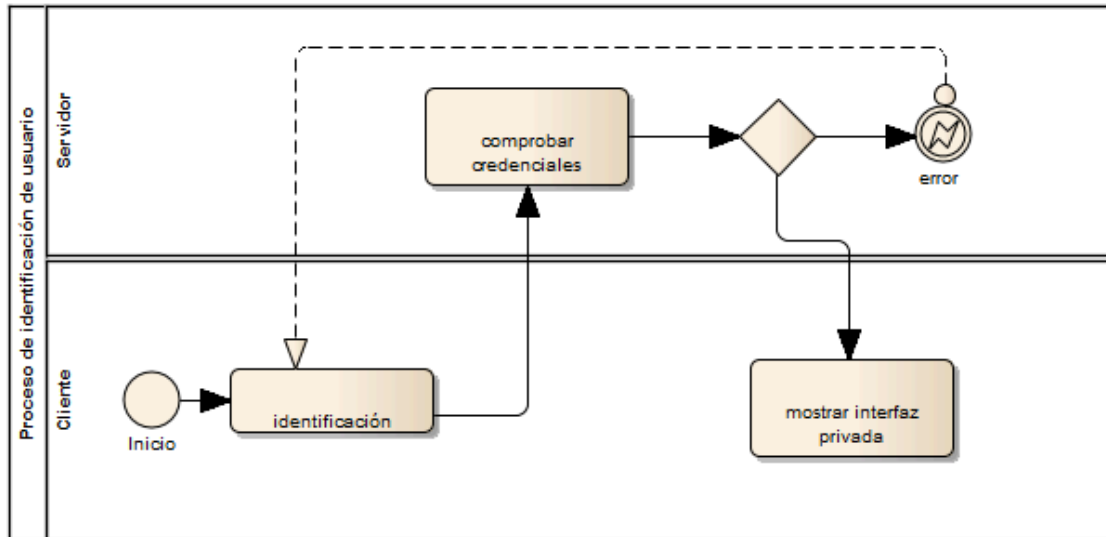


Figura 68: Diagrama BPMN de ejemplo

Este metamodelo de BPMN 2.0 pone las bases que permiten la construcción de un modelo formal para representar los procesos de negocio de un dominio dado a través de un conjunto de entidades comunes que son independientes del dominio en sí. De esta forma se consiguen crear los modelos de proceso que posteriormente pueden ser transformados de forma automática en artefactos software, lo que conlleva la automatización de algunas partes del proceso de desarrollo de software.

## 5.6 COMPARACIÓN DE MDE CON EL DESARROLLO TRADICIONAL

La utilización de un planteamiento derivado de la ingeniería dirigida por modelos introduce en el ciclo de vida del desarrollo de software un conjunto de diferencias con respecto al desarrollo de software realizado de forma tradicional. Según el trabajo de Kleppe [Kleppe et al. 2003] las principales diferencias son la productividad, la portabilidad, la interoperabilidad y el mantenimiento y documentación del sistema.

### 5.6.1 La productividad

Al acometer el desarrollo de un producto software existe una gran cantidad de documentación generada en las fases iniciales del desarrollo. Una de las situaciones más comunes durante el desarrollo es la realización de cambios dentro del software que llevan a que este no se parezca al concepto ideado inicialmente, por lo que se produce la necesidad de realizar cambios en toda la documentación.

MDE permite que cualquier cambio que se realice tenga un efecto inmediato y directo en los modelos, ya que los artefactos se generan de forma automatizada (en un planteamiento ideal), permitiendo un aumento de la productividad en el desarrollo. Existe otra posibilidad completamente válida: la interpretación o compilación de los modelos para que sean ejecutados en una máquina física o virtual.

### 5.6.2 La portabilidad

La naturaleza de la informática hace que aparezcan, de forma continua, nuevas tecnologías o versiones de tecnologías ya existentes que provocan la necesidad de realizar una migración de los sistemas. Este proceso de migración, que mantiene al día los sistemas que se migran, provoca un gasto económico y temporal considerable.

A través de la utilización de MDE, los modelos pueden ser independientes de la plataforma por lo que las modificaciones afectan solamente al resto de fases. Así, el efecto de las migraciones recae exclusivamente en las herramientas que se estén utilizando para realizar las transformaciones.

### 5.6.3 La interoperabilidad

En la actualidad, los sistemas informáticos necesitan comunicarse con otros sistemas para desempeñar sus funciones. Actualmente, es bastante probable que los sistemas estén utilizando tecnologías diferente por lo que obtener un buen nivel de interoperabilidad es una tarea complicada.

A la hora de trabajar con modelos que están relacionados entre si, y teniendo en cuenta que pueden estar pensados para trabajar con diferentes tecnologías, es probable que la comunicación entre los modelos no sea posible. La generación de puentes de

comunicación (también conocidos por el término inglés bridges) permite solucionar este problema y conseguir la interoperabilidad de los sistemas.

#### **5.6.4 El mantenimiento y la documentación**

La documentación es la parte del software en la que ponen más interés los encargados del mantenimiento y la utilización del sistema, por lo que durante el proceso de desarrollo no se enfatiza en la generación de una documentación de calidad. Esta deficiencia ha sido mitigada en cierto nivel por la utilización de los comentarios del código para la generación automatizada de documentación, aunque otras partes del sistema siguen sufriendo la baja calidad de la documentación.

A través del uso de MDE es posible generar todo el sistema a partir de los modelos por lo que toda la documentación de alto nivel estaría contenida en los modelos. Así, se produce una reducción de la cantidad de documentación que es necesario crear y mantener.

## 5.7 CONCLUSIONES

La ingeniería dirigida por modelos o MDE es una alternativa propuesta a principios de los años 2000 para corregir algunas de las limitaciones existentes en el proceso de desarrollo de software tradicional. A través de la utilización de modelos y las transformaciones de éstos es posible automatizar ciertas partes del ciclo de desarrollo del software, reduciendo los costes temporales y económicos derivados de la construcción de software.

Esta automatización es posible gracias a la existencia de los metamodelos, que permiten construir modelos formales que representan al dominio sobre el que se está trabajando y suponen el punto de unión entre el modelado de procesos de negocio y la ingeniería dirigida por modelos.

El enfoque propuesto por la ingeniería dirigida por modelos es el que se pretende utilizar en la propuesta de este trabajo para transformar los modelos de proceso de negocio, que construyan los expertos del dominio, en artefactos software personalizados que representen la funcionalidad descrita en dichos modelos.

Una vez introducidos los conceptos, soluciones y tecnologías existentes, se pasará a mostrar el desarrollo de la propuesta que compone el núcleo de este trabajo de investigación.





---

---

## PARTE III

# DESARROLLO DE LA PROPUESTA

---

---



# CAPÍTULO 6

## METODOLOGÍA DE MODELADO POR NIVELES PARA BPM

---

### *6.1 INTRODUCCIÓN*

En vista de las características y los inconvenientes de las notaciones que se han presentado anteriormente, se cree que una metodología por niveles adaptable a los conocimientos del usuario para la aplicación de BPM podría ser la solución para facilitar el modelado de procesos de negocio a los expertos de los mismos.

Además, se establecería un mecanismo de modelado que permitiría representar los procesos en forma de modelos con un grado de complejidad directamente proporcional a los conocimientos que posee el experto que los modela. De esta forma se lograría evitar que los expertos necesiten formación técnica para iniciarse en el modelado de procesos, logrando que los usuarios obtengan estos conocimientos a partir de la experiencia ganada con el modelado de sus procesos.

## **6.2 BPLOM: BUSINESS PROCESS LEVEL ORIENTED METHODOLOGY**

Según se ha especificado anteriormente, el modelado de procesos de negocio es una disciplina en la que participan los expertos de los procesos, personal que no tiene porque disponer de conocimientos técnicos, para documentar y explicar sus procesos de tal forma que los modelos generados puedan ser utilizados como alternativa a la captura de requisitos convencional. Las notaciones y lenguajes de modelado de procesos mostrados en el apartado anterior de este documento constituyen una muestra significativa en cuanto a complejidad y expresividad se refiere. Algunos de ellos poseen una complejidad excesiva para su uso por parte de expertos del dominio. Este es el caso del estándar BPMN [Wahl and Sindre 2005] o de las redes de Petri, cuyas características las hacen apropiadas para el modelado de procesos pero cuyo nivel de complejidad las aleja de ser usables para los expertos del negocio.

Por otro lado algunas de las simplificaciones de BPMN estudiadas, como los casos de jPDL o SBPMN, obtienen mejores resultados de usabilidad pero siguen ofreciendo un número de entidades elevado que no favorece la introducción del experto al modelado de sus procesos. Esto es debido a que utilizan desde el primer momento una serie de conceptos tecnológicos que posibilitan la declaración de procesos muy detallados pero bajan el nivel de abstracción del usuario.

Ante esta situación, hemos optado por definir una metodología para la aplicación de BPM a procesos de negocio de cualquier tipo, que promueva la adaptación de la notación de modelado a los conocimientos técnicos del usuario. A esta metodología la llamaremos BPLOM: Business Process Level-Oriented Methodology [Solís-Martínez et al. 2013].

BPLOM es una metodología por niveles para la aplicación de BPM que establece un lenguaje de modelado a partir de un subconjunto de BPMN, la notación estándar para el modelado de procesos de negocio introducida por el OMG. El nivel 0 ofrecerá a los usuarios un conjunto con el conjunto mínimo de símbolos necesarios para el modelado de procesos, lo que provoca que tenga una complejidad baja y permita que cualquier tipo de usuario pueda modelar sus procesos sin necesidad de poseer conocimientos técnicos. Los siguientes niveles irán introduciendo entidades de mayor complejidad a la notación, consiguiendo que esta gane una riqueza expresiva que permite definir los procesos de negocio con mayor nivel de detalle. A tenor de esta ganancia en expresividad se produce también un aumento de la complejidad de uso de la metodología, debido a una bajada del nivel de abstracción del que disfruta el usuario al utilizar las nuevas entidades incluidas en cada nivel; sin embargo, esta bajada del nivel de abstracción es proporcional a la ganancia de conocimientos técnicos que se le suponen al experto según va avanzando por los niveles de la metodología.

Esta circunstancia se encuentra reflejada en la captura incluida a continuación, donde se muestra el nivel 0 de la metodología como la base de la misma y se establece que la ganancia de niveles radica en un aumento de la expresividad de la notación y, por lo tanto, de la complejidad de uso de la misma.

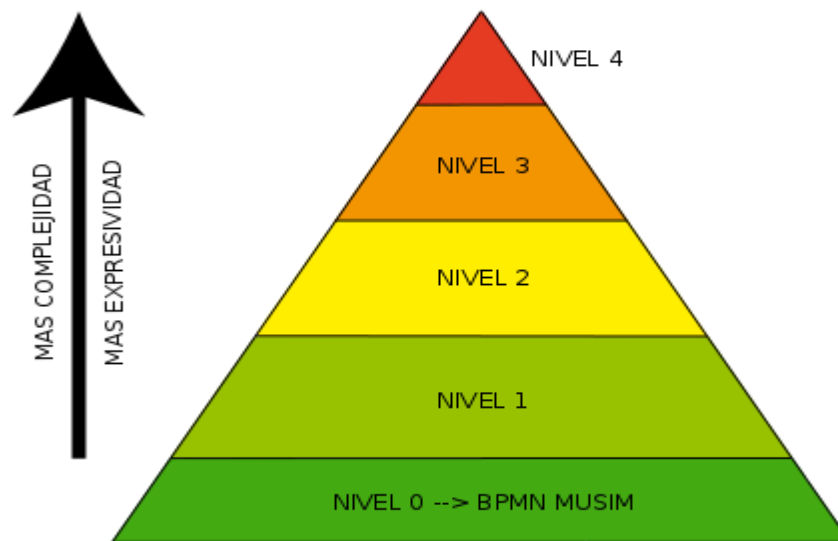


Figura 69: Pirámide de complejidad y expresividad de BPLOM

Para promover el uso de la propuesta se acompañará de dos prototipos que ofrecen soporte a la metodología: por un lado, una herramienta de modelado de procesos de negocio, llamada BPLLevel Modeler, que permite construir un modelo gráficamente a partir de este lenguaje; y, por el otro, una herramienta generadora de código, denominada BPLLevel Generator, que interpreta los modelos construidos con la herramienta anterior y genera aplicaciones informáticas personalizadas, multigénero y multiplataforma con soporte para los modelos construidos. Estos prototipos se mostrarán en más detalle en la parte IV de este documento.

### 6.2.1 Características de la metodología

A continuación se incluye un breve resumen de las características más significativas de la metodología de modelado por niveles propuesta.

- **Carácter incremental de los niveles.** Los niveles de esta metodología permiten la utilización de las entidades propias del nivel más todas las entidades incluidas en los niveles anteriores.
- **Ejecución de todos los procesos.** Todos los procesos que se modelen a partir de la metodología podrán ser traducidos a código y ejecutados, gracias a la herramienta generadora de código y sin importar el nivel de la metodología en el que se encuentren.
- **Independencia de la plataforma.** Aunque en este caso se haya asociado la notación a la utilización de la plataforma .NET, esta metodología no está ligada a ninguna plataforma por lo que podría utilizarse para generar código para otras plataformas como Java o iOS. Esta característica podrá verificarse en próximos

fragmentos de este documento, donde se detalla la construcción de aplicaciones para casos concretos utilizando la metodología.

- **Recorte de las fases del esquema de BPM**, ya que serán los propios expertos de los procesos los encargados de realizar los modelos a partir de los niveles ofrecidos y estos modelos serán cumplimentados y traducidos a código. De esta forma, ya no será necesaria la interpretación de los requisitos textuales y posterior traducción de estos a modelos por parte del analista, eliminando los posibles errores que este pudiera cometer debido a la ambigüedad del lenguaje natural y a su falta de conocimiento del dominio del problema [**Kotonya and Sommerville 1996**].

Por otro lado, el inconveniente principal de la metodología de modelado por niveles es el grado reducido de expresividad en comparación con el estándar. El conjunto de símbolos ofertado es menor que el ofertado por BPMN, por lo que la notación es menos rica en expresividad que la notación estándar definida por el OMG. Sin embargo, a partir de los resultados obtenidos en las pruebas que acompañan a esta investigación, se observa que con el conjunto de símbolos ofrecidos por todos los niveles de la metodología es posible definir procesos de cualquier tipo con una expresividad suficiente, eliminando aquellas construcciones menos utilizadas en la notación estándar de modelado de procesos de negocio.

### 6.3 PRESENTACIÓN DE LOS NIVELES DE LA METODOLOGÍA

A continuación se procederá a presentar todos los niveles incluidos en la metodología a partir de la enumeración de las entidades que los componen y mostrando también como sería la aplicación de la metodología a dos procesos de negocio reales.

#### 6.3.1 Nivel 0: BPMN MUSIM

BPMN MUSIM [Solís Martínez et al. 2011a] [Solís Martínez et al. 2011b] [Solís-Martínez et al. 2014], abreviatura de BPMN MUy SIMple, pretende ser la versión mínima de la notación BPMN con la que los expertos de los procesos puedan realizar sus modelados sin necesidad de disponer de unos conocimientos técnicos amplios, por lo que es la elección idónea para el nivel 0 de esta metodología.

La definición de esta nueva notación se basa en los resultados de varios estudios que permiten pensar que una reducción del estándar BPMN sería beneficiosa. A partir de estos estudios se establece el número de entidades que formarán parte de BPMN MUSIM y se proponen unas representaciones gráficas acordes a la simplificación del uso y comprensión de los modelados realizados con esta nueva notación.

##### 6.3.1.1 ¿Por qué BPMN MUSIM como nivel 0?

En primer lugar, un estudio publicado por Muehlen y Recker en 2008 [Muehlen and Recker 2008] refuerza los resultados acerca del bajo nivel de utilización de un número significativo de las entidades contenidas en BPMN que se habían mencionado en las otras publicaciones mencionadas [Recker 2008] [White and Miers 2008].

A través del análisis matemático y estadístico de 120 diagramas realizados utilizando la notación BPMN, Muehlen y Recker han conseguido acotar el conjunto de entidades de la notación estándar de modelado de procesos de negocio que más se utilizan en este tipo de diagramas. Así, tal y como muestra la Figura 70 donde se intenta resumir el contenido del artículo citado, el conjunto de entidades más utilizadas en el modelado de procesos de negocio con BPMN está compuesto por cuatro entidades: los eventos de inicio y fin, las actividades o tareas y las transiciones entre las entidades anteriores.

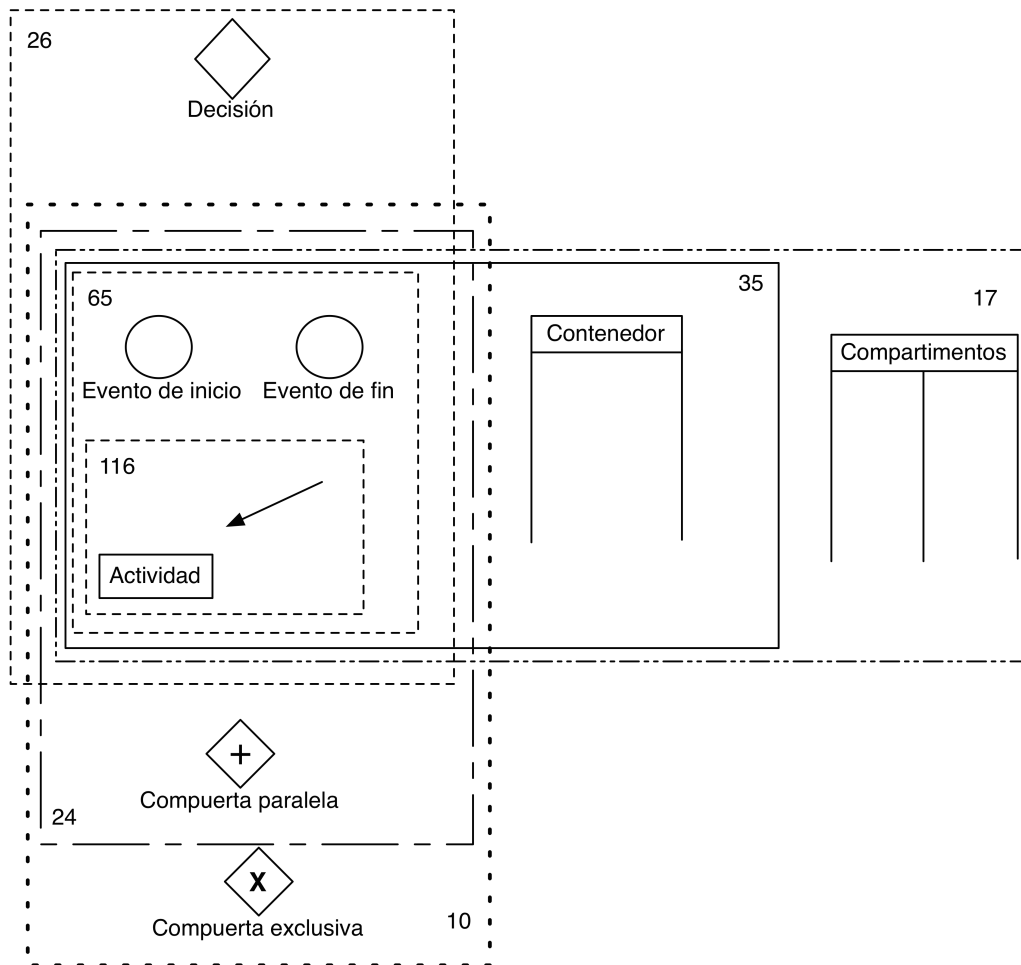


Figura 70: Estudio sobre elementos BPMN más usados según Muehlen y Recker

Otro punto a favor de partir de una simplificación de BPMN son los resultados obtenidos en SBPMN, la primera simplificación de BPMN llevada a cabo en el grupo de investigación MDE-RG [MDE-RG]. La Figura 71 muestra una de las gráficas incluidas en el artículo de presentación de esta notación [Fernández et al. 2010] donde se muestra tanto los fallos (*failures*) cometidos por los usuarios como los aciertos (*skills*) de éstos al utilizar las notaciones BPMN y SBPMN.



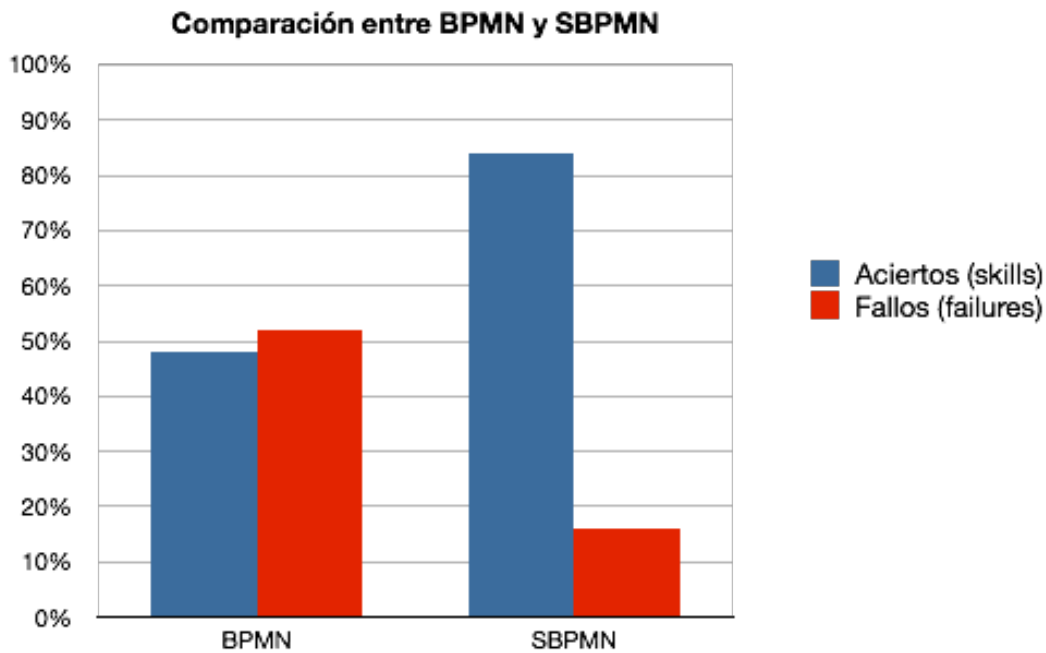


Figura 71: Comparación entre BPMN y SBPMN

Tal y como muestra la figura anterior, la utilización de BPMN supone la existencia de un nivel de fallos de algo más del 50% mientras que a través de la utilización de SBPMN se logra reducir este nivel de fallos a algo menos del 20%. Esta amplia reducción de la tasa de fallos permite establecer que la reducción propuesta por SBPMN supone una mejora en la utilización de una notación de modelado de procesos de negocio por parte de expertos del dominio con pocos conocimientos técnicos y de modelado.

Teniendo en cuenta que el número de entidades que se ofrecen en SBPMN para la realización de estos modelados es todavía elevado y considerando el conjunto de elementos de BPMN más utilizados según Muehlen y Recker, es posible pensar que una nueva reducción de la notación consiga ofrecer una mejora en el modelado de los procesos de negocio. Así, se toma la base de reducción propuesta en SBPMN para establecer una nueva reducción de BPMN que permita modelar los procesos de negocio a través del mínimo número de símbolos posible.

### 6.3.1.2 Entidades que forman BPMN MUSIM

BPMN MUSIM toma como entidades aquellos componentes de BPMN que son considerados esenciales para el modelado de procesos de negocio, de tal forma que incluye únicamente las entidades indispensables. Los elementos que componen BPMN MUSIM son: punto de inicio, punto de fin, actividad, transición y decisión.

Debido a que BPMN MUSIM es una simplificación del estándar BPMN se ha intentado acercar la representación gráfica de sus entidades a las del estándar, facilitando la comprensión de la nueva notación a los usuarios de BPMN; sin embargo, algunos de los

símbolos utilizados en BPMN pueden resultar confusos para los usuarios por lo que se ha optado por incorporar colores para simplificar la comprensión de BPMN MUSIM.

Además de lo anteriormente citado se ha intentado acercar esta representación gráfica a una de las conclusiones extraídas por Recker, Safrudin y Rosemann en un estudio sobre modelado llevado a cabo con personas noveles en la disciplina [Recker et al. 2010] donde se establece que se reconocen mejor aquellos modelos que constan de símbolos abstractos (círculos, cajas y flechas) en combinación con texto que aquellos que disponen de figuras más definidas. Durante su estudio estos autores han podido reportar algunos casos en los que los usuarios no entendían correctamente las figuras existentes y por lo tanto esto genera una ambigüedad en la interpretación y comprensión del modelo.

La tabla que se incluye a continuación muestra los detalles de cada una de las entidades incluidas en este nivel inicial de la metodología.

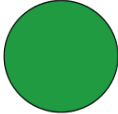
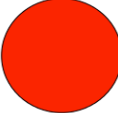
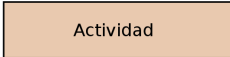

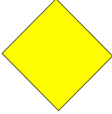
Nombre	Representación	Descripción
Punto de inicio		Todo proceso contará con un punto de inicio que marcará el comienzo del proceso.
Punto de fin		Los procesos contarán con uno o más puntos de final que marcan los puntos en los que el proceso acaba.
Actividad		Las actividades son la parte central de BPMN MUSIM. Esta entidad representa una tarea, bien realizada de forma automática o realizada por el usuario.
Transición		Las transiciones representarán el orden de ejecución de las distintas actividades que compongan un proceso. La punta de la flecha indica el sentido de la transición.
Decisión		Las decisiones permiten seleccionar alternativas durante la ejecución del proceso. Esta entidad irá acompañada de una condición de tipo booleano y poseerá dos transiciones salientes que representan los dos resultados posibles de la evaluación de la condición.

Tabla 5: Entidades de BPMN MUSIM, nivel inicial de la metodología

### 6.3.2 Nivel 1: Extensión de decisiones

Una vez establecido el nivel inicial de la metodología, cuya simplicidad facilita la introducción del experto al modelado de procesos, es necesario ir añadiendo componentes a la misma para dotarla de una expresividad comparable a la notación estándar BPMN. Teniendo en cuenta el gráfico presentado como base para la definición del nivel 0, un buen punto para la ampliación de primer nivel es una extensión de decisiones.

Tal y como se explica en el apartado de entidades desestimadas, esta metodología ha descartado la inclusión de los contenedores y compartimentos debido a la complejidad que aportan a la comprensión de los modelos. Así, las siguientes construcciones más utilizadas en el modelado según la figura incluida en la investigación de Muehlen y Recker son los tipos adicionales de decisiones, por lo que creemos que esta es la ampliación más acertada para el segundo nivel de la metodología.

La siguiente tabla muestra las entidades que forman este nivel, incluyendo su representación gráfica y una breve explicación de su cometido.



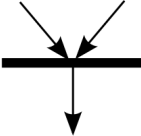




Nombre	Representación	Descripción
Decisión paralela		Una decisión paralela permite al usuario definir un marco dentro del proceso donde se ejecutan de forma simultánea dos caminos dentro del proceso.
Decisión inclusiva		La decisión inclusiva permite activar al menos uno de los caminos que salen de ella en función de una condición de entrada.
Punto de convergencia		La inclusión en la metodología de las decisiones paralelas e inclusivas obliga a incluir un punto de convergencia donde se realiza la espera necesaria para que todos los caminos de ejecución terminen antes del paso a la siguiente actividad.

Tabla 6: Entidades de la extensión de decisiones

### 6.3.3 Nivel 2: Extensión de eventos

Una vez establecidos los mecanismos para que los usuarios puedan modelar sus procesos con toma de decisiones complejas y teniendo en cuenta que según los resultados de Muehlen y Recker [Muehlen and Recker 2008] los usuarios consideran los eventos como una parte importante de BPMN, se establece que el nivel 2 de la metodología sea una extensión de eventos.

Los eventos representan la ocurrencia de acontecimientos durante la ejecución de los procesos, como el envío de un mensaje o de una señal a otro proceso, y pueden provocar un cambio en la ejecución de los mismos. Los eventos contemplados en la definición de esta metodología se encuentran en la tabla incluida a continuación.

Nombre	Representación	Descripción
Evento de mensaje		Los eventos de mensaje permiten al usuario modelar la recepción y envío de mensajes dentro de un proceso concreto, sirviendo de mecanismo de comunicación entre distintos procesos y/o usuarios. La distinción entre los mensajes entrantes y los mensajes salientes viene marcada por las flechas incluidas en la representación gráfica de esta entidad.
Evento temporal		Un evento temporal es el que permite definir una condición basada en aspectos temporales dentro de un proceso, como por ejemplo que el proceso tenga que esperar a que se complete una tarea o a que pase un tiempo determinado antes de continuar.
Evento de error		Los eventos de error permiten marcar el momento en el que el proceso se detiene debido a un error, terminándose su ejecución de inmediato.
Evento de cancelación		Los eventos de cancelación serán los encargados de señalar el momento en que se cancela la ejecución del proceso por cualquier circunstancia.

Evento de señal		Los eventos de señales permiten que un proceso envíe señales a otros procesos a modo de aviso para que continúen ejecutándose.
-----------------	---	--

Tabla 7: Entidades de la extensión de eventos

### 6.3.3.2 Utilización de los eventos en los modelos de proceso

Una vez establecidos los distintos tipos de eventos que pueden utilizarse en el modelado de los procesos de negocio es necesario explicar como realizar la inclusión de estos en los modelos.

Los eventos pueden aparecer en cualquier punto de un modelo de proceso, a excepción de los eventos de cancelación y error que aparecerán siempre como elementos al final de una de las ramas del proceso. Por ejemplo, un evento de mensaje entrante podría aparecer inmediatamente después del punto de inicio, entre dos actividades o antes de la toma de una decisión.

Los eventos son construcciones que hacen que el proceso tenga que esperar a su ocurrencia para continuar con su ejecución. Así, colocando un evento de mensaje saliente antes de una decisión se provoca que el proceso quede parado hasta que el mensaje sea enviado e incluyendo un evento temporal antes de una actividad se obliga al proceso a esperar el tiempo indicado antes de la ejecución de la misma. Por lo tanto los eventos tendrán que situarse siempre después de la entidad que los provoca y antes de la entidad a la que afectan en su ejecución.

Pongamos un ejemplo: se necesita modelar un sistema que ejecuta una actividad que provoca el envío de un mensaje y que, una vez enviada la comunicación, toma una decisión para ejecutar un camino u otro; este comportamiento se modelaría situando la actividad en primer lugar, colocando el evento de mensaje saliente tras ella y enlazando la decisión como entidad posterior al evento de mensaje, de tal forma que el proceso quede esperando al envío del mensaje para pasar a ejecutar la decisión correspondiente. Este comportamiento se podrá observar, con mayor claridad, en el [capítulo 7](#) a través de los procesos de ejemplo a los que se aplicará la metodología.

### 6.3.4 Nivel 3: Extensión de actividades

El cuarto nivel de la metodología propuesta a través de esta investigación se corresponde con una extensión de actividades. Hasta la entrada de este nivel, los expertos que quisieran modelar sus procesos a partir de la metodología solamente disponían de una entidad de tipo actividad, una construcción que representa a una tarea automática o realizada por el usuario.

Sin embargo, el modelado de procesos de negocio complejos requiere una serie de construcciones relacionadas con las actividades que permitan incluir subprocesos y llamadas a funciones en sus modelos. Es por ello que la extensión de actividades incluye

actividades de llamada y subprocesos. Los detalles están incluidos en la tabla que se muestra a continuación.


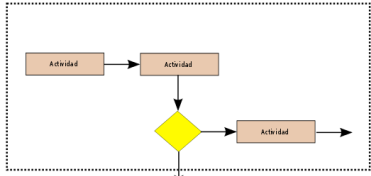
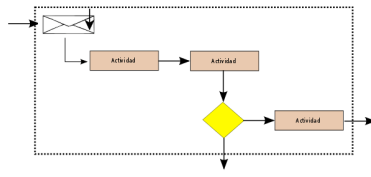
Nombre	Representación	Descripción
Actividad de llamada		<p>Una actividad de llamada hace referencia a una tarea global que se repite con frecuencia en varios procesos de distinta índole, por ejemplo la identificación de un usuario a través de su nombre y contraseña. A través de esta entidad se pueden realizar llamadas a estas tareas sin necesidad de definirlos como una actividad propia de cada proceso en el que aparecen. Tendrá la misma representación que una actividad normal pero añadiendo una bola del mundo en la esquina superior izquierda.</p>
Subproceso		<p>Un subproceso es un conjunto de tareas acotado que se ejecuta de inicio a fin de forma seguida y sin interrupción alguna. En caso de que la ejecución de alguna de las actividades produzca un error, se cancelará la ejecución del proceso que lo incluye. El subproceso queda marcado por un rectángulo de bordes discontinuos que rodea a todos los elementos del modelo que están incluidos dentro del subproceso.</p>
Subproceso de evento		<p>Un subproceso de evento representa un conjunto de tareas que se desempeñan de forma ininterrumpida a partir de la aparición de un evento en el proceso. Esta entidad se comportará de igual manera al subproceso normal ante la aparición de un error en la ejecución de una de sus actividades. La representación gráfica de un subproceso de evento será igual a la del subproceso normal, aunque en este caso existe la obligación de que la primera entidad del subproceso sea el evento en cuestión.</p>

Tabla 8: Entidades de la extensión de actividades

### 6.3.5 Nivel 4: Extensión de datos

El último nivel de la metodología está destinado a aquellos expertos de los procesos de negocio que tienen más conocimientos técnicos y desean realizar modelos con un nivel de detalle tecnológico más alto. Por ello, el último nivel de la metodología de modelado por niveles se corresponde con una extensión de datos.

A lo largo de la definición de la metodología se han ido añadiendo entidades que la dotan de expresividad sin provocar un descenso acusado del nivel de abstracción del usuario. Sin embargo, llegado el último nivel, se ha optado por ofrecer soporte a características relacionadas de forma más estrecha con la tecnología, como son las entidades relacionadas con los datos que manejan los procesos.

Esta circunstancia hace que el nivel de abstracción del usuario descienda ligeramente, aunque teniendo en cuenta que este nivel está destinado a usuarios con ciertos conocimientos tecnológicos este descenso no debe significar un problema.

La tabla que se incluye a continuación muestra las entidades incluidas en este último nivel de la metodología y la representación gráfica que se les ha otorgado.




Nombre	Representación	Descripción
Objeto de datos		Un objeto de datos representa información que fluye a través del proceso en distintas formas como documentos o datos introducidos en un formulario.
Colección de objetos		Una colección de objetos representa a un conjunto de objetos de datos que circula por el proceso que se modela, como por ejemplo una lista de documentos.
Almacén		Un almacén representa el momento en el que el proceso lee o escribe datos en un soporte como una base de datos. Esto implica que los datos generados durante el proceso y metidos en el almacén sobreviven a la instancia del proceso que se está ejecutando.

Tabla 9: Entidades de la extensión de datos

## 6.4 ENTIDADES DESCARTADAS EN LA VERSIÓN INICIAL DE LA METODOLOGÍA

La realización de esta versión inicial de la metodología se centra en ofertar a los expertos de los procesos de un mecanismo de modelado que se adapte al nivel de conocimiento y pericia que puedan tener en la técnica de modelado de procesos.

Tal y como se ha comprobado, BPMN ofrece un nivel de detalle tecnológico que es demasiado elevado para ciertos expertos de procesos y también dispone de una serie de construcciones que complican el modelado de procesos para estos; así, se ha decidido no incluir alguna de las entidades ofrecidas por BPMN en favor del acercamiento del modelado de procesos al nivel de conocimientos de la mayor parte de los expertos. Sin embargo, las construcciones descartadas en esta versión inicial de la metodología de modelado por niveles BPLOM son susceptibles de ser incluidas en posteriores revisiones de la misma según los resultados y las conclusiones que puedan ir extrayéndose de la utilización de la metodología en nuevos ámbitos, circunstancia que se encuentra recogida en la sección de trabajo futuro del [capítulo 10](#) de este documento.

A continuación se incluye una lista detallada de las construcciones que no se han incluido y las razones por las que se ha considerado su exclusión de esta versión inicial de la metodología.

### 6.4.1 Contenedores

Los contenedores definidos en el estándar BPMN permiten la definición de más de un proceso en un mismo diagrama, incluyendo dentro de cada contenedor un proceso y permitiendo la existencia de comunicación entre actividades contenidas dentro de cada proceso y contenedor.

Uno de los objetivos que persigue esta investigación es lograr que cada usuario modele los procesos de los que es propietario de forma independiente a otros procesos que puedan existir en su entorno. El cometido de los contenedores es posibilitar la definición de más de un proceso en un mismo diagrama para mostrar la comunicación existente entre éstos. Esta circunstancia puede traer complicaciones al experto por dos vías: la primera, que sean definidos un alto número de procesos en el mismo diagrama por lo que la comprensión de los procesos no sea sencilla; y la segunda, que uno de los procesos incluidos en los contenedores no sea propiedad del experto que está realizando el modelado, por lo que este no podría modelarlo de forma correcta.

Además, teniendo en cuenta la existencia de mecanismos de comunicación como los mensajes o las señales, que ya forman parte de esta metodología, no se considera necesario la inclusión de los contenedores como mecanismo de comunicación.

### 6.4.2 Datos de entrada y datos de salida

Los datos de entrada y de salida representan en BPMN 2.0 a aquellos inputs que pueden ser leídos por un proceso y los outputs que genera la ejecución del mismo. En el último



nivel de esta metodología se ha introducido al usuario los conceptos objeto de datos y colección de objetos, entidades que representan a uno o varios documentos con la información que recibe o genera el proceso.

En esencia, los datos de entrada o de salida de un proceso se pueden representar como un objeto de datos o una colección de objetos, por lo que se ha decidido ofrecer solamente una de las dos alternativas de representación de datos. Así, los datos de entrada y de salida han quedado descartados en favor de los objetos y las colecciones, favoreciendo un enfoque de los procesos más acorde a la realidad tecnológica de la actualidad donde se utiliza ampliamente el concepto de objeto; este enfoque tecnológico es posible debido a la participación de estos conceptos en el nivel más alto de nuestra metodología, ya que este está dirigido a expertos de proceso con el mayor nivel de conocimientos tecnológicos.

### **6.4.3 Flujo condicional y flujo por defecto**

El flujo condicional que se oferta como entidad dentro de la notación BPMN 2.0 se corresponde con una construcción que permite activar o no un camino del proceso en función del resultado de la evaluación de la condición que acompaña al flujo. Esta entidad actúa de forma idéntica a la entidad decisión incluida en el primer nivel de esta metodología, que activa un camino de entre las dos alternativas posibles de evaluación de la condición que acompaña a la decisión. Por lo tanto, se considera que la entidad flujo condicional ofrece un comportamiento ya ofrecido por otra entidad.

El flujo por defecto es una entidad que permite la ejecución del camino indicado en caso de que los flujos condicionales que hayan sido evaluados sean todos falsos. Debido a la eliminación del flujo condicional en esta metodología por las razones descritas anteriormente, no tiene sentido incluir el flujo por defecto ya que este se incluye como una entidad complementaria al flujo condicional.

### **6.4.4 Compuertas complejas**

Las compuertas complejas son las entidades de tipo decisión que se aplican en aquellos casos en los que el resto de decisiones no son idóneas. En esta metodología se incluyen decisiones de varios tipos: la decisión normal, donde se escoge un camino u otro en función de una condición de entrada; la decisión paralela, que permite la ejecución simultánea de varios caminos en un proceso; la decisión inclusiva, una combinación de las dos anteriores donde se activa al menos uno de los caminos del proceso; y la decisión basada en eventos, que permite combinar cualquiera de las decisiones mencionadas anteriormente con la existencia de un evento que las desencadene.

Este catálogo de decisiones que incluye nuestra metodología permite modelar de forma completa procesos complejos por lo que no se considera necesario incluir una entidad como las compuertas compleja, elementos que complicarían el modelado de procesos a aquellos usuarios que opten por utilizar el nivel 1 de la metodología sin que esta nueva entidad les reporte una ganancia ostensible.

### **6.4.5 Transacciones**

Las transacciones, conjuntos de actividades relacionadas lógicamente y adheridas a un protocolo transaccional particular, son otra de las construcciones eliminadas en esta metodología. La inclusión de los subprocesos, conjunto de actividades relacionadas entre sí ejecutado de principio a fin, permite modelar el comportamiento de las transacciones, especificando además que éstas sean lanzadas ante la aparición de un evento.

Por lo tanto, ante la existencia de una entidad que permite modelar el comportamiento de las transacciones de forma completa y con mayor número de opciones gracias a la inclusión de los eventos se ha decidido descartar la inclusión de las transacciones como una parte de las entidades ofrecidas por la metodología descrita.

### **6.4.6 Conversaciones y coreografías**

Las conversaciones y las coreografías son otro de los mecanismos ofertados por OMG dentro de BPMN para lograr la comunicación entre varios procesos. En concreto, las conversaciones permiten establecer un intercambio de mensajes que están relacionados entre sí de forma lógica entre uno o varios participantes mientras que la coreografía permite el intercambio de mensajes entre dos participantes. Bajo la metodología que se propone aquí, se establece que el intercambio de mensajes se realiza entre dos procesos a través de la utilización de las señales que permiten que un proceso envíe un mensaje a otro a través de un evento de tipo señal.

De esta manera se pretende simplificar el intercambio de información entre dos procesos, evitando ofrecer dos entidades con características similares y evitando que un usuario tenga que incluir en su definición de proceso la parte del otro proceso con la que realiza la comunicación o coreografía; esto se consigue debido a que no es necesario que el usuario conozca el otro proceso ni detalle el momento de envío/recepción de la señal mientras que en la coreografía o comunicación es necesario que el experto incluya dentro de su modelado al otro proceso y establezca el momento y la manera en la que se realiza la conversación o coreografía.

### 6.5 COMPARACIÓN DE EJEMPLO DE MODELADO ENTRE BPMN Y BPLOM

Una vez que se han introducido los niveles que componen la metodología BPLOM y que se ha explicado las razones por las que se ha decidido no incluir ciertas construcciones en esta versión inicial de la metodología, es conveniente realizar una comparación de un mismo proceso de negocio modelado mediante dos alternativas: la notación estándar BPMN y la metodología que se propone en esta tesis. Para realizar esta comparación se ha decidido utilizar a modo de ejemplo el proceso de negocio asociado al encargo de una pizza para servir a domicilio, ya que este proceso se muestra como ejemplo de uso de la notación BPMN en la documentación disponible en el sitio web del Object Management Group [OMG].

El proceso de encargo de la pizza se inicia cuando el cliente escoge la pizza que desea y realizar el encargo. Una vez que el encargo es recibido en el restaurante se comienza a preparar la pizza en el restaurante. En caso de que el cliente no reciba la pizza en el periodo de tiempo estipulado éste realizará una reclamación al restaurante para saber qué ocurre con su pedido. Sin embargo, si el pedido se completa de forma satisfactoria el cliente recibe la pizza en su domicilio y abona la cantidad correspondiente al repartidor. Este intercambio marca el final del proceso.

La imagen que se incluye a continuación muestra el proceso anterior tal y como indica el OMG en la documentación mencionada anteriormente. Se puede observar como el proceso marca, utilizando los contenedores y compartimentos para marcar la participación de los distintos actores del proceso. Además, se puede observar como se produce el paso de mensajes entre el cliente y el restaurante en diversas de las fases del proceso, marcados por unas flechas discontinuas que unen las actividades y eventos existentes.

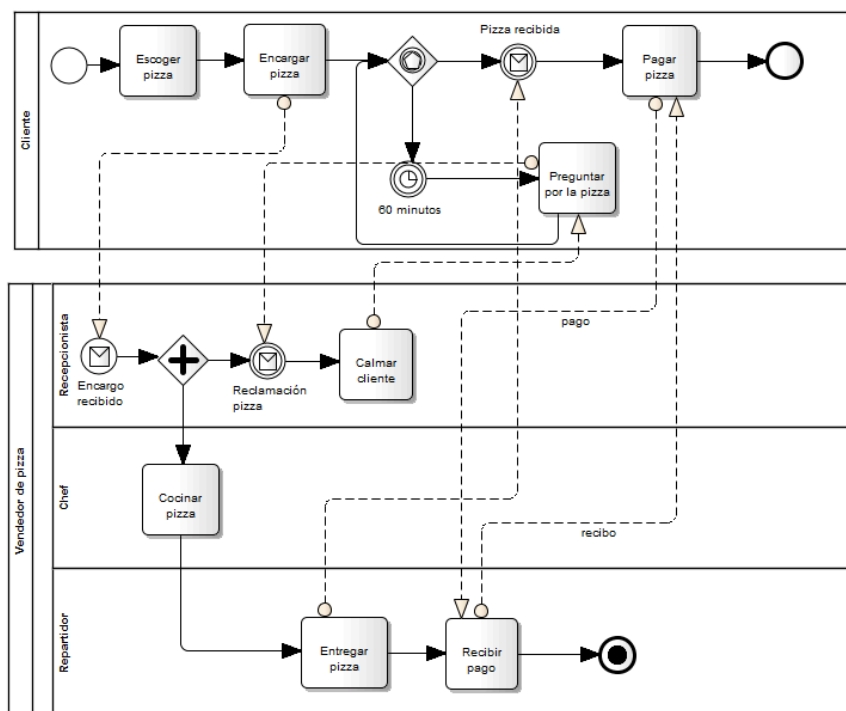


Figura 72: Proceso de encargo de pizza modelado con BPMN

Utilizando todos los niveles de la metodología propuesta en este capítulo, el modelado del proceso de encargo de la pizza se ha realizado de la siguiente manera. En primer lugar, ante la apuesta que se hace por la separación de los procesos en varios modelos según el experto que sea propietario de los mismos, el proceso de encargo de una pizza se ha separado en varios subprocesos independientes pero conectados entre sí. El primer subproceso que se muestra es el de realización del encargo de la pizza, que se corresponde con el contenedor identificado como cliente del modelo BPMN anteriormente mostrado.

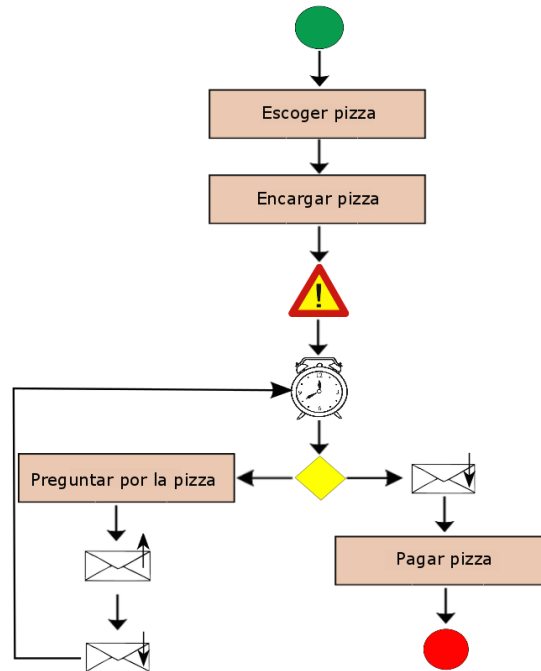


Figura 73: Subproceso de realización del encargo de la pizza

La parte del proceso que representa la interacción del vendedor de pizza o restaurante ha sido dividida en 3 subprocesos, número que se corresponde con la cantidad de expertos que intervienen en la recepción del pedido de la pizza, la preparación de la misma y su reparto al domicilio del cliente. Estos tres subprocesos, que se muestran en las capturas incluidas a continuación, se han denominado de la siguiente forma: subproceso de tratamiento del encargo, subproceso de preparación del encargo y subproceso de reparto del encargo.

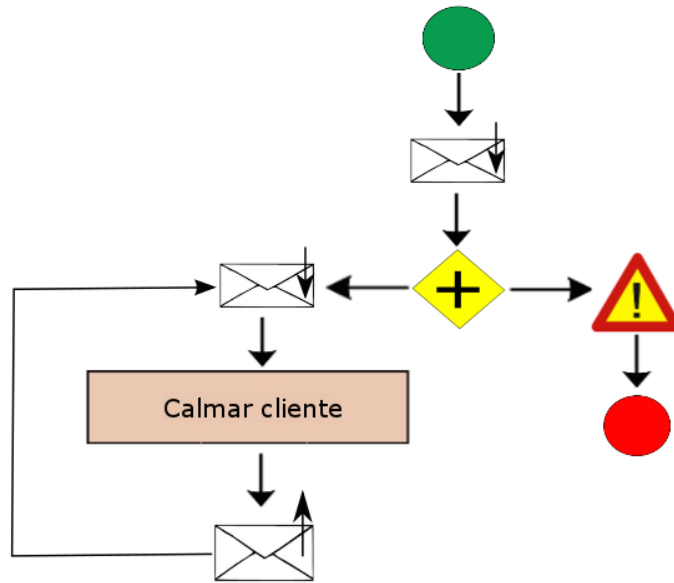


Figura 74: Subproceso de tratamiento del encargo

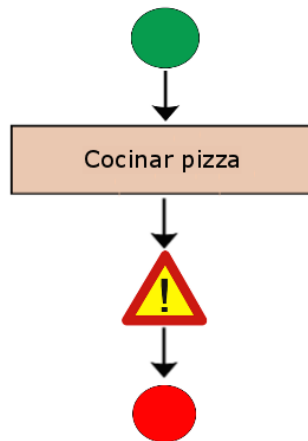


Figura 75: Subproceso de preparación del encargo

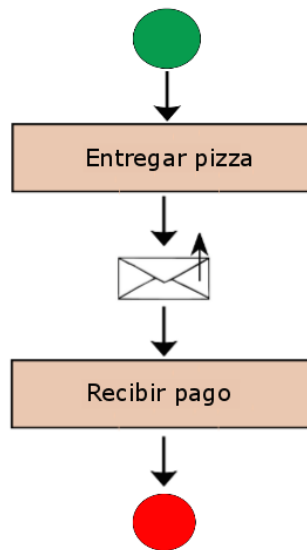


Figura 76: Subproceso de reparto del encargo

Tal y como se observa en las imágenes de los subprocesos anteriores, el otro cambio significativo es el método de comunicación utilizado entre los subprocesos, ya que en el caso de la metodología BPLOM se utilizan eventos de señal en los subprocesos de recepción del encargo y preparación del encargo para indicar al siguiente subproceso en qué momento debe iniciarse. El intercambio de mensajes entre ambos subprocesos se realiza de forma muy similar en ambos casos, ya que si bien en BPMN se utiliza el flujo de mensajes para la realización de este intercambio en BPLOM se utilizan los eventos de mensaje salientes y entrantes para representar la misma función.

A partir de los modelos construidos para ilustrar este ejemplo es posible observar una de las ventajas que supone la utilización de la metodología BPLOM para los expertos del dominio sin conocimientos técnicos y de modelado: la facilidad de comprensión de los modelos en comparación con los de BPMN. Al apostar por la separación de los procesos en función de los expertos que sean propietarios de los mismos se ha conseguido que los modelos de proceso resultantes sean más simples de comprender, ya que solamente se incluyen detalles que el experto conoce perfectamente y, además, existen menos entidades en un mismo modelo.

## 6.6 CONCLUSIONES

Este capítulo recoge la definición de la metodología de modelado BPLOM, que se basa en la adaptación del conjunto de símbolos de modelado al nivel de conocimiento técnico y de modelado que posee el experto del dominio que la utiliza. La metodología BPLOM está estructurada en niveles de tal forma que con cada nuevo nivel que se avanza se introducen nuevos símbolos y se produce un aumento tanto de la expresividad del lenguaje de modelado como de la complejidad resultante de su uso.

A través del ejemplo del proceso de encargo de una pizza se ha realizado una comparación entre los modelos resultantes de la aplicación de la notación estándar BPMN y la metodología BPLOM propuesta en este trabajo, mostrando las principales diferencias existentes entre los modelos en función de las características de ambas propuestas.

Tras esta introducción de la propuesta se procederá a mostrar su aplicación en el modelado de procesos de negocio de distintas características en el caso de estudio incluido a continuación.





# CAPÍTULO 7

## CASO DE ESTUDIO

---

---

### 7.1 INTRODUCCIÓN

Al finalizar la definición de una metodología de este tipo es necesario poner la propuesta a prueba a través de su utilización en casos de uso reales. En un primer momento se mostrará un pequeño estudio por los tipos de aplicaciones que se creen más interesantes para la aplicación de la propuesta. Después, se utilizará la metodología por niveles para el modelado de tres tipos de procesos relacionados con los tipos de aplicaciones incluidos en el estudio anterior:

- Dos procesos de negocio tradicionales de distinta complejidad y ámbito, intentando mostrar el grado de expresividad de la metodología así como la adaptación de la misma a procesos de distinta clase.
- Varios procesos que representan la lógica o el funcionamiento de diversos tipos de videojuegos.
- Un proceso que representa la visualización de un catálogo de productos y la realización de un pedido.

Con esta selección de procesos se intentan demostrar las siguientes cualidades de la metodología propuesta: la capacidad expresiva, modelando procesos con grados de complejidad diferentes; y la independencia del dominio, permitiendo modelar procesos de dominios con diferentes necesidades.

La mecánica para mostrar la aplicación de la metodología a los procesos anteriormente mencionados será la siguiente: en primer lugar se describirán los procesos a través de un texto y posteriormente se irá aplicando la metodología nivel por nivel, mostrando como queda cada proceso con cada uno de los niveles.

## 7.2 TIPOS DE APLICACIONES PARA LA APLICACIÓN DE LA METODOLOGÍA

El dinamismo que rodea al mundo de la informática, donde surgen nuevas tecnologías y tipos de dispositivos con bastante frecuencia, invita a seleccionar un conjunto de tipos de aplicaciones a las que ofrecer soporte cuando se acomete un trabajo como el reflejado en este trabajo de investigación. Así, a la hora de aplicar la metodología de modelado por niveles que se propone en esta investigación se realizó un pequeño recorrido por los tipos o géneros de aplicaciones con mayor interés en la actualidad para decidir cuáles serían las tecnologías para las que los prototipos ofrecerían soporte.

En primer lugar se estableció que sería interesante ofrecer la capacidad de implantar la metodología BPLOM en el modelado y generación de aplicaciones informáticas con carácter empresarial o profesional. La aparición de la Web a finales del siglo XX y su popularización ha supuesto un gran cambio de la informática a nivel profesional, donde la mayor parte de las empresas han cambiado su forma de trabajar: desaparecen las estaciones de trabajo con software de escritorio que ofrece soporte a las tareas propias de la empresa y surgen aplicaciones web que ofrecen la misma funcionalidad a un coste menor. Es por esto que se decidió que la metodología fuera utilizada para modelar procesos de negocio existentes en el mundo empresarial y que los prototipos fueran capaces de generar aplicaciones web que representaran la funcionalidad descrita en los modelos. Así, el primer ejemplo (*véase epígrafe 7.3*) de utilización de la metodología se centra en el modelado de los procesos de gestión de incidencias y contratación de personal, dos procesos de negocio presentes en el mundo empresarial.

El segundo campo de aplicación de la metodología que se ha considerado verdaderamente interesante es el modelado y la generación de aplicaciones para dispositivos móviles. La aparición de los llamados smartphones, o teléfonos móviles inteligentes, ha supuesto una de las revoluciones informáticas del siglo XXI. Equipados con procesadores potentes, pantallas con buen nivel de resolución y ofreciendo una autonomía aceptable, los smartphones han supuesto un gran cambio debido a su capacidad para ejecutar aplicaciones complejas y a la posibilidad de disponer de conexión a Internet a través de las redes móviles. La industria del videojuego es uno de los grande beneficiados de la aparición de estos dispositivos, ya que la potencia que pueden desplegar y la movilidad que otorgan al jugador los hacen perfectos para jugar en cualquier parte. Es por esto que las tiendas de aplicaciones se han llenado de videojuegos para este tipo de dispositivo y se considera adecuado ofrecer la posibilidad de que la metodología BPLOM sirva para modelar y generar este tipo de aplicaciones, contando en este caso con el apoyo de la aplicación GADE4ALL.

Siguiendo con el campo de los teléfonos inteligentes, otro de los géneros de aplicaciones que han tenido cierto impacto en este tipo de dispositivos es el de las aplicaciones de catálogo de productos. Debido a la posibilidad de conexión a internet de la que disponen estos terminales y haciendo uso de las plataformas de comercio electrónico que tienen auge en estos momentos, existe la posibilidad de ofrecer una experiencia completa de compra: navegación por los productos de un catálogo, consulta de los detalles del producto y creación de un pedido. En la actualidad, tal y como se comenta en el *epígrafe 7.5*, existen una serie de productos software comerciales que ofrecen a las empresas la posibilidad de crear aplicaciones para mostrar sus catálogos de productos en los

dispositivos móviles. Sin embargo, la mayor parte de estos productos software no crean aplicaciones nativas para los dispositivos y no permiten realizar el pedido a través de la propia aplicación sino que crean catálogos utilizando documentos PDF encastrados en una aplicación de móvil o incluso a través de componentes HTML5. Debido a estas deficiencias creemos que es posible aplicar la metodología de modelado por niveles para modelar el proceso de visualización y compra a través de un catálogo y generar aplicaciones móviles que ofrezcan soporte para este tipo de actividad de forma nativa.

### 7.3 MODELADO DE PROCESOS DE NEGOCIO TRADICIONALES CON LA METODOLOGÍA

El primer caso de uso real al que se enfrenta la metodología de modelado por niveles para el modelado de procesos de negocio está compuesto por dos procesos de negocio: la gestión de incidencias y la selección de personal. Ambos procesos son los que representan estos sistemas en la empresa Isastur **[Isastur]**, colaboradora en este apartado del caso de estudio para este trabajo.

Es necesario señalar que la definición de los procesos de negocio tradicionales para el nivel 0 de la metodología fue realizada por los expertos de cada uno de los procesos, dentro del marco del trabajo fin de master que conduce al desarrollo de este trabajo de investigación.

#### 7.3.1 Gestión de incidencias

El primer proceso que se pretende modelar con los niveles de la metodología es el de gestión de incidencias. En este proceso participan dos actores independientes: por un lado el cliente, que es el encargado de introducir en el sistema de gestión las distintas incidencias que le surjan a la hora de utilizar el equipamiento del que dispone; por el otro el técnico, que será el encargado de poner solución a las incidencias introducidas por el cliente.

Debido a la existencia de esta dupla de actores, la gestión de incidencias se divide en dos subprocesos paralelos: el proceso de negocio que modela la introducción de las incidencias en el sistema por parte de cliente y el proceso que representa la resolución de las incidencias por parte del técnico.

##### 7.3.1.1 Descripción del proceso

A continuación se incluye una pequeña descripción de los dos subprocesos que componen la gestión de incidencias.

###### 7.3.1.1.1 Subproceso de inserción de incidencias

El subproceso de adición de incidencias comienza cuando el cliente entra al sistema a través de la introducción de su usuario y contraseña. Una vez iniciada la sesión, la aplicación se encarga de buscar en el sistema las incidencias introducidas en el mismo por parte del usuario y las estudia para saber si el usuario tiene alguna incidencia resuelta pendiente de calificar; en caso de que exista alguna incidencia resuelta pendiente de calificar el cliente deberá valorar la resolución del problema realizada por el técnico correspondiente.

La aplicación permitirá al cliente introducir su incidencia en el sistema a través del rellenado de un formulario en el que constan los detalles del problema. Una vez que el usuario selecciona la opción de introducción de la incidencia en el sistema, la aplicación asigna de forma automática esta incidencia a uno de los técnicos; dicha asignación se

realiza en función de la carga de trabajo de los técnicos existentes, que viene determinada por el número de incidencias pendientes de resolver que tenga cada uno.

**7.3.1.1.2 Subproceso de resolución de incidencias**

El otro subproceso que compone el sistema de gestión de incidencias informáticas es el de recepción y resolución de incidencias. Este subproceso comienza con el inicio de sesión del técnico de resolución de incidencias, momento tras el que el sistema le presenta al técnico los detalles de las incidencias que le han sido asignadas.

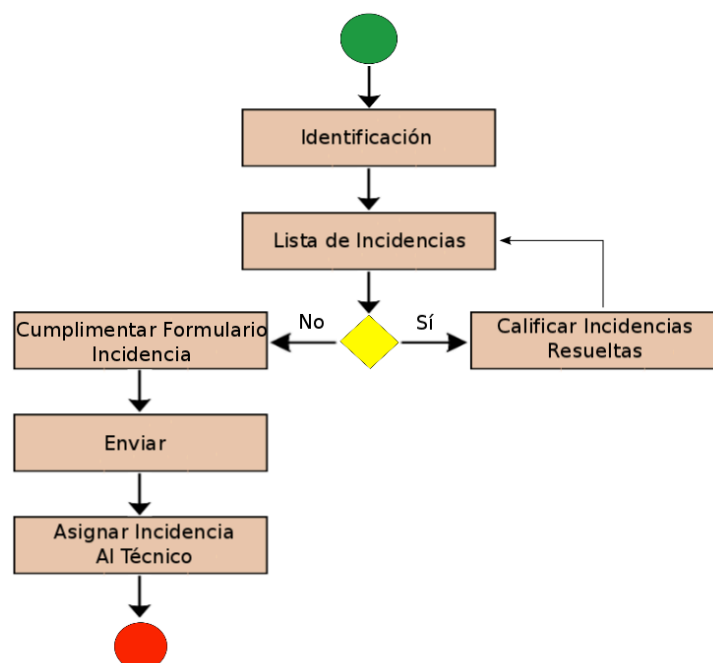
Una vez que el técnico conoce las incidencias que tiene bajo su responsabilidad, puede seleccionar una de ellas para conocer los detalles de la misma y proceder a introducir en el sistema tanto las acciones que ha realizado para resolver la incidencia como el estado en el que queda la incidencia a partir de su actuación.

*7.3.1.2 Aplicación de la metodología de modelado por niveles*

A continuación se realiza un recorrido por el modelado de cada uno de los subprocesos con cada uno de los niveles de la metodología BPLOM.

**7.3.1.2.1 Modelado del proceso utilizando el nivel 0**

Tal y como se puede observar en las figuras incluidas a continuación, el proceso de gestión de incidencias se puede modelar de manera sencilla con el nivel 0 de la metodología, obteniendo como resultado dos modelos de proceso de fácil comprensión para los expertos del mismo.



**Figura 77: Subproceso de inserción de incidencias según el nivel 0**

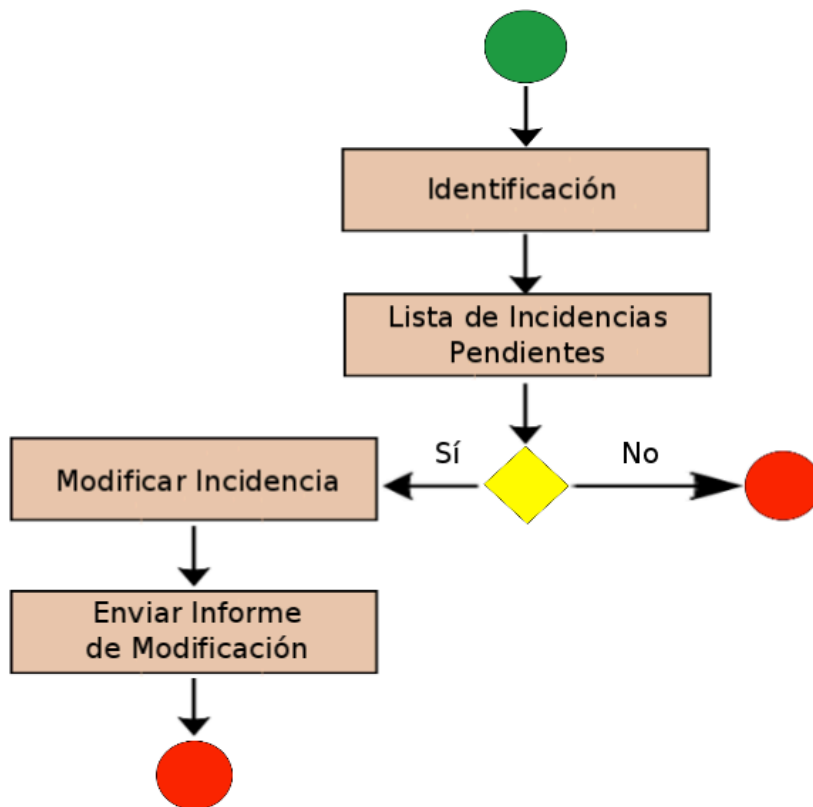


Figura 78: Subproceso de resolución de incidencias según el nivel 0

#### 7.3.1.2.2 Modelado del proceso utilizando el nivel 1

A continuación se incluyen los modelos del proceso de gestión de incidencias informáticas según las entidades ofrecidas en el nivel 1 de la metodología. A través de las entidades ofrecidas como aplicación de la notación de modelado en este nivel, se han introducido una serie de variantes en los modelos del proceso de tal forma que se aprovechan las nuevas capacidades del nivel. Así, el nivel de complejidad mostrado por los modelos es ligeramente superior al mostrado por las capturas anteriores.

Tal y como se observa en las capturas el proceso de inserción de incidencias hace uso de la decisión inclusiva, incluida en el nivel 1 de la metodología, en detrimento de la decisión incluida en el nivel inicial; de esta forma se cambia el concepto del modelo, estableciéndose que en la llegada a la decisión se activa al menos uno de los dos caminos y dejando la opción de que puedan activarse ambos. El proceso de resolución de incidencias queda inalterado debido al hecho de que no es posible que se activen los dos caminos de la decisión a la vez, ya que uno de estos conlleva la terminación del proceso.

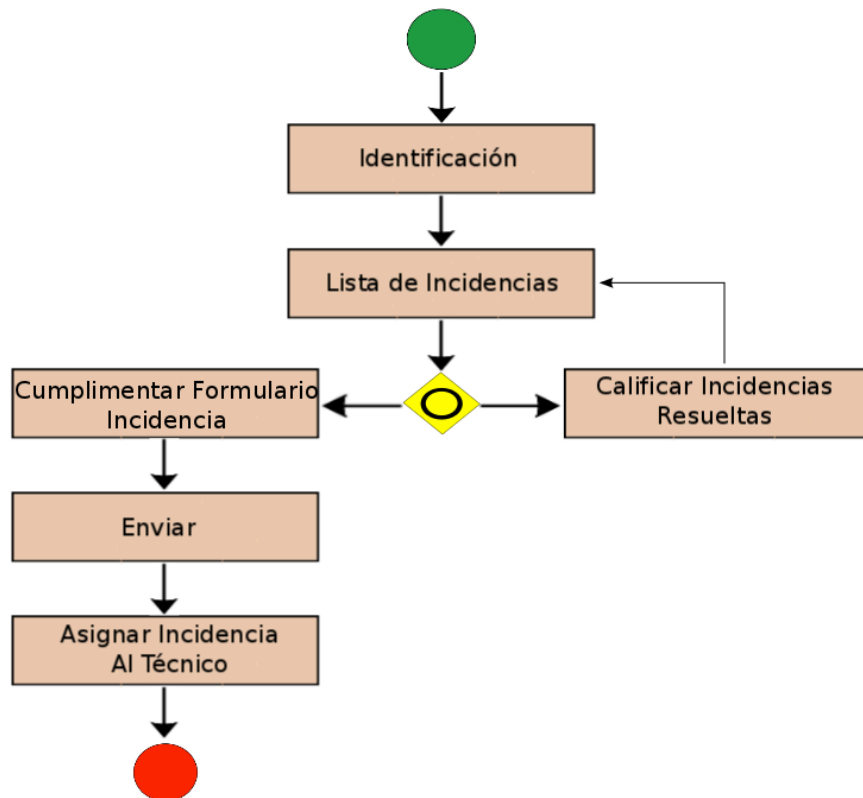


Figura 79: Subproceso de inserción de incidencias según el nivel 1

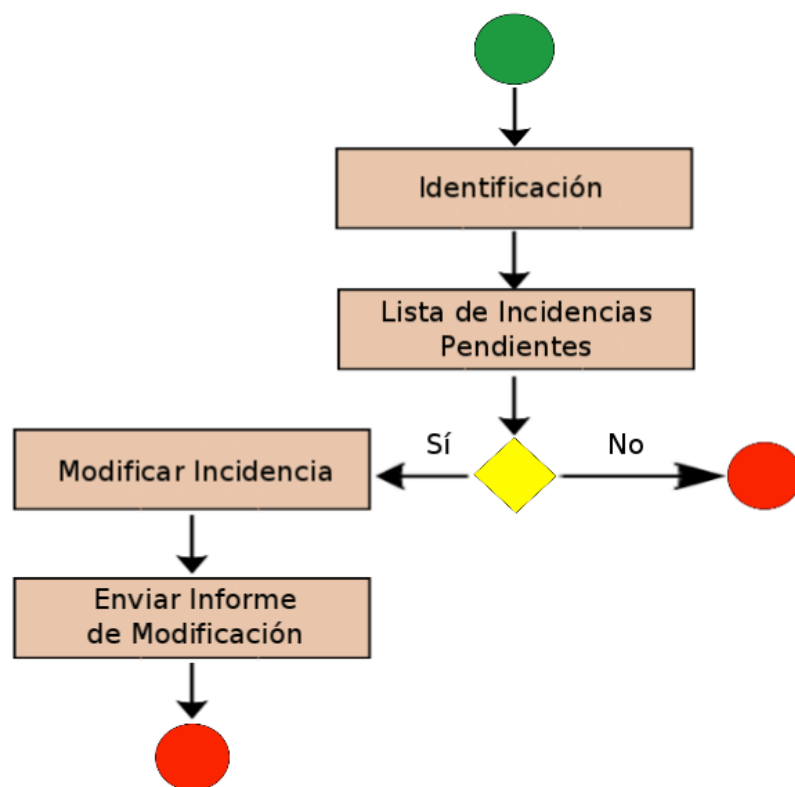


Figura 80: Subproceso de resolución de incidencias según el nivel 1

### 7.3.1.2.3 Modelado del proceso utilizando el nivel 2

A continuación se incluyen los modelos del proceso de gestión de incidencias informáticas según las entidades ofrecidas en el nivel 2 de la metodología. Utilizando los eventos proporcionados como extensión de este nivel de la metodología se ha procedido a aumentar la expresividad de los modelos, añadiendo una serie de funcionalidades que ofrecen mejoras a la hora de la ejecución de los procesos.

Por ejemplo, en el proceso de inserción de incidencias se han añadido los siguientes eventos de mensaje: uno que envía un mensaje al técnico al que se ha asociado una nueva incidencia para avisarle de este hecho y otro que envía al técnico que resolvió una incidencia un aviso de que la resolución ha sido calificada por el usuario que insertó la incidencia en el sistema.

Sin embargo, los cambios más significativos en la aplicación de este nivel los ha sufrido el subproceso de resolución de incidencias. Por un lado, se ha intercambiado la actividad de envío del informe de modificación por un evento de mensaje saliente que emite un mensaje de modificación de la incidencia que le llegará al usuario que ha introducido la incidencia en el sistema. Además, se ha modificado la forma de finalización del proceso en caso de que no existan incidencias pendientes; según los niveles anteriores el proceso finalizaba en caso de no existir incidencias pendientes de resolver pero ese comportamiento ha sido sustituido por una cancelación del proceso, evento que refleja de manera más precisa la naturaleza de un proceso de estas características ya que la falta de incidencias pendientes de tratamiento se interpreta más como una causa de cancelación del proceso que como una finalización normal del mismo.

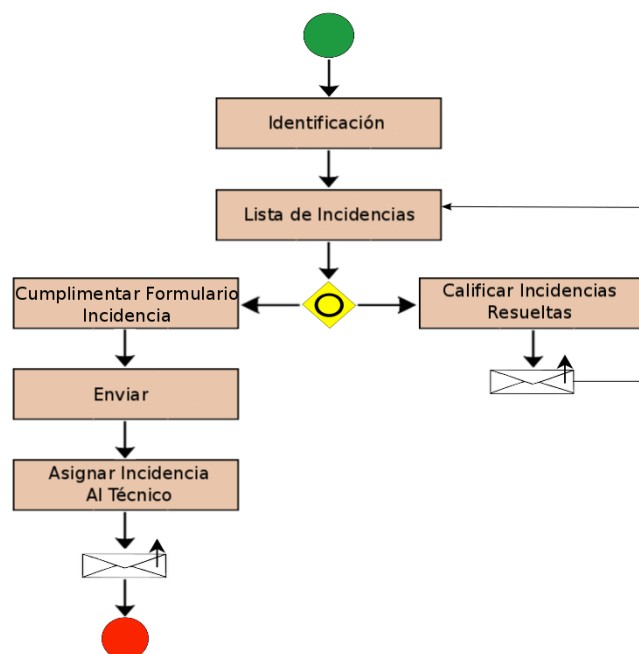


Figura 81: Subproceso de inserción de incidencias según el nivel 2



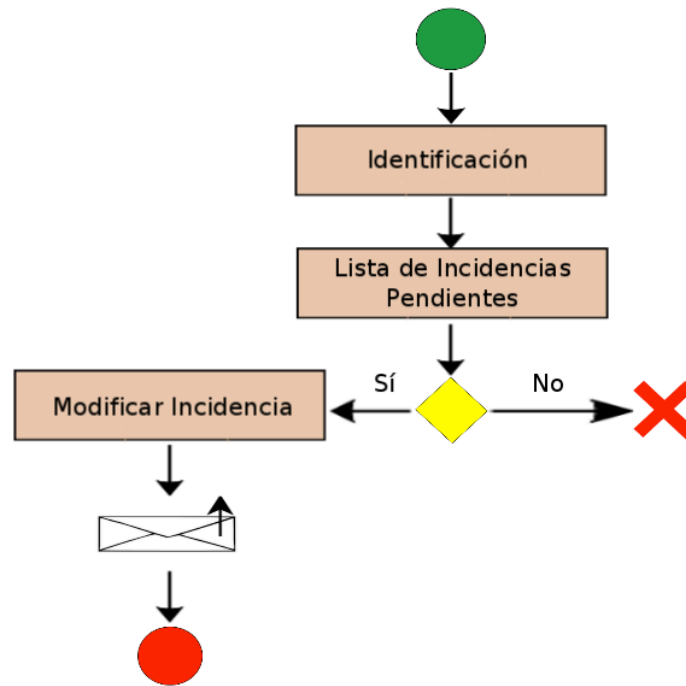


Figura 82: Subproceso de resolución de incidencias según el nivel 2

#### 7.3.1.2.4 Modelado del proceso utilizando el nivel 3

Tal y como se puede apreciar en las figuras siguientes, los subprocesos que componen el sistema de gestión de incidencias informáticas se han visto modificados de dos formas distintas a través de la utilización del nivel 3 de la metodología presentada. La primera modificación realizada es patente en la primera actividad de ambos subprocesos, la identificación; esta tarea, que hace referencia al hecho de que cualquier usuario de los sistemas debe validarse como tal a través de la introducción de su cuenta de usuario y su contraseña, es considerada una actividad de llamada. Esta consideración se realiza porque la identificación del usuario es una tarea que se repite en todos los procesos, pudiendo representarse como una llamada a una actividad comúnmente repetida.

Por otro lado, la otra modificación sufrida por los subprocesos radica en la inclusión de la actividad subproceso en ambos modelos de proceso. Un subproceso representa aquel conjunto de tareas que es necesario acometer de forma ininterrumpida para el correcto devenir del proceso y en este caso se ha utilizado dicha entidad en dos puntos: por un lado, en el conjunto de tareas que recogen la información de una incidencia, la meten en el sistema y se la asignan a un técnico; y, por el otro lado, en el conjunto de actividades que permiten a un técnico modificar una incidencia que tiene asignada y enviar el informe de modificación correspondiente. De esta forma se garantiza la ejecución completa de aquellas actividades más importantes en un proceso.

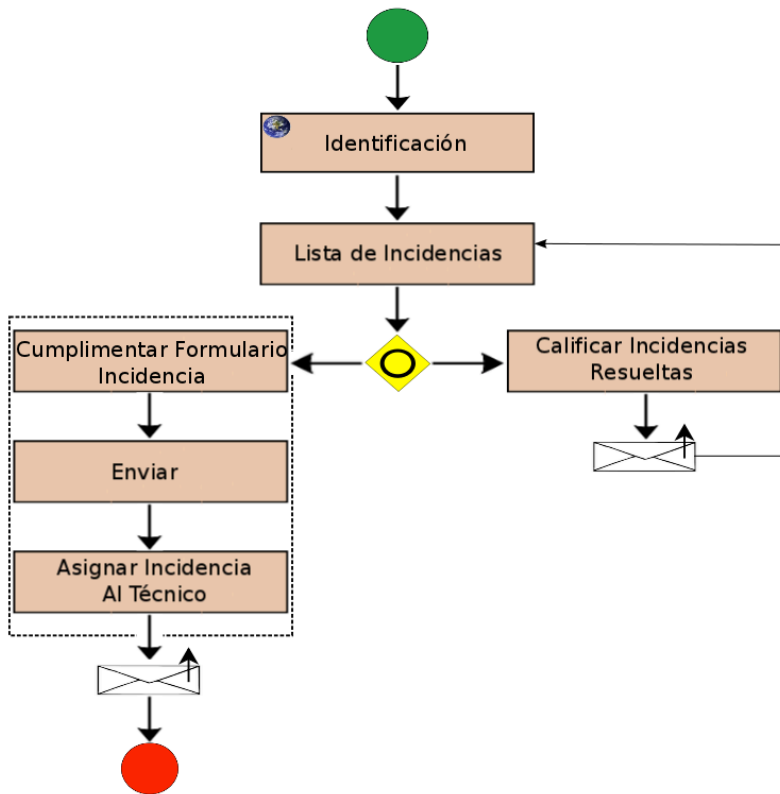


Figura 83: Subproceso de inserción de incidencias según el nivel 3

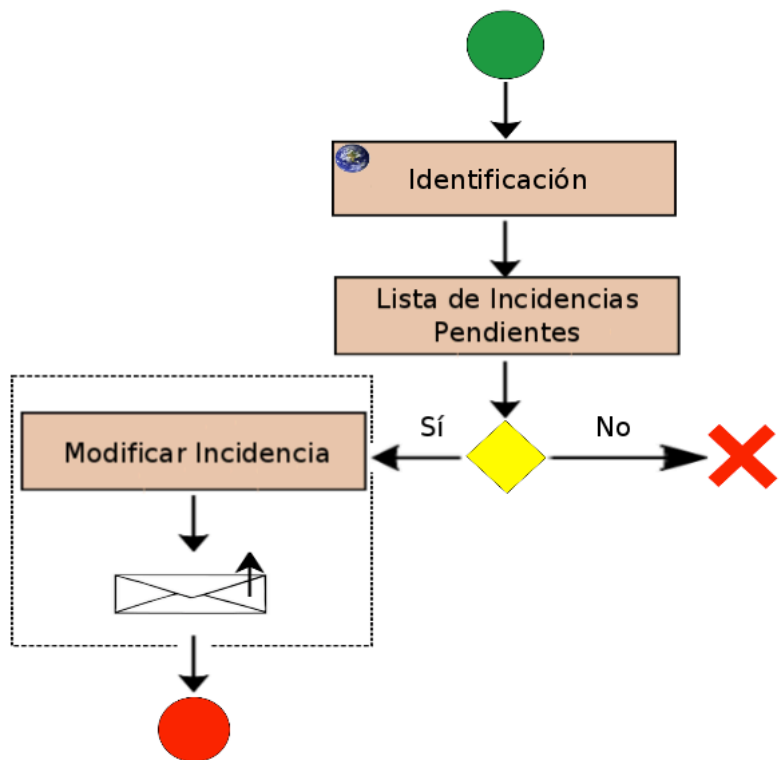


Figura 84: Subproceso de resolución de incidencias según el nivel 3

7.3.1.2.5 Modelado del proceso utilizando el nivel 4

Las dos imágenes incluidas a continuación muestran los subprocesos que representan al sistema de gestión de incidencias informáticas una vez se han aplicado todos los niveles de la metodología.

Los cambios aplicados en esta última modificación, la correspondiente al cuarto y último nivel, traen consigo la utilización de todas las entidades de datos: el objeto de datos para representar a la incidencia creada a partir de los datos introducidos en el formulario del primer subproceso, la colección de objetos para representar a las listas de incidencias (tanto las que son propiedad del usuario en el primer proceso como las que están asignadas al técnico en el segundo) que se manejan en ambos subprocesos y el almacén para representar las escrituras de datos que se realizan para introducir los datos correspondientes en el sistema.

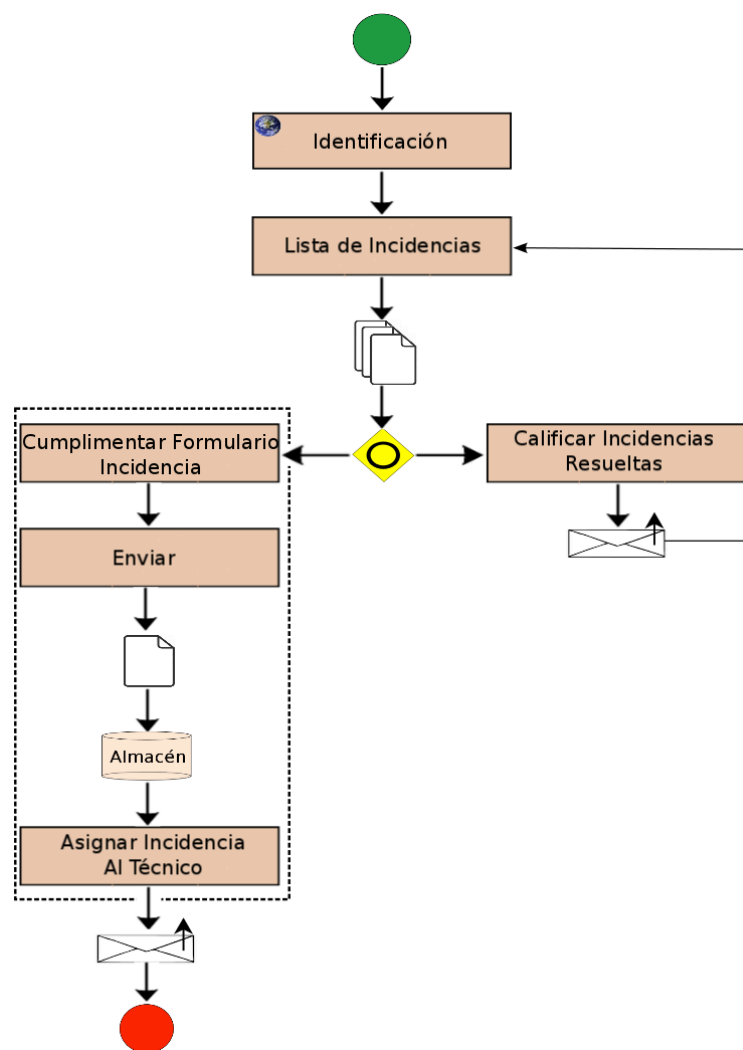


Figura 85: Subproceso de inserción de incidencias según el nivel 4

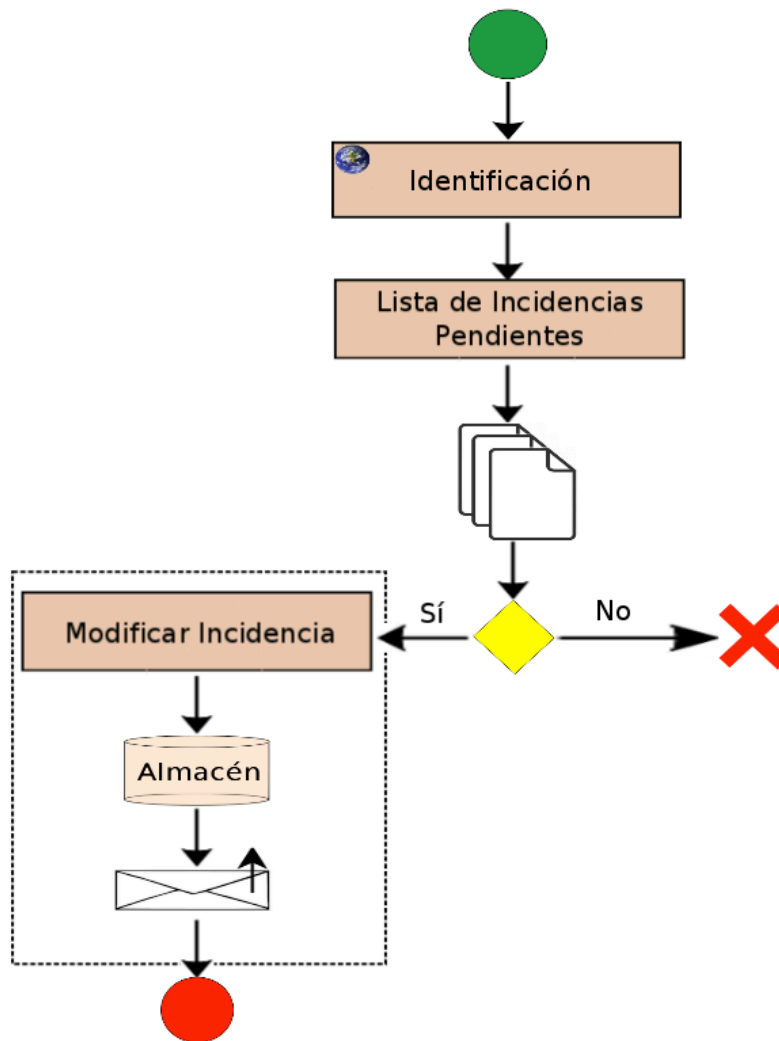


Figura 86: Subproceso de resolución de incidencias según el nivel 4

### 7.3.2 Selección de personal

El proceso de selección de personal es el sistema que rige la incorporación de nuevos trabajadores a una empresa. Una gran variedad de actores contribuyen, en una o más etapas, a la realización de este proceso; entre estos actores se encuentran los siguientes: el jefe de recursos humanos y los directores o jefes de secciones y departamentos, también conocidos como cliente interno.

A partir de la existencia de este elevado número de actores y de la complejidad de las distintas etapas por las que transcurre un proceso de selección de personal, el proceso de contratación de personal ha sido dividido en 7 subprocesos: solicitud de nuevo empleado, valoración de solicitud, valoración de proceso, reclutamiento, valoración de candidatos por recursos humanos, valoración de candidatos por cliente interno y oferta de contratación.

### 7.3.2.1 Descripción del proceso

A continuación se incluye una descripción de cada uno de los subprocesos que componen el proceso de selección de personal de una empresa.

#### 7.3.2.1.1 Subproceso de solicitud de nuevo empleado

El subproceso de solicitud de un nuevo empleado es iniciado por los directores de división o los jefes de unidad/departamento en el momento en el que necesitan contratar un nuevo empleado para cubrir una vacante. Una vez identificado en el sistema, el actor que quiera solicitar un nuevo empleado puede consultar las solicitudes de nuevos empleados que ha introducido en el sistema para después pasar a crear una nueva solicitud. La creación de la solicitud puede realizarse de dos formas distintas: por un lado, en caso de que no exista el perfil deseado se podrá cumplimentar un cuestionario con los detalles del perfil y, por el otro lado, se podrá optar por seleccionar un puesto o tipo de puesto en caso de que el perfil exista en el sistema. En ambos casos, una vez realizadas las tareas pertinentes se producirá el envío de la solicitud al departamento de recursos humanos.

#### 7.3.2.1.2 Subproceso de valoración de solicitud

El subproceso de valoración de una solicitud es la segunda etapa del proceso de selección de personal; una vez introducida la solicitud en el sistema, el responsable del departamento de recursos humanos se identificará y procederá a hacer un análisis del perfil solicitado. En caso de que el responsable concluya que el perfil es adecuado, lo validará y se producirá el envío de un mensaje de estado a la persona que haya introducido la solicitud. En caso de que el perfil no sea validado, se procede a denegar la solicitud de un nuevo empleado.

#### 7.3.2.1.3 Subproceso de valoración del proceso de selección

Una vez que se haya valorado la solicitud positivamente, validándose el perfil del empleado que se haya introducido en el sistema, será el momento de valorar el proceso de selección que se realizará para cubrir el puesto en cuestión; siendo dicha tarea realizada de nuevo por el responsable del departamento de recursos humanos.

El subproceso de valoración del proceso comienza con la decisión de cubrir la vacante a través de una promoción interna o no. En caso de que se cubra con una promoción interna se pasa la solicitud a la jefatura correspondiente y se espera la respuesta de ésta, y en caso de que el proceso interno sea aprobado se enviará la oferta de empleo al departamento correspondiente. Si la vacante no se cubriera a través de un sistema de promoción interna habrá que decidir si se externaliza el proceso; si el proceso se externaliza se seleccionará la consultora correspondiente y se enviará el perfil validado en el paso anterior para que sea la consultora la que defina el medio en el que se ofertará el puesto; si por el contrario se decide no externalizar el proceso será la propia empresa la que defina y elija el medio donde se publicará la oferta de trabajo. En ambos casos será necesario que el anuncio sea validado, produciéndose una modificación del mismo hasta que éste sea aprobado.

#### **7.3.2.1.4 Subproceso de reclutamiento**

El siguiente subproceso en el sistema de selección de personal es el de reclutamiento, que radica en la confección de una lista de candidaturas a la oferta de trabajo aprobada en el paso anterior. Será el responsable de recursos humanos el encargado de identificarse y proceder a la recepción de las candidaturas correspondientes a través de los distintos medios de recepción de currículum que disponga la compañía; en este caso los medios de recepción disponibles son un cuestionario web propio y los portales de empleo existentes en Internet. Todas las personas que hayan optado por responder a la oferta en los medios mencionados serán incluidas en la primera lista de candidatos.

#### **7.3.2.1.5 Subproceso de valoración de candidatos por recursos humanos**

La valoración de los candidatos por parte de recursos humanos es la primera valoración que se realiza de los candidatos dentro del proceso de selección de personal. Una vez identificado, el responsable de recursos humanos realizará una preselección de candidatos tras la que será necesario comprobar si existen candidatos válidos. En caso afirmativo, el responsable cita a los candidatos para una entrevista y los valora en base a una entrevista personal y a unos test psicotécnicos, realizándose un informe de cada candidato según la información recogida. Al finalizar el proceso de valoración, se evalúa de nuevo la existencia de candidatos válidos y en caso de que existan se procede a crear una lista de candidatos para el cliente interno y se le envía un mensaje que le avisa de la elaboración de dicha lista.

#### **7.3.2.1.5 Subproceso de valoración de candidatos por cliente interno**

Después de que el responsable de recursos humanos haya realizado su valoración y haya enviado la lista de candidatos al cliente interno, es hora de que éste realice su propia valoración. Tras la identificación el cliente interno realiza una valoración de las candidaturas en base a los datos del currículum y los informes y en el caso de que tras esta valoración tenga candidaturas válidas procederá a realizar sus propias entrevistas.

Como resultado de estas entrevistas realizará una nueva valoración de los candidatos y en caso de que existan candidatos válidos se procederá a configurar la lista de candidatos que optan al puesto antes de la última fase del proceso de selección.

#### **7.3.2.1.6 Subproceso de oferta de contratación**

La última fase del proceso de selección de personal es la relativa a la realización de las propuestas de empleo a los candidatos y la contratación de alguno de ellos para ocupar el puesto que se oferta. Tras su identificación, el responsable de recursos humanos obtendrá el listado de candidatos finalista y comenzará a preparar la propuesta para el primero de ellos; tras presentar la propuesta al cliente interno, que le establecerá el margen de negociación, le presentará la propuesta al candidato, que tendrá dos opciones: rechazar la oferta, acción tras la que se revisará la propuesta realizada con el fin de conocer si existe el margen para realizarle una propuesta mejorada al candidato actual o si será necesario revisar la lista de candidatos finalistas y pasar al siguiente candidato de la lista; o aceptar la oferta, momento que implica el envío al departamento de nóminas de las condiciones de contratación del nuevo empleado.

### 7.3.2.2 Aplicación de la metodología de modelado por niveles

A continuación se procede a aplicar la metodología de modelado por niveles propuesta en este trabajo de investigación a los subprocesos que componen el proceso de selección de personal.

#### 7.3.2.2.1 Modelado del proceso utilizando el nivel 0

El resultado de la aplicación del nivel más básico de la metodología a los 7 subprocesos que componen el sistema de selección de personal se muestra en las siguientes imágenes.

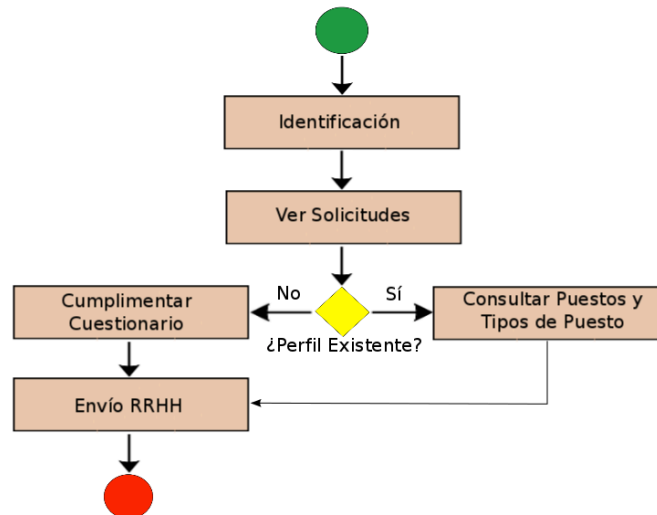


Figura 87: Subproceso de solicitud de nuevo empleado según el nivel 0

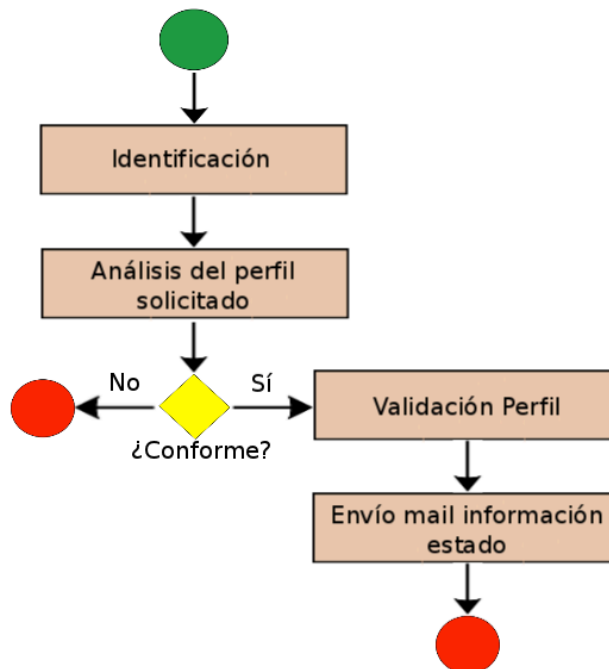


Figura 88: Subproceso de valoración de solicitud según el nivel 0

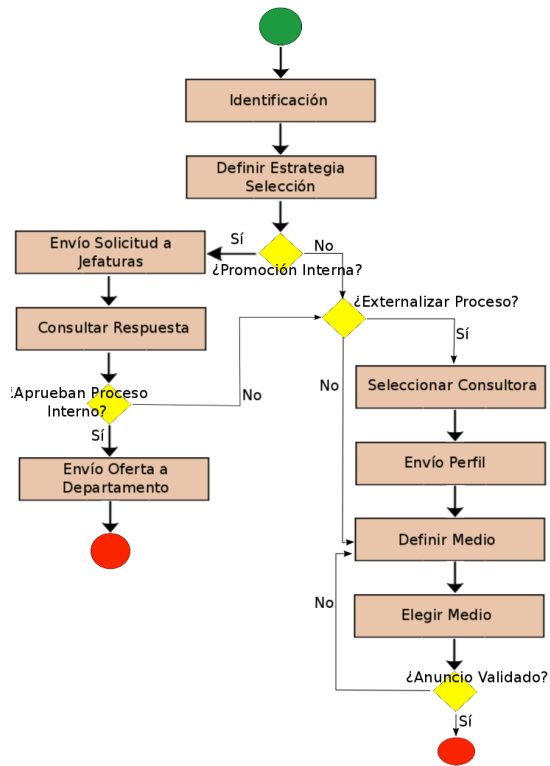


Figura 89: Subproceso de valoración de proceso según el nivel 0

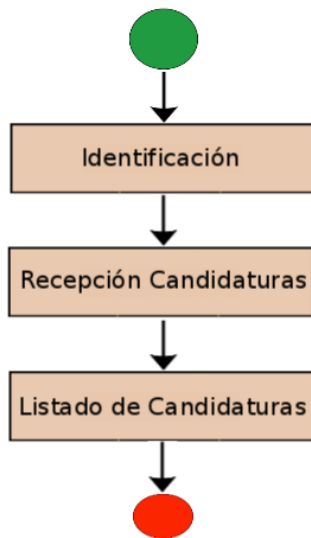


Figura 90: Subproceso de reclutamiento según el nivel 0



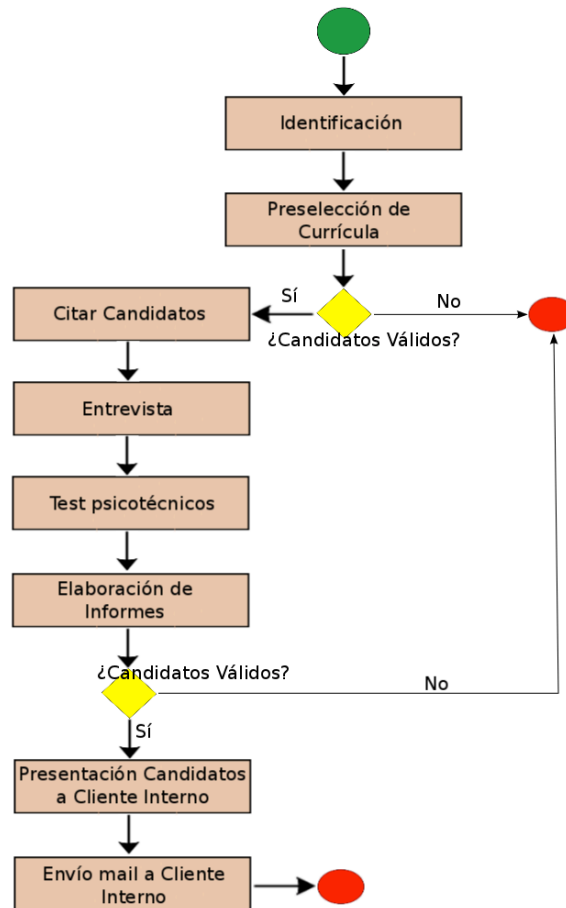


Figura 91: Subproceso de valoración de candidatos por RR.HH según el nivel 0

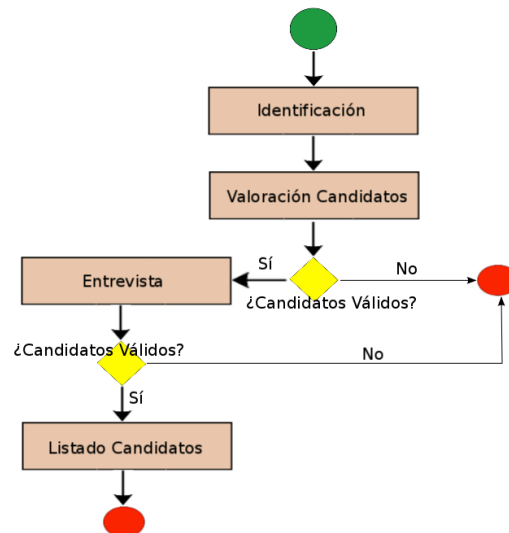


Figura 92: Subproceso de valoración de candidatos por C.I según el nivel 0

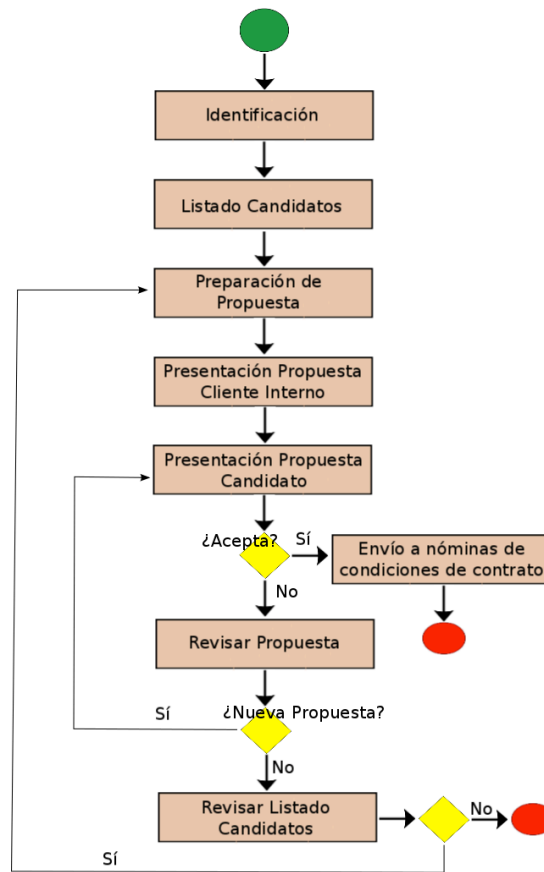


Figura 93: Subproceso de oferta de contratación según el nivel 0

#### 7.3.2.2.2 Modelado del proceso utilizando el nivel 1

A continuación se muestran los modelos de proceso para todos los subprocesos que componen el sistema de selección de personal según las entidades disponibles en el nivel 1 de la metodología.

Según se puede observar en las imágenes de los procesos la ampliación de decisiones que supone el nivel 1 de la metodología no se adapta a las circunstancias de todos los subprocesos (por ejemplo, el subproceso de solicitud de nuevo empleado no ha sufrido ningún cambio), siendo esto debido a las características de los subprocesos en cuanto a su configuración y el tipo de decisiones que incluyen. En la mayoría de los casos, las decisiones que aparecen en los subprocesos cuentan con caminos de ejecución excluyentes, no siendo posible utilizar ni decisiones inclusivas ni paralelas; como ejemplo basta con mencionar la decisión más repetida en los subprocesos que es la que permite establecer si un candidato es válido o no.

Esta circunstancia se repite en la mayoría de los subprocesos del sistema de selección de personal, aunque uno de los subprocesos incluidos en el mismo sí que ha sufrido cambios significativos: el subproceso de reclutamiento. En la descripción textual de los subprocesos se establecía que el reclutamiento obtiene los datos de los posibles candidatos para el puesto a través de dos vías distintas como son los portales de empleo y la web propia. Debido a la expresividad reducida del nivel 0 se había modelado esa característica a través de una única actividad de recepción de candidaturas pero como el

nivel 1 ofrece un aumento de expresividad acorde al escenario de ejecución en paralelo que requiere la recepción de las candidaturas por dos vías al mismo tiempo se ha realizado un cambio sustancial en el proceso; la actividad de recepción de candidaturas se ha desdoblado en dos actividades que se ejecutan de forma paralela y que permiten obtener los datos de las dos fuentes de forma simultánea. Así se logra representar de forma más cercana a la realidad el sistema de reclutamiento del proceso de selección de personal.

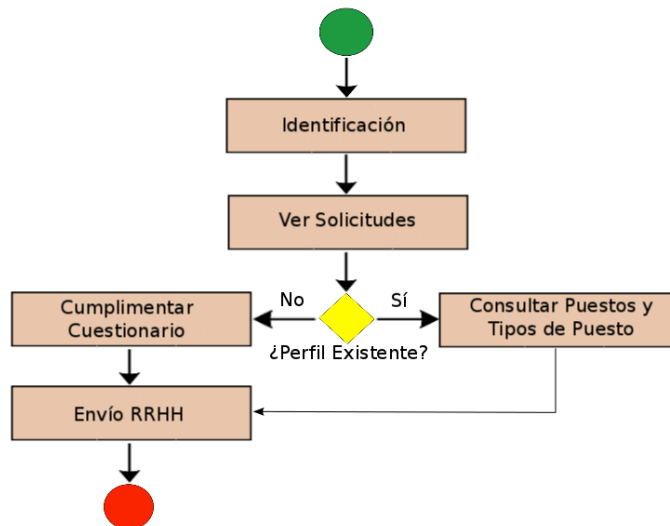


Figura 94: Subproceso de solicitud de nuevo empleado según el nivel 1

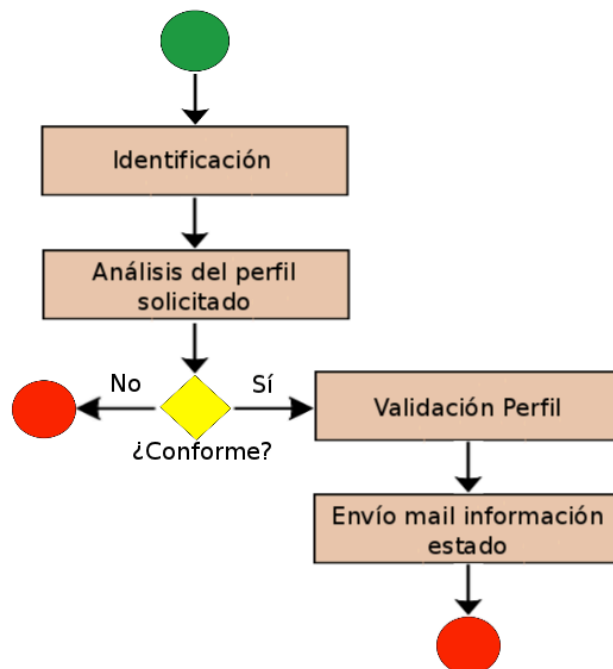


Figura 95: Subproceso de valoración de solicitud según el nivel 1

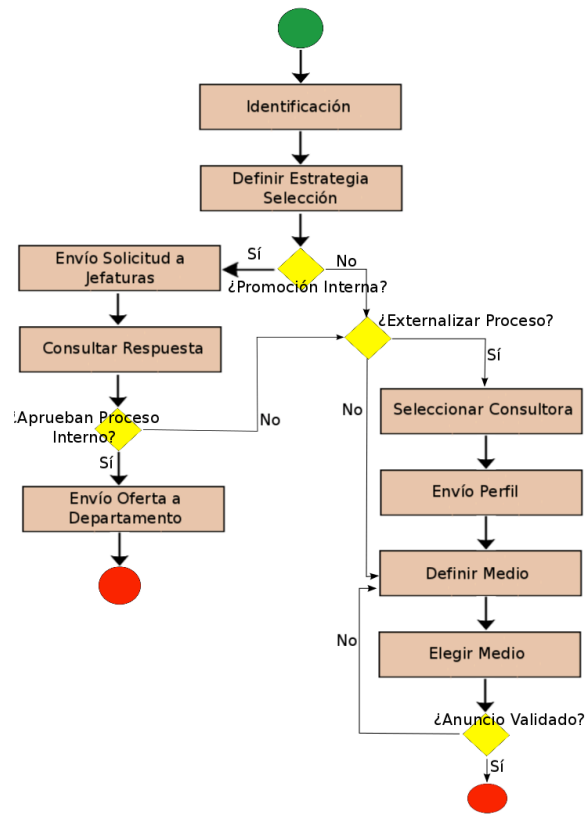


Figura 96: Subproceso de valoración de proceso según el nivel 1

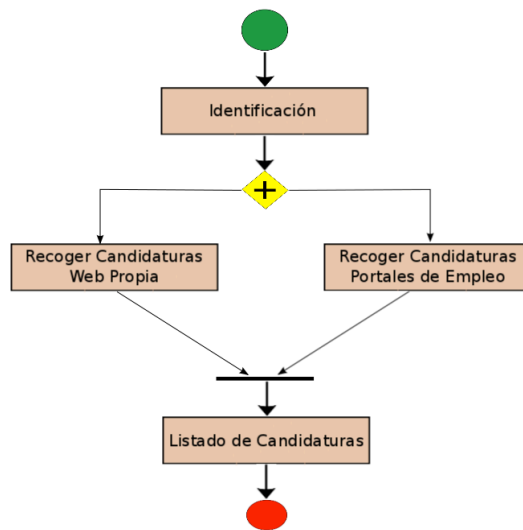


Figura 97: Subproceso de reclutamiento según el nivel 1

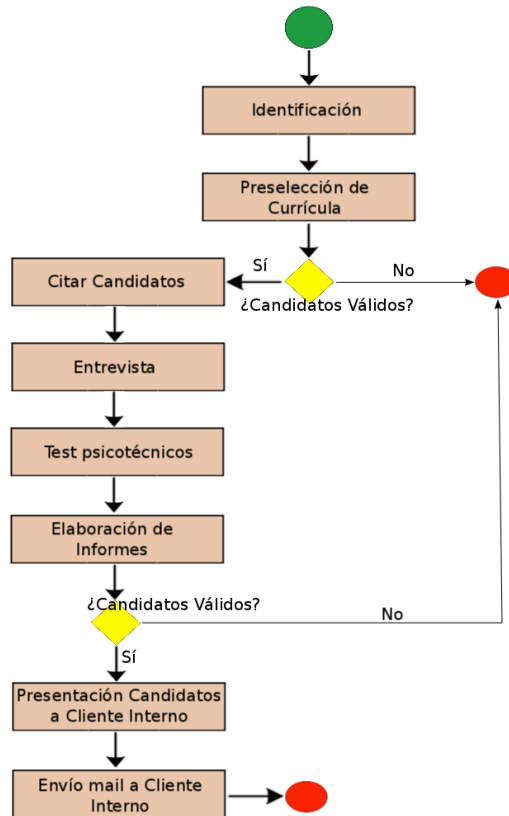


Figura 98: Subproceso de valoración de candidatos por RR.HH según el nivel 1

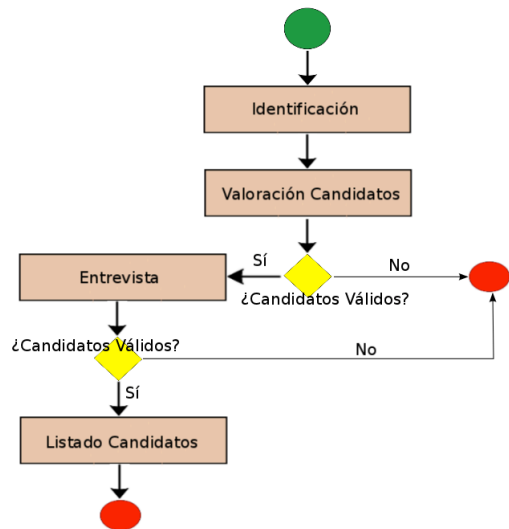


Figura 99: Subproceso de valoración de candidatos por C.I según el nivel 1

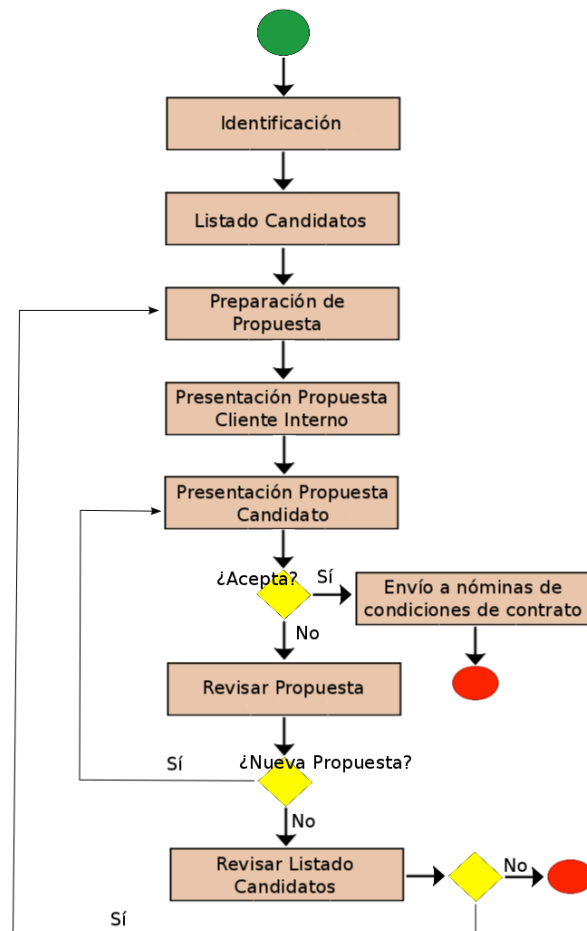


Figura 100: Subproceso de oferta de contratación según el nivel 1

### 7.3.2.2.3 Modelado del proceso utilizando el nivel 2

Las imágenes que se incluyen a continuación se corresponden con el resultado de aplicar la extensión de eventos de la metodología por niveles a los subprocesos que componen el sistema de selección de personal. A diferencia de lo que ocurriera con la aplicación de la extensión de decisiones, los subprocesos del sistema de selección de personal han sufrido cambios significativos con la aplicación de este nivel.

Las modificaciones introducidas con la aplicación de este nivel se fundamentan en la utilización de dos eventos principalmente: el evento de cancelación y el evento de mensaje. El evento de cancelación aparece como modificación en aquellos procesos en los que se comprueba la existencia de candidatos válidos, ya que en caso de no existir candidatos válidos en alguno de los subprocesos del sistema de contratación de personal se procede a cancelar la selección. Por otro lado, el evento de mensaje se utiliza como instrumento de comunicación entre los distintos procesos, enviando mensajes a los distintos actores para informar de la ejecución de los subprocesos y de la necesidad de continuar con el proceso de selección; por ejemplo, cuando se termina la valoración de la solicitud se incluye un evento de mensaje saliente que permite informar del estado de la solicitud.

Otro de los eventos utilizados en uno de los procesos es el evento de señal. En concreto, este evento es utilizado en el último subproceso como mecanismo de aviso para el proceso correspondiente a la inclusión del candidato en plantilla, de tal manera que cuando termina de ejecutarse el subproceso de oferta de contratación se le pasan los datos a este otro proceso y se le envía una señal para que se ejecute con los datos del mensaje.

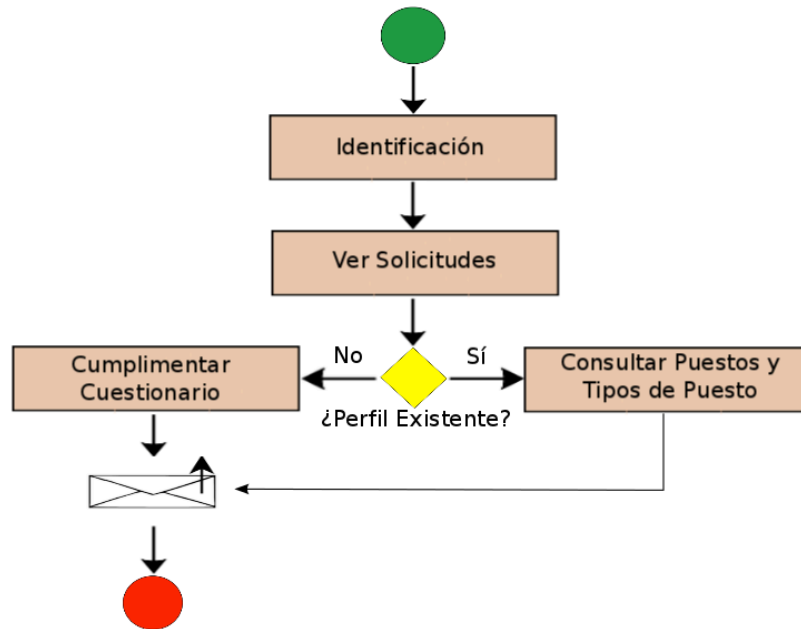


Figura 101: Subproceso de solicitud de nuevo empleado según el nivel 2

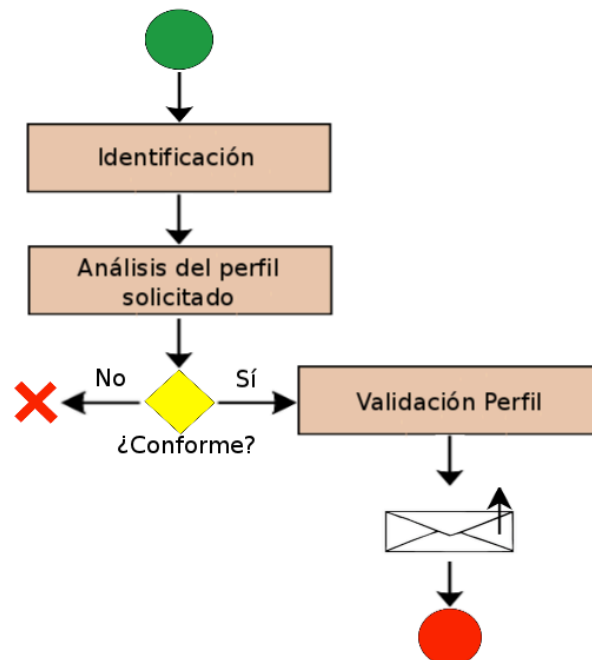


Figura 102: Subproceso de valoración de solicitud según el nivel 2

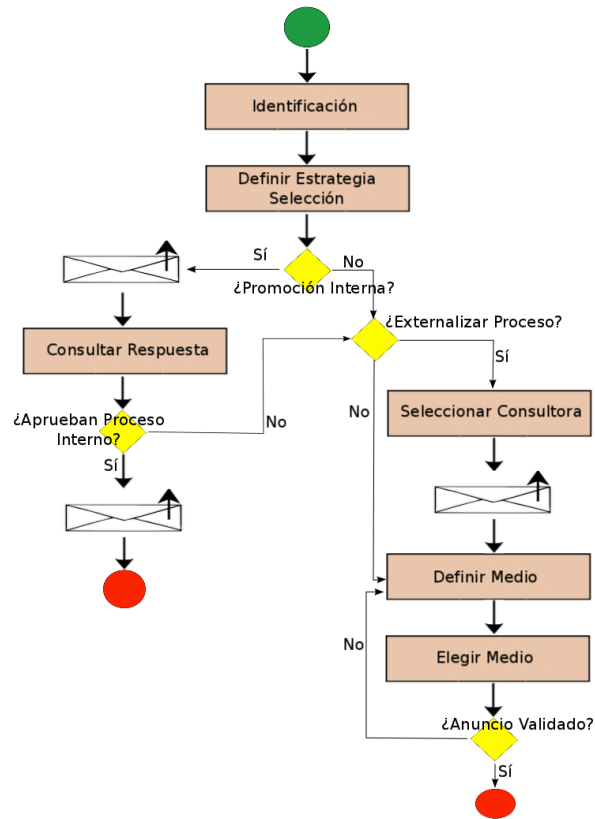


Figura 103: Subproceso de valoración de solicitud según el nivel 2

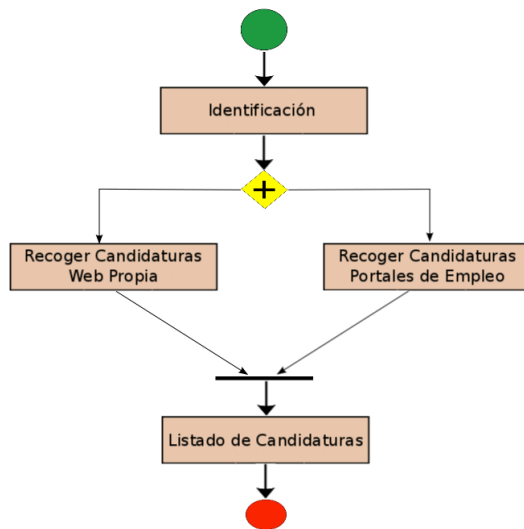


Figura 104: Subproceso de reclutamiento según el nivel 2



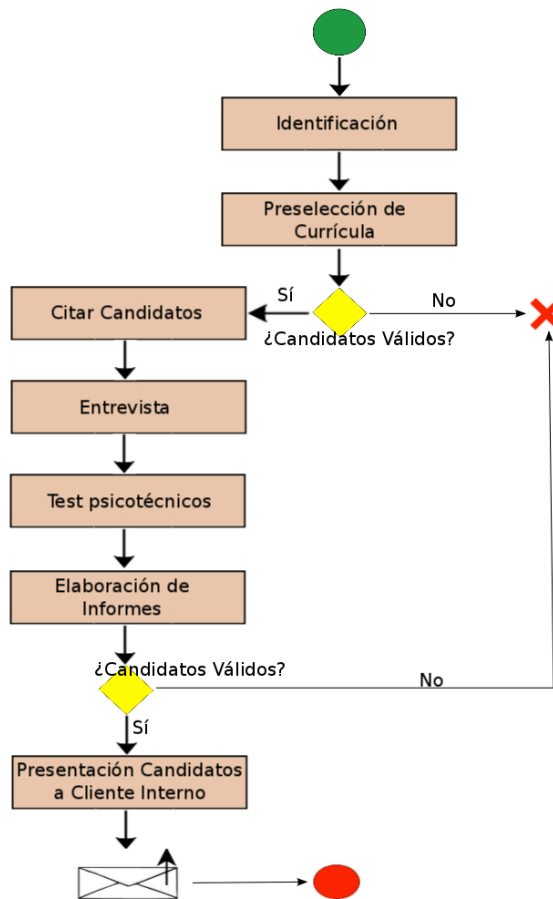


Figura 105: Subproceso de valoración de candidatos por RR.HH. según el nivel 2

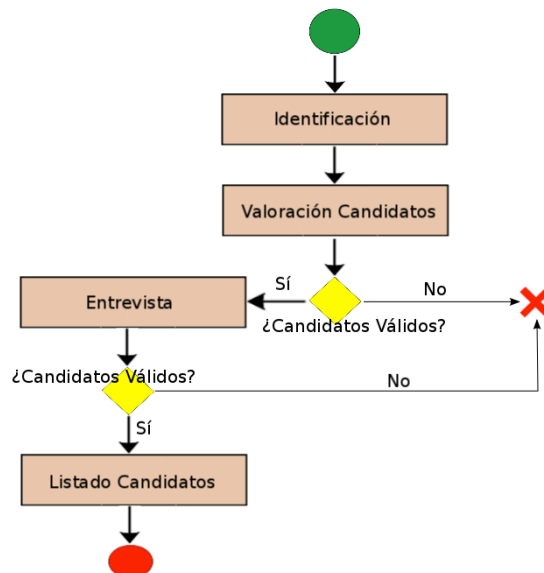


Figura 106: Subproceso de valoración de candidatos por C.I. según el nivel 2

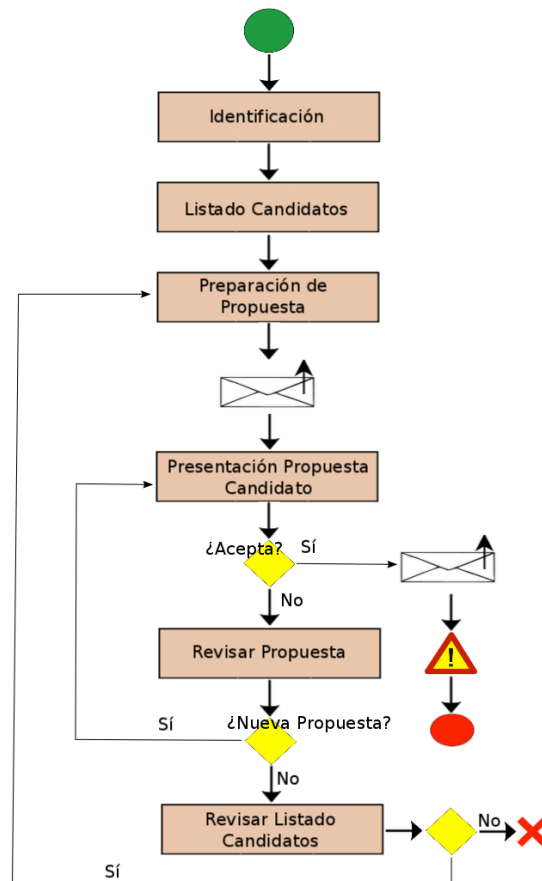


Figura 107: Subproceso de oferta de contratación según el nivel 2

#### 7.3.2.2.4 Modelado del proceso utilizando el nivel 3

Las imágenes que se incluyen a continuación muestran los cambios experimentados por los subprocesos del sistema de selección de personal tras la aplicación de la extensión que supone el nivel 3 de la metodología presentada.

Al igual que ocurriera en el caso del proceso de gestión de incidencias todos los subprocesos han visto modificada su primera actividad, la identificación, debido a que esta se corresponde con una actividad común a varios procesos y por lo tanto se etiqueta como una actividad de llamada. Además, se ha procedido a establecer subprocesos que han de ejecutarse de principio a fin sin interrupciones, como en el caso de la externalización de la publicación de la oferta laboral en el segundo subproceso del sistema de selección de personal o la negociación con el candidato a ocupar la vacante del último subproceso.

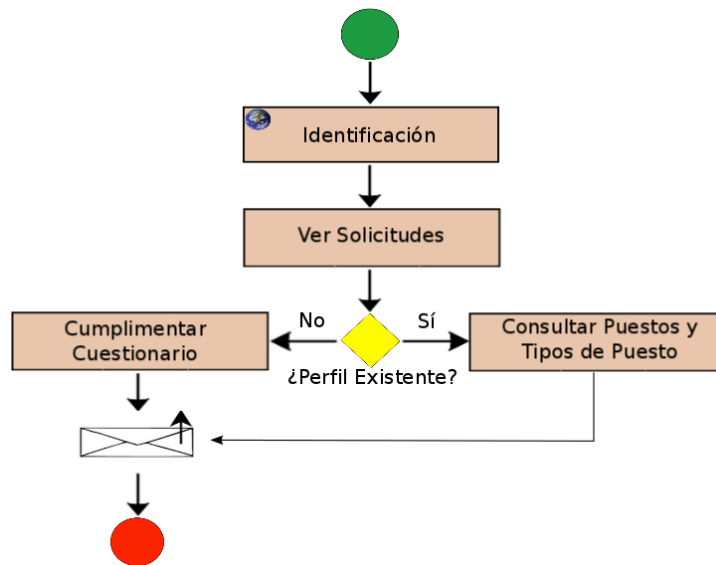


Figura 108: Subproceso de solicitud de nuevo empleado según el nivel 3

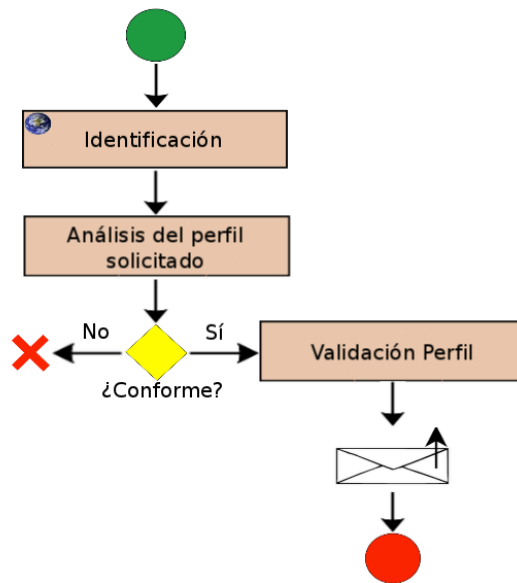


Figura 109: Subproceso de valoración de solicitud según el nivel 3

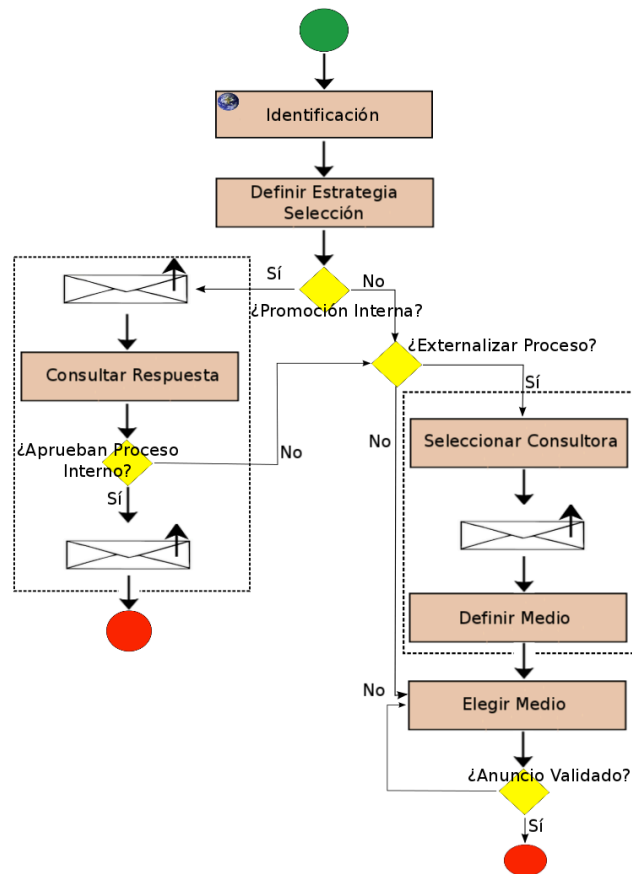


Figura 110: Subproceso de valoración de proceso según el nivel 3

Con respecto al subproceso de valoración del proceso, mostrado en la captura anterior, es necesario señalar que se ha modificado su flujo de ejecución en la parte final del mismo. Al establecer el subproceso de externalización desde la selección de la consultora hasta la definición del medio donde se debe publicar la oferta de trabajo, ha sido necesario establecer la actividad de elección del medio como el punto de continuación en caso de optar por una selección interna de los parámetros de publicación de la oferta.

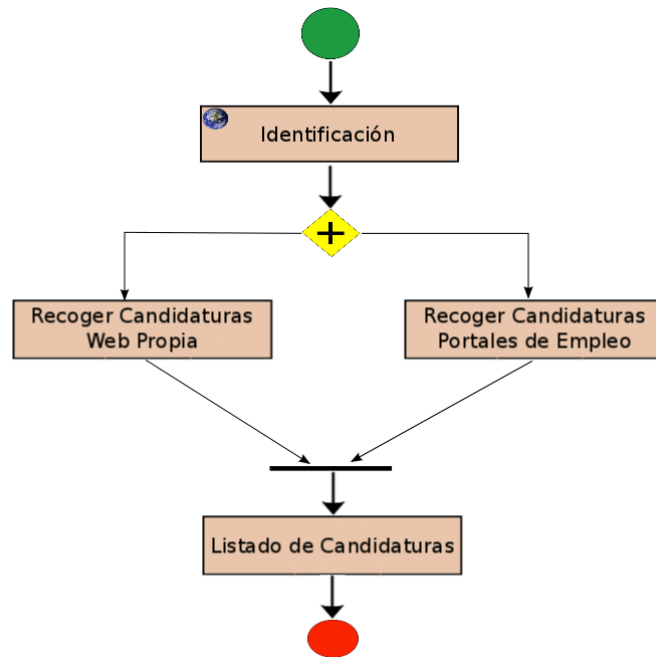


Figura 111: Subproceso de reclutamiento según el nivel 3

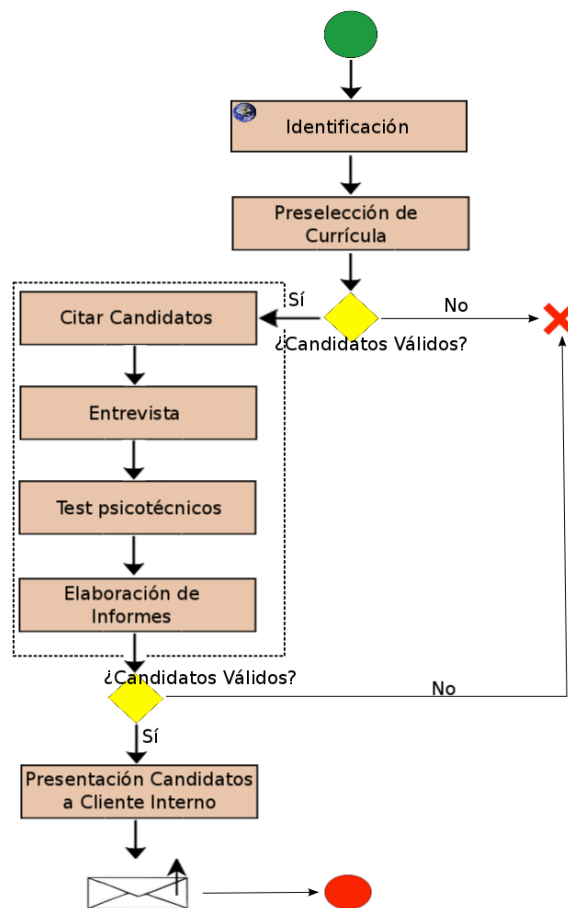


Figura 112: Subproceso de valoración de candidatos por RR.HH según el nivel 3

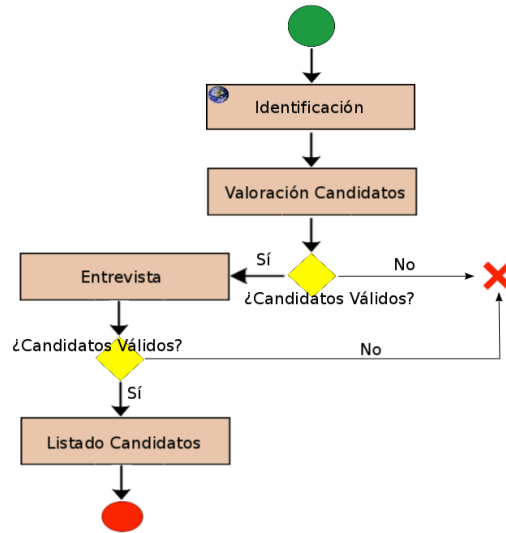


Figura 113: Subproceso de valoración de candidatos por C.I según el nivel 3

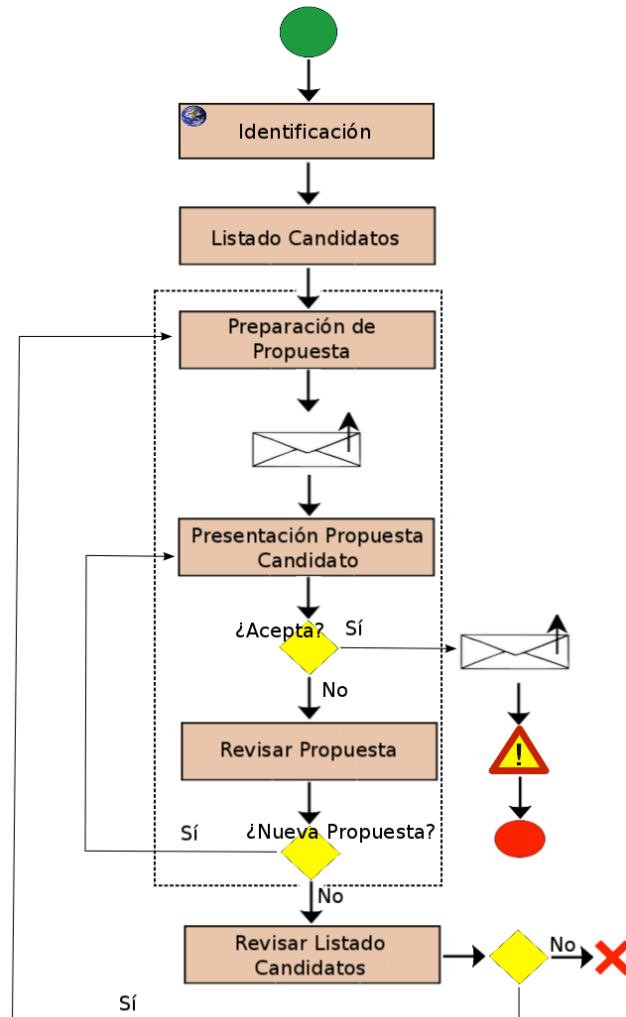


Figura 114: Subproceso de oferta de contratación según el nivel 3

7.3.2.2.5 Modelado del proceso utilizando el nivel 4

Las imágenes mostradas a continuación representan a los procesos del sistema de selección de personal una vez se les ha aplicado el último nivel de la metodología propuesta. A través de este último nivel se han añadido detalles relacionados con el manejo de objetos, colecciones de objetos y actividades de almacenamiento de datos.

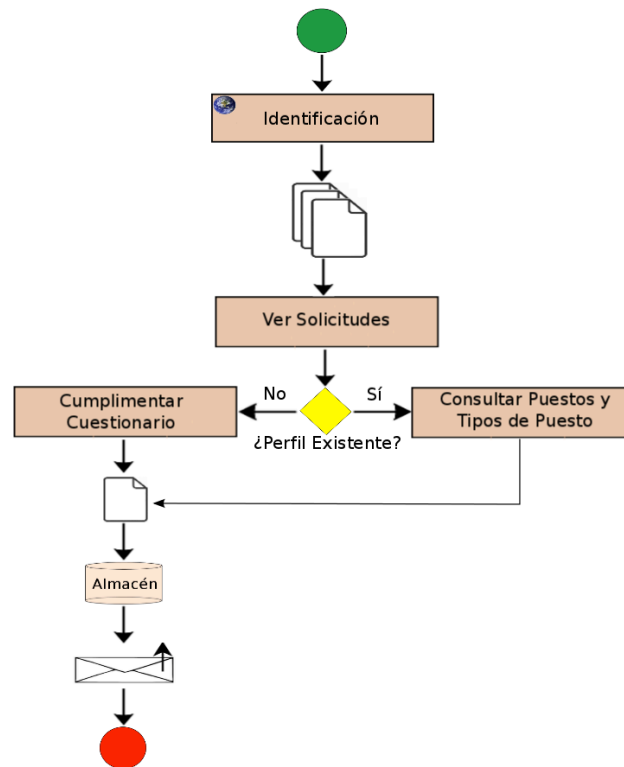


Figura 115: Subproceso de solicitud de nuevo empleado según el nivel 4

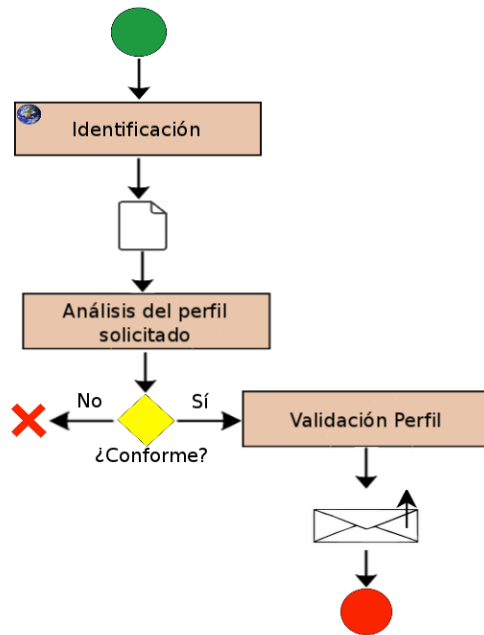


Figura 116: Subproceso de valoración de solicitud según el nivel 4

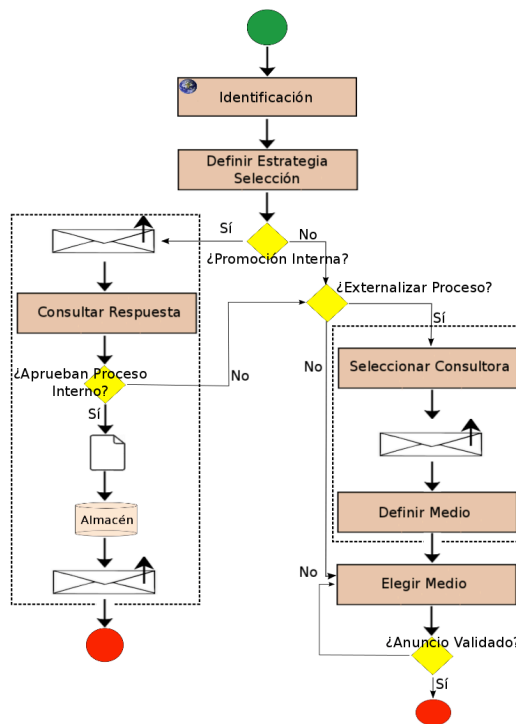


Figura 117: Subproceso de valoración de proceso según el nivel 4



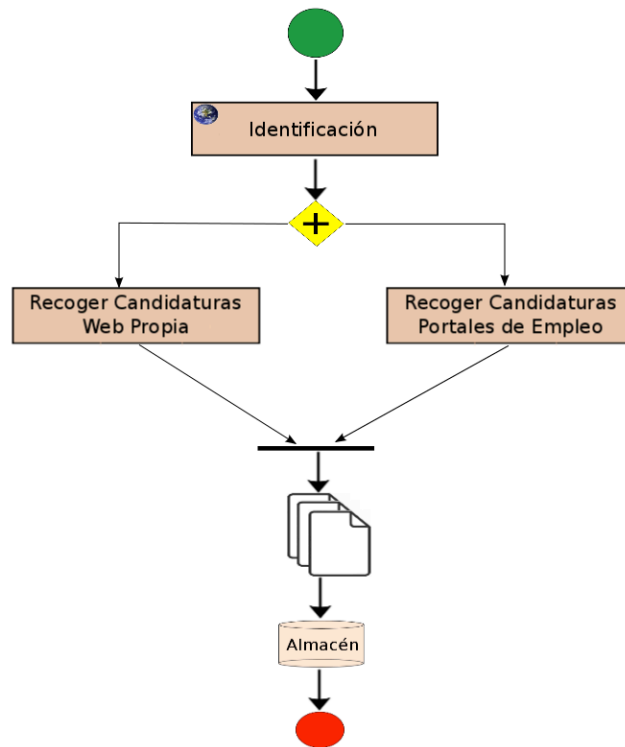


Figura 118: Subproceso de reclutamiento según el nivel 4

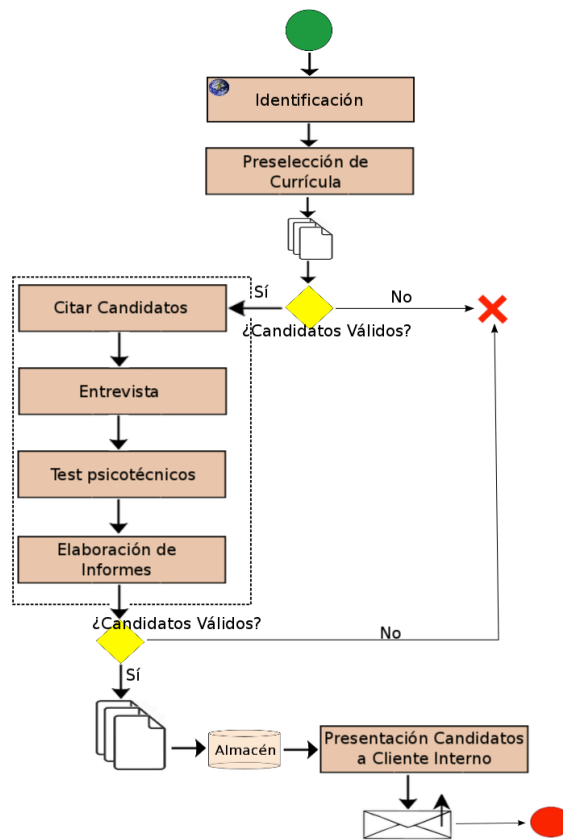


Figura 119: Subproceso de valoración de candidatos por RR.HH según el nivel 4

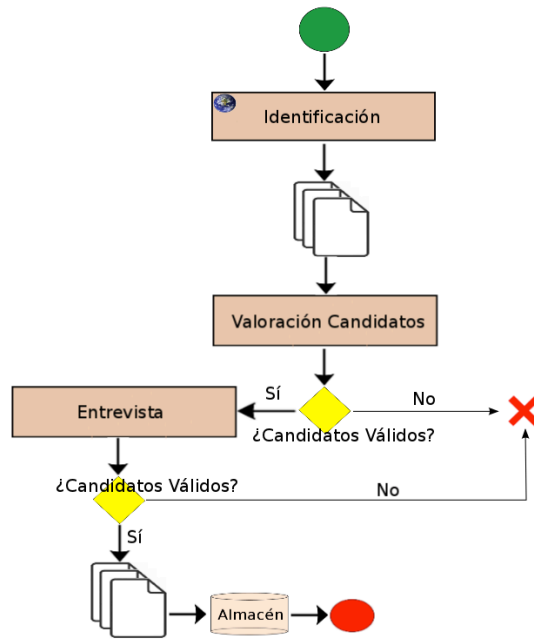


Figura 120: Subproceso de valoración de candidatos por C.I según el nivel 4

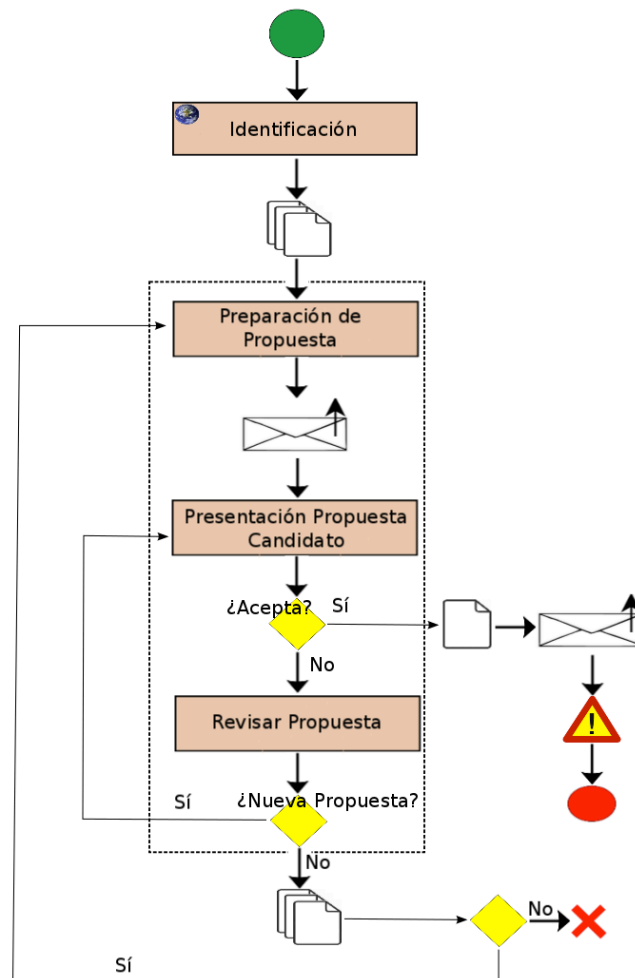


Figura 121: Subproceso de oferta de contratación según el nivel 4

## 7.4 MODELADO DE LA LÓGICA DE UN VIDEOJUEGO PARA DISPOSITIVOS MÓVILES

Una vez visto que la aplicación de la metodología de modelado por niveles puede ser aplicada de manera satisfactoria a procesos de negocio habituales como la gestión de incidencias informáticas o el proceso de contratación de personal, intentaremos llevar la utilización de la metodología un paso más adelante.

En la actualidad, con la aparición de distintos tipos de dispositivos móviles (como smartphones o tablets), la industria de los videojuegos ha sufrido un importante cambio en las condiciones de juego, ya que los videojuegos pueden ejecutarse sobre distintas plataformas y dispositivos que los usuarios llevan encima todo el día.

A este cambio de escenario de uso de los juegos es necesario sumar la aparición de las tiendas de aplicaciones de las que disponen las distintas plataformas y la conectividad a Internet que disponen la mayor parte de los dispositivos, lo que hace posible que los usuarios descarguen los juegos directamente en los dispositivos sin necesidad de esperar a encontrarse frente a un ordenador para poder hacerlo.

Ante esta situación, durante la definición de los distintos elementos de la metodología y su puesta en escena dentro de procesos de negocio como los mostrados anteriormente, se hizo patente la posibilidad de aplicar la misma técnica para definir el “*game loop*” o bucle del juego de los videojuegos a través de la utilización de los elementos disponibles en BPLOM; el bucle del juego es la lógica que representa el funcionamiento del juego a nivel de movimientos y simulación del comportamiento de todos sus elementos. Además, realizando una separación de tipos de videojuegos es posible extraer la parte común de los mismos para generar una plantilla y realizar la transformación adecuada para que el modelo que represente el bucle del juego pueda ser traducido a código e introducido dentro de dicha plantilla de forma sencilla.

### 7.4.1 Familias de videojuegos

El primer paso necesario para la aplicación del modelado de procesos al desarrollo de videojuegos es la división de éstos en familias o tipos para proceder a realizar la abstracción que permita generar la plantilla del videojuego para la posterior transformación del modelo a código y su inclusión dentro de la plantilla.

La división en familias que se ha realizado durante el proceso de adaptación de la metodología por niveles al modelado de videojuegos ha sido la siguiente, si bien no se trata de la única forma de agrupación posible:

- **Estrategia por turnos:** La estrategia por turnos es uno de los tipos de videojuego más típicos. La mecánica de funcionamiento de estos juegos se basa en un grupo de personajes controlado por el usuario y otro grupo controlado por la inteligencia artificial, siendo el objetivo del mismo moverse y atacar al contrario hasta destruirlo. Es posible encontrar juegos de estrategia por turnos en los que existan otros tipos de objetivos como destruir a un personaje contrario determinado (el líder del contrario) o llegar a una determinada zona del mapa y controlarla

durante un número de turnos establecido. El juego más conocido de esta familia es el Risk, donde el usuario maneja un ejército de tropas y tiene que ir conquistando un mapa del mundo con ellas.



Figura 122: Captura de pantalla del juego Risk

- **Juegos touch:** Los juegos touch son aquellos en los que el usuario controla limitadamente un vehículo o un personaje a partir de los toques que este realiza en la pantalla del dispositivo. El escenario de este tipo de juego suele estar en movimiento y además existen una serie de elementos que van apareciendo en el escenario y que han de ser recolectados por el usuario para ganar puntos u otras recompensas, aunque también existen elementos perjudiciales para el usuario (pérdida de puntos, de vidas o de tiempo). En este tipo de juegos normalmente el usuario ha de llegar a un punto establecido del mapa para pasar de nivel o bien recolectar un número de elementos para obtener la puntuación que le permite completar la misión del nivel. Últimamente este género de videojuegos es conocido como *endless runner* y existen varios ejemplos de juegos de este tipo con un gran número de descargas en las tiendas de aplicaciones: Jetpack Joyride, Temple Run y Flappy Bird, entre otros.



Figura 123: Captura de pantalla del juego Jetpack Joyride



Figura 124: Captura de pantalla del juego Temple Run

- **Juegos de plataformas:** Los juegos de plataformas son aquellos en los que el usuario controla a un único personaje que ha de ir moviendo por un escenario con el fin de eliminar a los enemigos y alcanzar la meta del nivel, utilizando las distintas plataformas existentes a lo largo del mapa para alcanzar zonas elevadas. La diferencia de este tipo de juegos con respecto a los juegos touch radica en la capacidad de movimiento del personaje, ya que en el caso de los juegos de plataformas el personaje puede moverse en todas direcciones e incluso saltar contando con los bordes del escenario como única limitación. Uno de los juegos clásicos de este género es el Metal Slug, una saga de juegos de plataformas donde el usuario controla unos soldados con los que tiene que navegar por los mapas al tiempo que destruye a los enemigos y recoge elementos que dan puntos.





Figura 125: Captura de pantalla de un juego de la saga Metal Slug

- **Juegos de puzle:** Los juegos de puzle se basan en la aparición de piezas o figuras en una parte de la pantalla y tienen como objetivo que el usuario encaje dichas piezas de tal forma que un número determinado de piezas iguales sean adyacentes y se destruyan, desapareciendo de la pantalla. El cometido del juego es lograr que la pantalla quede libre de piezas. Se trata de uno de los tipos de juego más clásicos ya que irrumpió con fuerza con el Tetris y a día de hoy sigue vigente con alternativas como Puzzle Bobble, el juego de las burbujas de colores.



Figura 126: Captura de pantalla del juego Puzzle Bobble

- **Juegos de trivial:** Otro de los tipos clásicos de juegos, donde se presenta al usuario una pregunta con un número determinado de respuestas posibles y un espacio de tiempo en el que se ha de responder a la pregunta. En caso de que la respuesta

indicada por el usuario sea la correcta este obtendrá puntos, ganando el usuario que más puntos haya obtenido. Un juego de reciente éxito dentro de esta categoría es Atriviate, anteriormente conocido como Triviados, que representa una partida de trivial en la que podemos competir contra un amigo que tenga instalada la aplicación en su dispositivo.

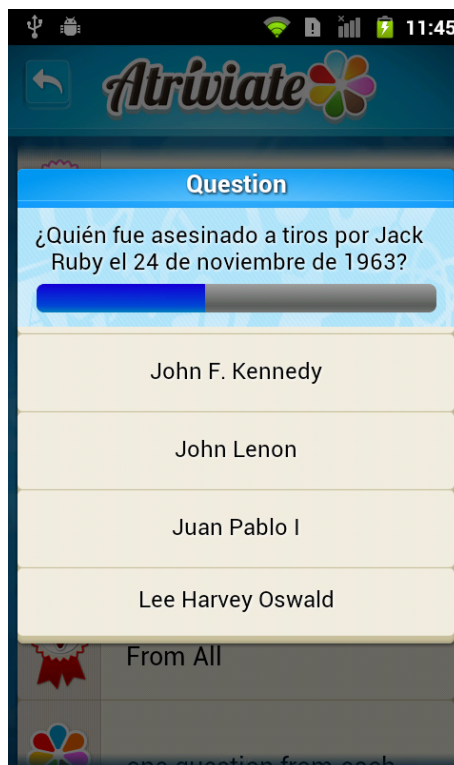


Figura 127: Captura de pantalla de Atriviate

#### 7.4.2 Familias idóneas para la aplicación de la metodología

Una vez realizada esta división de los juegos en los distintos grupos o familias, es necesario establecer cuales de las familias disponen de una estructura lo suficientemente rica y compleja para su representación utilizando la metodología propuesta. Tanto los juegos de tipo trivial como los juegos de tipo puzzle disponen de una lógica demasiado cerrada, que no permite sacar partido de la expresividad de la notación debido a que las modificaciones en la lógica del juego son mínimas. Por esta razón se cree que ambas familias no son del todo adecuadas para su inclusión en este estudio.

El resto de las familias descritas (estrategia por turnos, plataformas y juegos touch) si que se acercan más a las características del modelado de procesos, ya que las distintas acciones y reacciones de los personajes se asemejan a las experimentadas por el usuario de una aplicación informática convencional. Por lo tanto, estas tres familias serán las utilizadas como campo de pruebas para el modelado de videojuegos a partir de BPLOM.

Al igual que ocurriera en el caso de los procesos de negocio mostrados anteriormente, en esta sección se mostrará la evolución de los procesos según van pasando por los distintos niveles de la metodología. De esta forma se consigue establecer la evolución que se

observará en los procesos según el incremento en los conocimientos de modelado que vaya experimentando el experto.

#### 7.4.2.1 Aplicación de la metodología a los juegos de estrategia por turnos

La primera familia contemplada es la de los juegos de estrategia por turnos. Como ya se ha comentado anteriormente este tipo de juegos se basan en la destrucción de un conjunto de enemigos a través del control de un grupo de personajes que pueden moverse y atacar. En función de la situación de los personajes en el mapa el usuario puede decidir entre utilizar todas, alguna o ninguna de sus tropas para atacar y/o mover durante un turno.

##### 7.4.2.1.1 Modelado del proceso utilizando el nivel 0

A continuación se puede observar la representación del bucle del juego para la familia de estrategia por turnos según el nivel 0 de la metodología propuesta. La primera actividad se corresponde con la carga de todos los datos del juego, desde los personajes controlados por el usuario hasta los personajes controlados por la máquina y los elementos que componen el escenario.

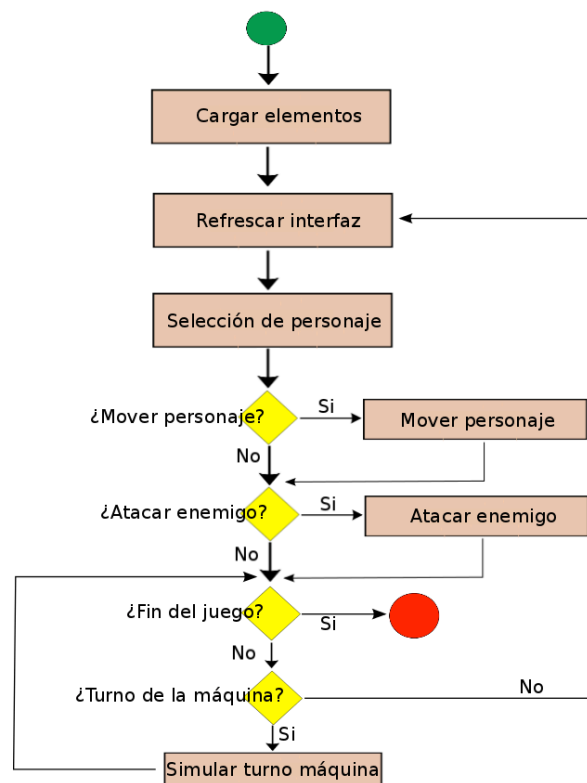


Figura 128: Juego de estrategia por turnos según el nivel 0

Una vez cargados todos los componentes necesarios para el desarrollo del juego, se refresca el interfaz y después llega la primera acción del jugador: la selección del personaje con el que el usuario desea interactuar. Una vez seleccionado, el usuario podrá mover el personaje o atacar a un enemigo con él. Esta secuencia de acciones se



repetirá hasta que llegue el turno de la máquina, delimitado por la imposibilidad de atacar o moverse de los personajes del usuario.

Una vez alcanzado el turno de la máquina, la inteligencia artificial del juego simulará las interacciones y se procederá a comprobar si se ha alcanzado el final del nivel. En caso afirmativo se termina la ejecución del proceso, en caso contrario se vuelve al refresco de la interfaz y se sigue con el juego.

#### 7.4.2.1.2 Modelado del proceso utilizando el nivel 1

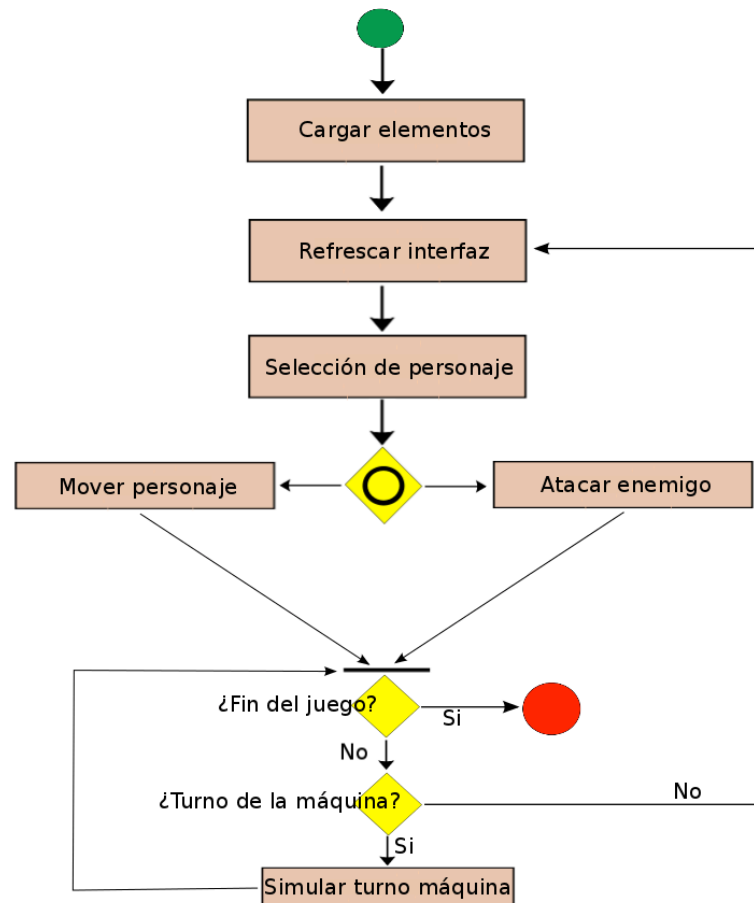


Figura 129: Juego de estrategia por turnos según el nivel 1

La imagen anterior se corresponde con la evolución del proceso del juego de estrategia por turnos después de aplicarle las entidades propias del nivel 1 de la metodología. Tal y como se observa en la imagen, la única modificación del modelo de proceso se corresponde con la utilización de una decisión inclusiva para representar la posibilidad de que el usuario seleccione a uno de sus personajes y pueda moverse, atacar o ambas.

### 7.4.2.1.3 Modelado del proceso utilizando el nivel 2

El uso del nivel denominado como extensión de eventos provoca la aparición en el proceso de dos nuevas entidades: un evento de señal que avisa al enemigo en cuestión de que ha recibido un ataque por parte del usuario y un evento de mensaje saliente que se desencadena a la finalización del juego.

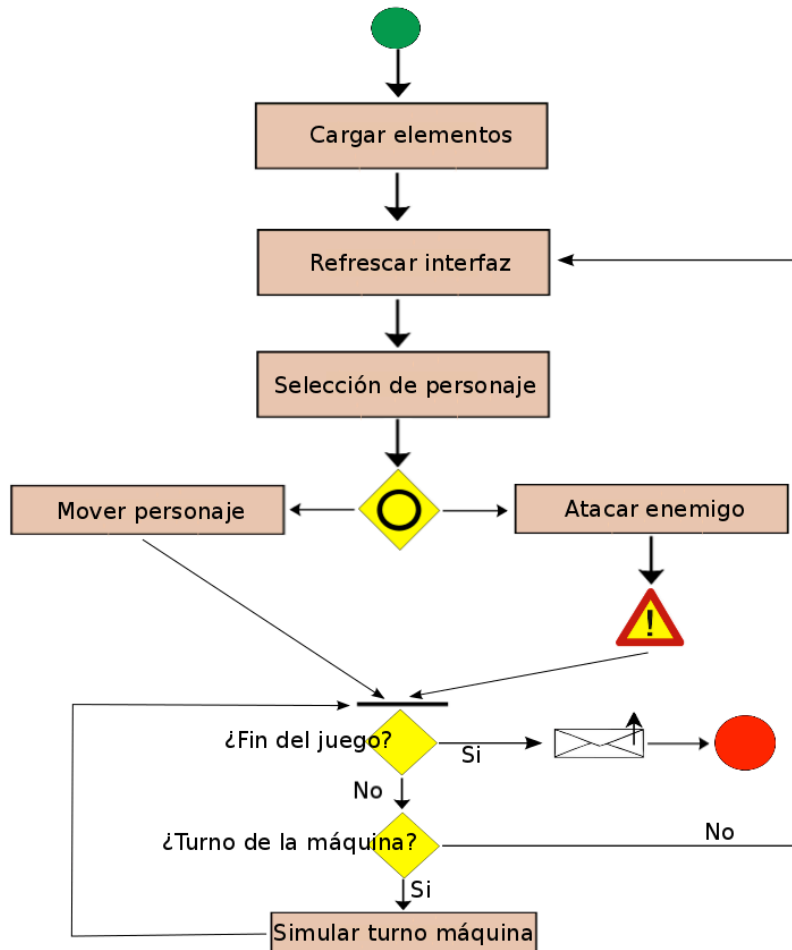


Figura 130: Juego de estrategia por turnos según el nivel 2

### 7.4.2.1.4 Modelado del proceso utilizando el nivel 3

El nivel 3 de la metodología, que propone la utilización de subprocesos y actividades de llamada, se utiliza en el proceso de los juegos de estrategia por turnos a través de las actividades de llamada.

La siguiente ilustración muestra el uso de una actividad de llamada en tres momentos del proceso: la carga de los elementos del juego, el refresco del interfaz gráfico y la simulación del turno de la máquina. Estas actividades pueden repetirse en distintos tipos de juegos modelados a través de la metodología, tal y como se observará más adelante; por lo tanto, se utiliza la entidad de la actividad global para representar esta circunstancia.

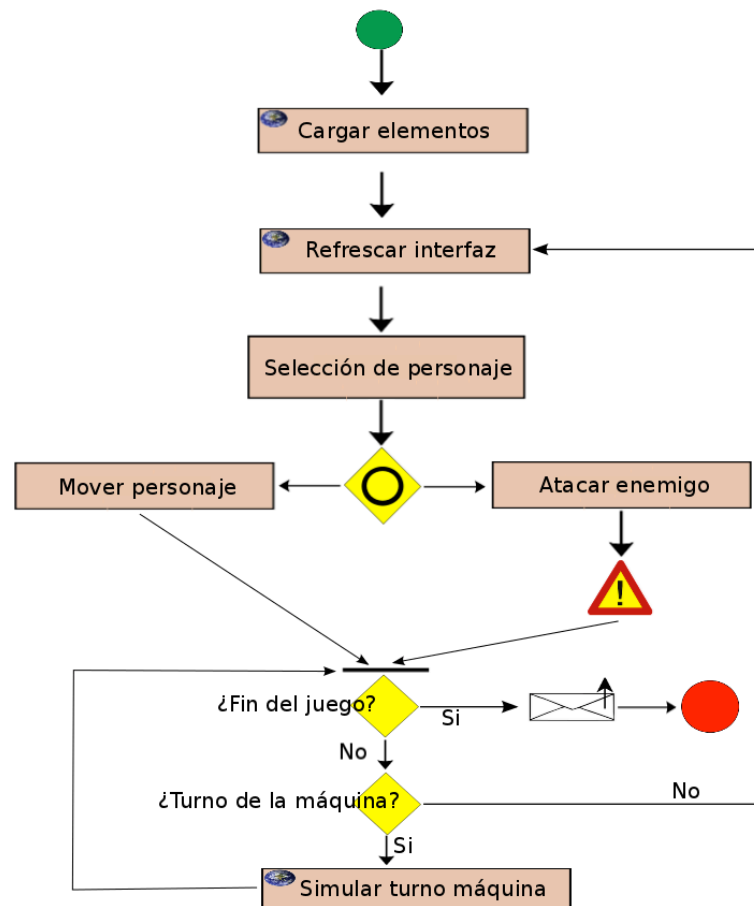


Figura 131: Juego de estrategia por turnos según el nivel 3

La actividad denominada como selección de personaje se mantiene como una actividad normal ya que ninguna de las otras familias de juegos dispone de un conjunto de personajes controlados por el usuario. Por lo tanto, esta actividad solamente se utiliza en este tipo de juegos y no se marca como una actividad de llamada.

#### 7.4.2.1.5 Modelado del proceso utilizando el nivel 4

La imagen que se incluye a continuación muestra el proceso de negocio que representa la lógica de un juego de estrategia por turnos según el nivel 4 de la metodología propuesta. El único cambio con respecto a los modelos anteriores son las tres primeras entidades que aparecen en el modelo. Estas se corresponden con la carga del conjunto de personajes, el conjunto de enemigos y el mapa que aparecen en la pantalla del juego a través de las entidades de datos propias de este nivel. De esta forma se omite la necesidad de crear actividades específicas para la carga de los datos del juego.

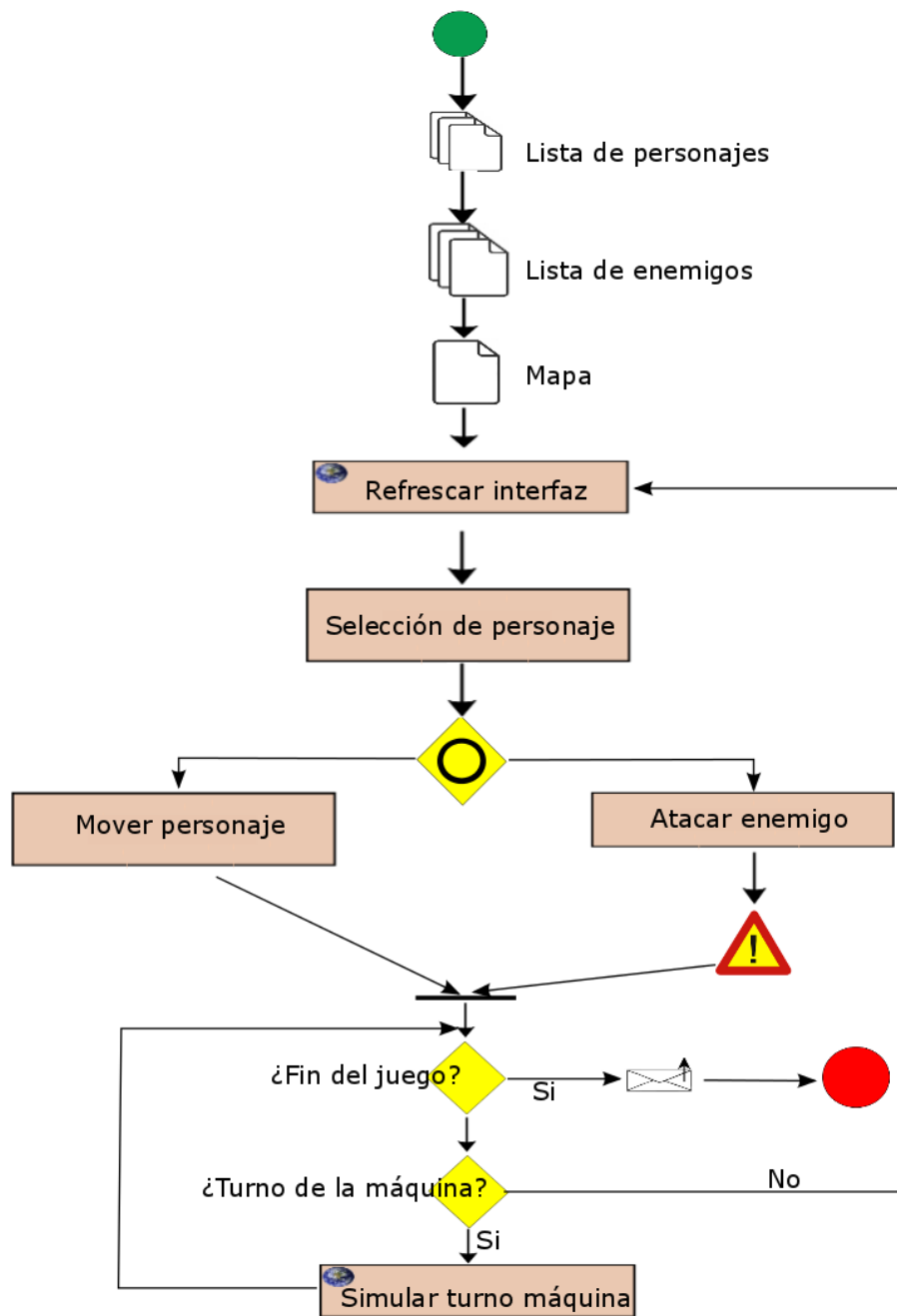


Figura 132: Juego de estrategia por turnos según el nivel 4

#### 7.4.2.2 Aplicación de BPLOM a juegos de la familia touch

En segundo lugar se ha decidido aplicar la metodología de modelado de procesos de negocio por niveles a los juegos de tipo touch.

En estos juegos el usuario controla a un único personaje que puede mover por la pantalla de forma limitada con el fin de coger o recolectar elementos que le aporten puntuación u otras recompensas del juego. También puede ser necesario en este tipo de juegos esquivar elementos que tengan un efecto dañino en el personaje controlado por el usuario.

Una vez realizado el movimiento o la recolección del elemento, se procede a mover los elementos del juego y refrescar el interfaz gráfico. En último lugar se comprobará si se ha llegado al final del juego, terminándose la ejecución del proceso en caso afirmativo.

**7.4.2.2.1 Modelado del proceso utilizando el nivel 0**

A continuación se puede observar la representación del bucle del juego para la familia de juegos de tipo touch según el nivel 0 de la metodología propuesta.

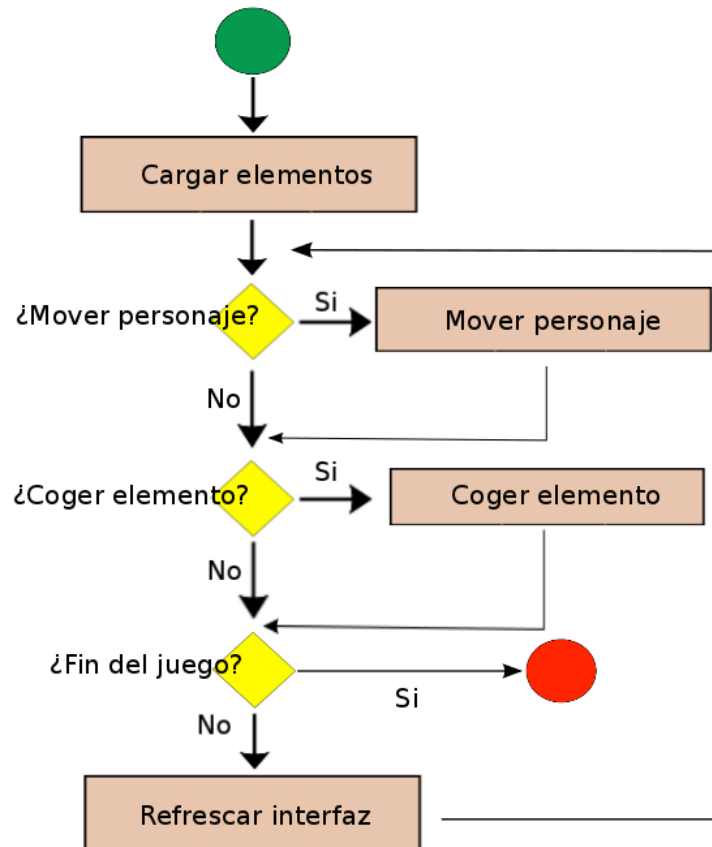


Figura 133: Juego touch según el nivel 0

Al igual que ocurriera en la familia anterior, la primera actividad de un juego de tipo touch es la carga de todos los componentes necesarios para el desarrollo del juego. Una vez que se completa esa carga de recursos empieza la lógica del juego donde el usuario puede mover al personaje y recoger los elementos que se encuentran diseminados por el nivel.

Una vez analizadas las acciones referentes al personaje del juego es necesario comprobar si el usuario ha llegado al final del nivel. Si el usuario ha llegado al final del nivel se procede a terminar el proceso. En caso contrario, será necesario refrescar el interfaz gráfico del juego; esta acción conlleva el movimiento de todos los elementos del juego y la actualización de las animaciones que los representan.

**7.4.2.2.2 Modelado del proceso utilizando el nivel 1**

La extensión de decisiones de BPLOM permite una serie de cambios notables en los juegos, tal y como se mostró con el modelo de proceso de nivel 1 del juego de estrategia por turnos. Al igual que en aquel caso, el modelo de proceso para el juego de tipo touch

incorpora una decisión inclusiva en este nivel para simplificar la toma de decisión sobre movimiento del personaje o recolección de elemento.

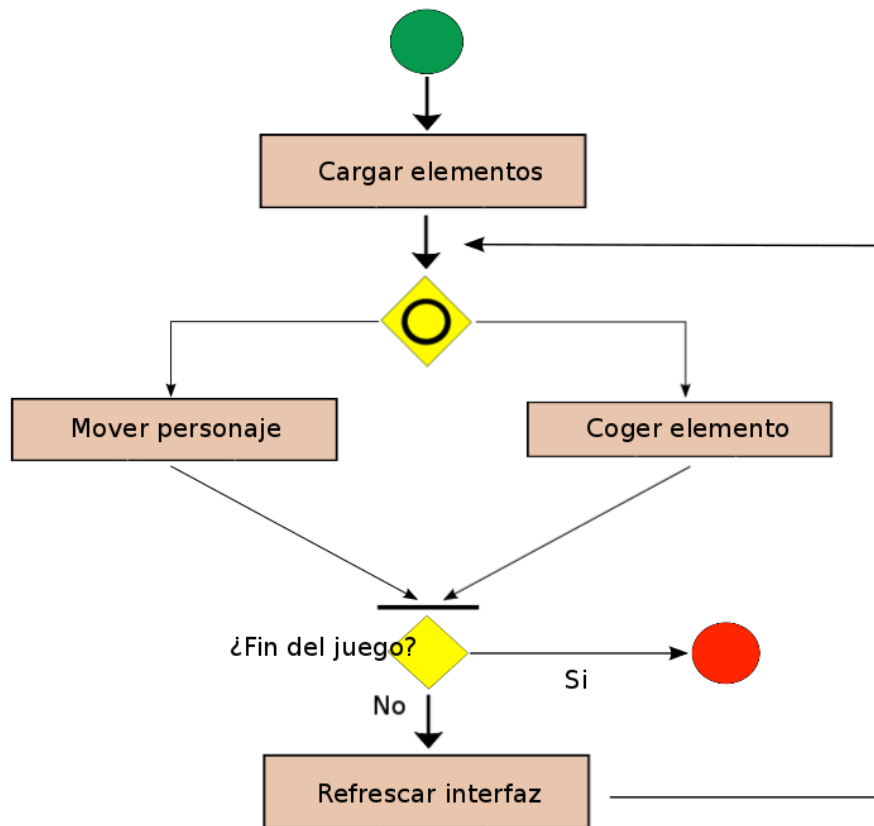


Figura 134: Juego touch según el nivel 1

#### 7.4.2.2.3 Modelado del proceso utilizando el nivel 2

El nivel 2 de BPLOM, que ofrece la posibilidad de utilizar eventos en la definición de los procesos, contribuye en el modelo de los juegos de touch con 3 eventos distintos. En primer lugar, el personaje recibe un evento de mensaje entrante cuando va a interactuar con un elemento del escenario, de tal forma que éste sepa que ha de ejecutar la recolección de este elemento.

La siguiente aparición de un evento también está relacionada con la recolección de los elementos del nivel pero en este caso se corresponde con el evento de señal que se le envía al elemento que ha sido recogido por el usuario. Este evento le indica al elemento que ha sido recogido y pasa a estar inactivo.

En último lugar, cuando se produce el fin del juego, se envía un mensaje saliente al usuario para indicarle que ha alcanzando el final del nivel.

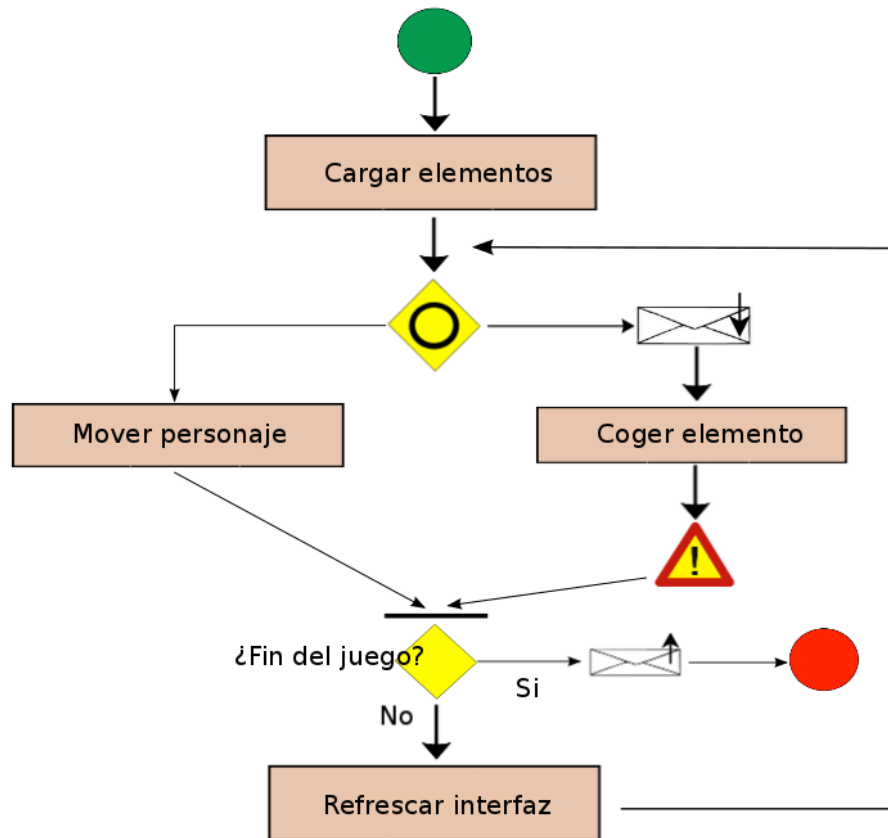


Figura 135: Juego touch según el nivel 2

#### 7.4.2.2.4 Modelado del proceso utilizando el nivel 3

La extensión de actividades de BPLOM tiene efecto sobre los juegos de tipo touch a través de la utilización de las actividades de llamada.

Este tipo de actividad, diseñada para favorecer la reutilización de actividades a través de varios procesos de negocio cuando es necesario repetir comportamientos en distintos procesos, aparece en dos lugares del proceso: en la parte inicial donde se produce la carga de los elementos que componen el juego y en la parte final del proceso a la hora de proceder al refresco del interfaz gráfico.

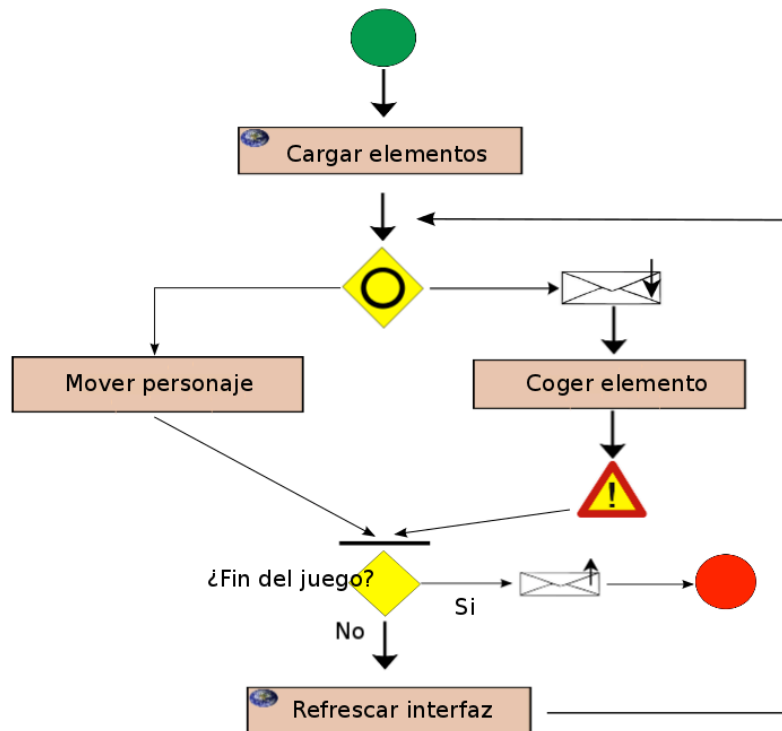


Figura 136: Juego touch según el nivel 3

#### 7.4.2.2.5 Modelado del proceso utilizando el nivel 4

La figura que se incluye a continuación establece el modelo de proceso de un juego de tipo touch según las entidades existentes en el último nivel de la metodología.

La diferencia que se puede observar con respecto a los juegos de estrategia por turnos es la utilización de un objeto de datos en lugar de una colección para albergar a los personajes controlados por el usuario; esto se debe a que en este tipo de juegos se tiene la posibilidad de controlar solamente a un personaje.

Al igual que en los juegos de estrategia, el segundo elemento de la extensión de datos que se utiliza en el proceso se corresponde con una colección de objetos destinada a la carga del resto de elementos del juego. A partir de ese momento no se incluye ningún otro componente del nivel 4 de la metodología, por lo que el resto del proceso se mantiene con respecto al del nivel 3.



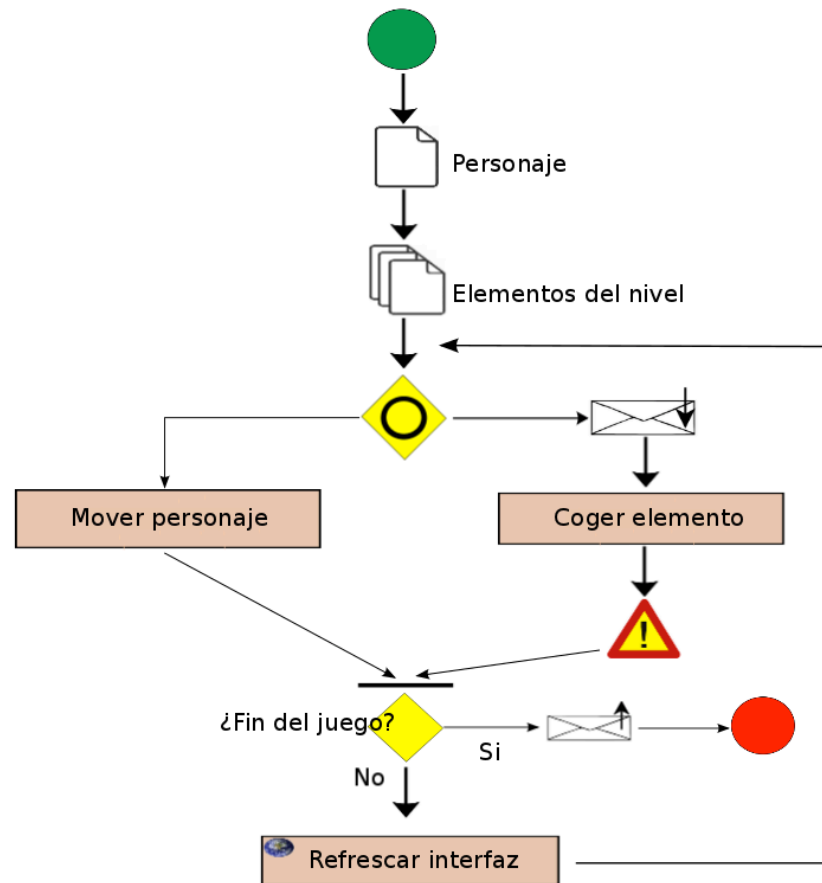


Figura 137: Juego touch según el nivel 4

### 7.4.2.3 Aplicación de la metodología a juegos de plataformas

En último lugar se decidió aplicar la metodología de modelado propuesta en este documento para representar a los juego encuadrados dentro de la familia plataformas.

En este tipo de juegos el usuario controla a un personaje que puede moverse libremente por un escenario determinado, eliminando los enemigos que salgan a su paso y recogiendo los elementos dispuestos a lo largo del escenario para tal fin, con el objetivo de llegar a una posición que ha sido definida como el final o la meta del nivel.

Una vez que el usuario realiza la interacción con el nivel se procede a simular el movimiento y los disparos de los enemigos, siempre y cuando se cumplan las condiciones que lo permiten.

Concluida la simulación de la inteligencia artificial se procede a comprobar si se ha alcanzado el final del nivel. En el caso de los juegos de plataformas el final puede llegar por tres condiciones: se consume el tiempo máximo para pasarse el nivel, el jugador se queda sin puntos de vida o el jugador ha alcanzado la meta del nivel. En caso de que se valide alguna de estas condiciones se procede a terminar el nivel mientras que en caso contrario se procederá a refrescar los gráficos del juego y continuar con su ejecución.

### 7.4.2.3.1 Modelado del proceso utilizando el nivel 0

A continuación se incluye la representación gráfica del modelo de proceso que representa a los juegos de plataformas. De nuevo, y al igual que ocurriera en los otros procesos, se comienza por cargar los elementos que participan en el juego para después comenzar con el bucle del juego en si.

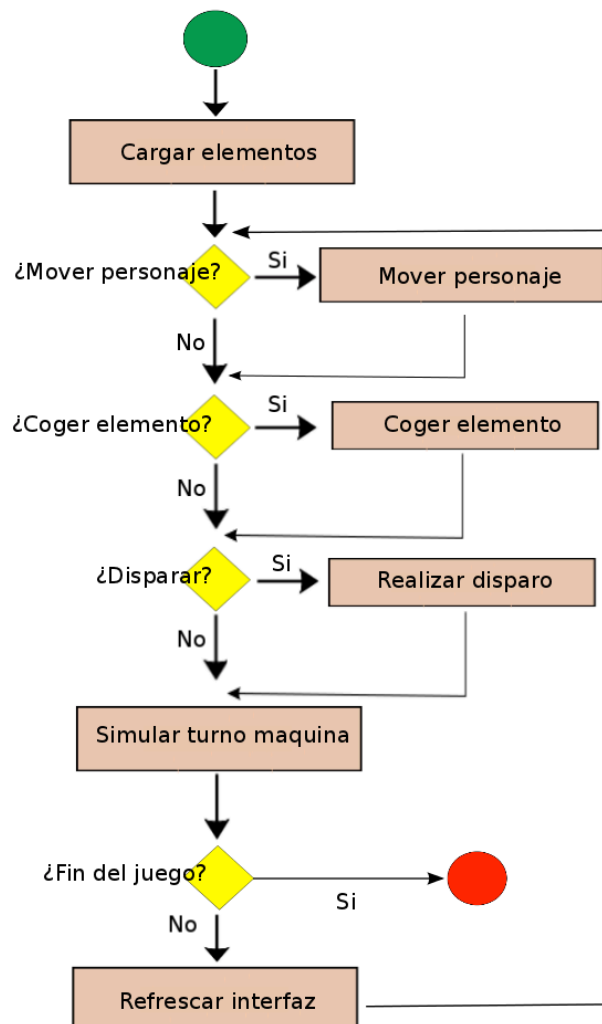


Figura 138: Juego de plataformas según el nivel 0

Es necesario hacer notar la estructura de ejecución necesaria para este tipo de juego, encadenando los resultados de las decisiones de tal forma que se produzca la interacción con los elementos del juego de una manera que parezca que se ejecuta todo en paralelo. Esta misma estructura pudo observarse en los modelos del resto de juegos para el nivel 0.

### 7.4.2.3.2 Modelado del proceso utilizando el nivel 1

A diferencia de los modelos mostrados anteriormente, el juego de plataformas contiene una estructura de ejecución en paralelo, que no había sido vista hasta el momento.

La inclusión de esta estructura, visible en la parte central del modelo, permite que el personaje del juego realice dos acciones al mismo tiempo: la recolección de los objetos existentes en el nivel y la realización de disparos contra los enemigos. Esta ejecución en

paralelo en combinación con la decisión inclusiva le confieren al juego la capacidad de mover al personaje y disparar y recoger elementos al mismo tiempo.

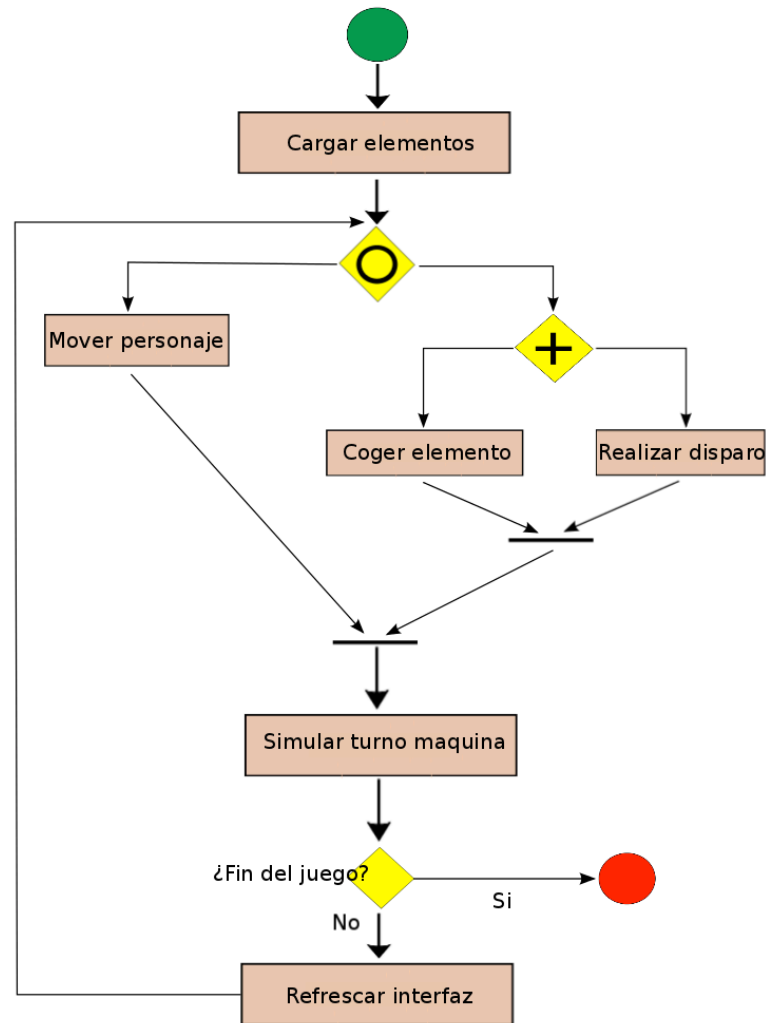


Figura 139: Juego de plataformas según el nivel 1

#### 7.4.2.3.3 Modelado del proceso utilizando el nivel 2

La extensión de eventos tiene una aplicación mayor en este juego de plataformas que en los otros juegos que han sido modelados hasta el momento, ya que dada la naturaleza del juego es necesario enviar mayor número de avisos entre los componentes.

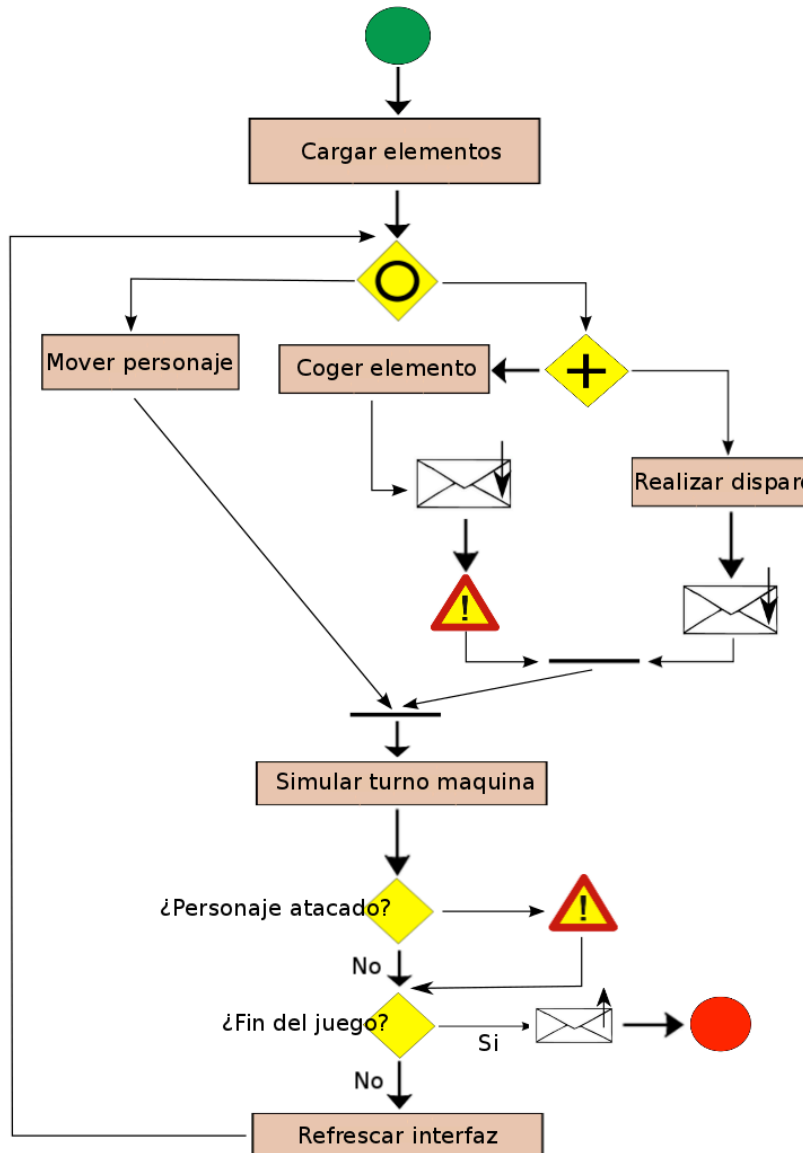


Figura 140: Juego de plataformas según el nivel 2

En primer lugar están los avisos que se efectúan al tiempo que el personaje captura los elementos y realiza sus disparos. En el caso de los elementos, el personaje recibe un mensaje entrante que el indica que va a recoger un objeto y que debe efectuar la animación asociada mientras que el elemento en si recibe una señal que le indica que ha sido capturado. En la otra rama de la ejecución en paralelo, la encargada de la realización de los disparos, el personaje también recibe un mensaje entrante en el que se le indica que ha disparado y que debe reaccionar ante esta circunstancia.

Tras la simulación del turno de la maquina, el personaje puede haber sido objeto de ataque por parte de los enemigos del juego por lo que será necesario incluir una nueva decisión donde se debe averiguar si el personaje ha sido objeto de ataque alguno. En caso afirmativo, el personaje recibirá una señal que le indicará que ha sido alcanzado por el ataque. Por último, y al igual que ocurre en el resto de procesos, a la finalización del nivel se crea un mensaje saliente que avisa al usuario de que ha concluido el nivel.

7.4.2.3.4 Modelado del proceso utilizando el nivel 3

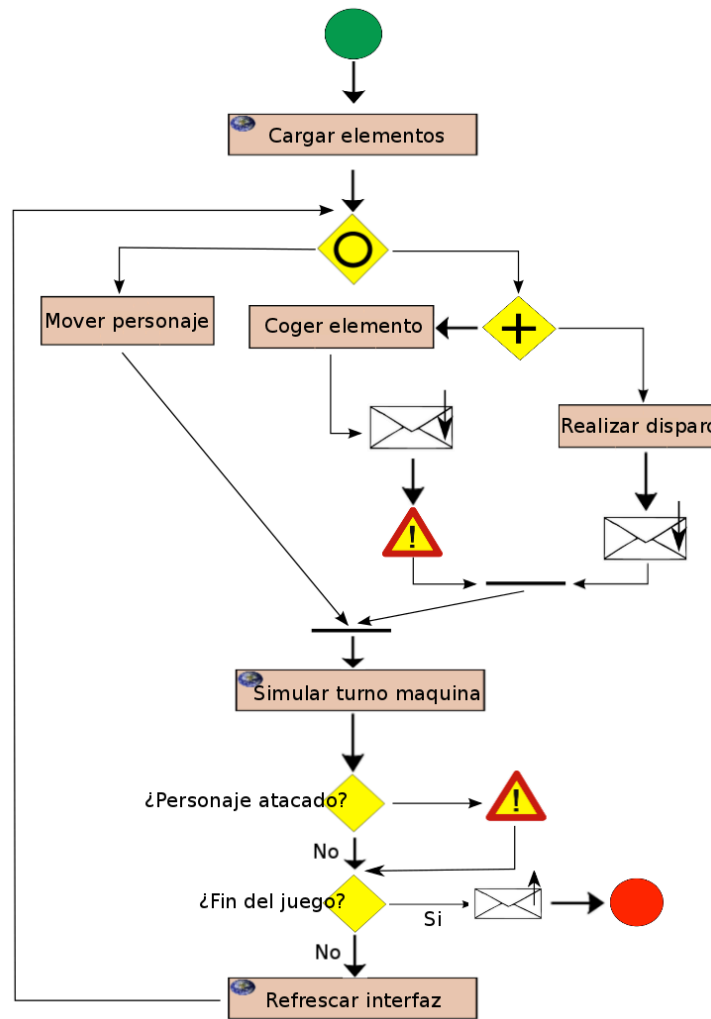


Figura 141: Juego de plataformas según el nivel 3

La aplicación del nivel 3 de la metodología BPLOM en el modelo de proceso para el juego de plataformas tiene el mismo efecto que en los casos anteriores: se sustituyen las actividades comunes a otros procesos con actividades de llamada. En este caso, las actividades sustituidas por actividades de llamada son la carga de los elementos, el refresco del interfaz y la simulación del turno de la máquina.

7.4.2.3.5 Modelado del proceso utilizando el nivel 4

El modelo de proceso que se muestra a continuación describe el funcionamiento de un juego encuadrado dentro de la familia de plataformas según el nivel 4 de la metodología.

Al igual que en los modelos de proceso anteriores será necesario realizar la carga de todos los elementos participantes antes de que se pueda proceder a ejecutar el bucle del juego. En este caso se cargarán los siguientes elementos: el personaje principal, los enemigos, los objetos que pueden ser recogidos por el personaje y el nivel o mapa del juego, compuesto por un conjunto de casillas.

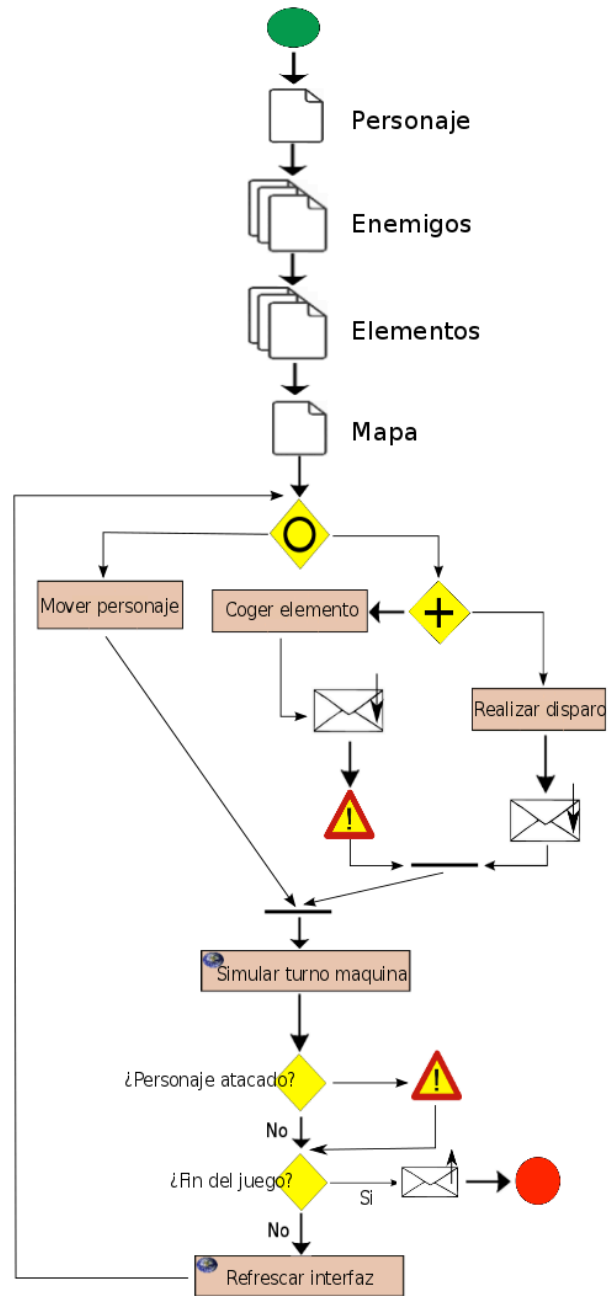


Figura 142: Juego de plataformas según el nivel 4

## 7.5 MODELADO DE UNA APLICACIÓN DE CATÁLOGO PARA DISPOSITIVOS MÓVILES

El último apartado de este caso de estudio se corresponde con la utilización de la metodología BPLOM para acometer el modelado de una aplicación de catálogo para dispositivos móviles. Tal y como se explicó en el apartado sobre tipos de aplicaciones en las que aplicar la metodología, se considera que la unión de los smartphones con el comercio electrónico puede dar lugar a un nicho de mercado interesante. Ante la existencia de diversas alternativas comerciales que ofrecen la posibilidad de construir aplicaciones de catálogo para dispositivos móviles de una forma híbrida, donde no se aprovechan las capacidades de los dispositivos ni se permite la realización de un pedido, se ha habilitado un apartado en el caso de estudio para demostrar la capacidad de adaptación de la metodología a este campo.

### 7.5.1 Descripción del proceso

El proceso de negocio que está detrás de una aplicación de catálogos es el siguiente. Al iniciarse la aplicación se produce la carga de todos los catálogos que se hayan definido, teniendo cada uno de ellos un número de productos asociado. Cuando el usuario de la aplicación selecciona un catálogo para visualizar su contenido, la aplicación realiza la carga de los detalles de los productos asociados a dicho catálogo.

Una vez que los detalles de los productos han sido cargados el usuario puede navegar a través de ellos de forma sencilla para observar sus características y una galería de imágenes del producto; en caso de que el producto le interese dispondrá de un botón que le permitirá añadir el producto al carro de la compra.

Cuando el usuario haya terminado de navegar por los catálogos incluidos en la aplicación y haya finalizado el montaje de su pedido, el usuario tendrá la oportunidad de realizar el pedido desde la propia aplicación según los parámetros de configuración con los que haya sido construida.

### 7.5.2 Aplicación de la metodología por niveles al proceso

A continuación se incluye el desglose de aplicación de la metodología de modelado por niveles al proceso de negocio para aplicaciones de catálogos.

#### 7.5.2.1 Modelado del proceso utilizando el nivel 0

La imagen que se incluye a continuación muestra cómo sería la aplicación del nivel inicial de la metodología de modelado por niveles al proceso de negocio que describe una aplicación de catálogos. Tal y como se puede observar la primera actividad del proceso es la carga de los distintos catálogos que se han definido, permitiéndose entonces la navegación del usuario por éstos y la selección de uno de ellos para consultar los detalles de los productos incluidos en el mismo.

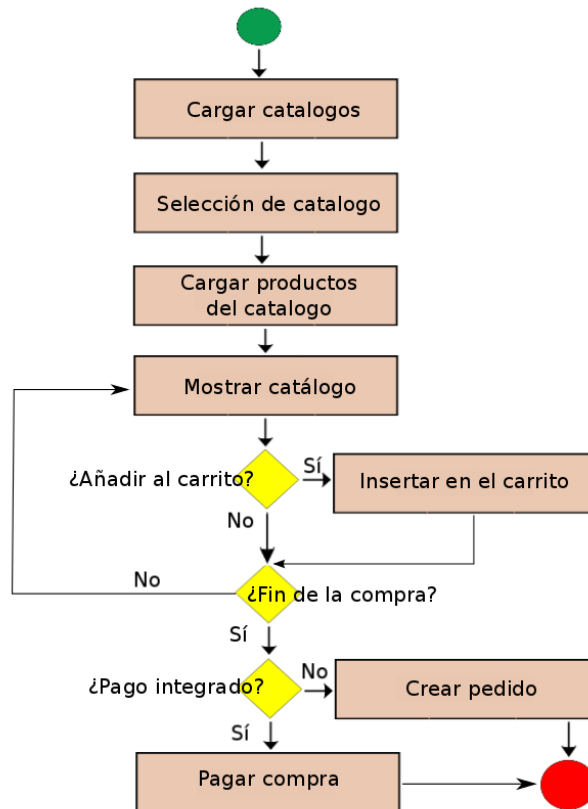


Figura 143: Proceso de negocio de aplicación de catálogos según el nivel 0

Tras la carga de los productos que se encuentran en el catálogo, el usuario puede visualizar los detalles de cada producto en la pantalla de su dispositivo y utilizar el control pertinente para añadir el producto que está visualizando a su carro de la compra. Cuando el usuario haya terminado con la selección de los productos de su pedido, la aplicación le permite realizar un pedido: si está habilitado el pago de la compra desde la aplicación, el usuario será dirigido a la pantalla correspondiente para la realización de la transacción económica mientras que si la aplicación no soporta esta operación se creará un pedido que será enviado al propietario del catálogo de forma electrónica y el resto de la transacción se realizará fuera de la aplicación.

### 7.5.2.2 Modelado del proceso utilizando el nivel 1

La extensión de decisiones de la metodología de modelado que se propone en este trabajo de investigación no supone la introducción de cambios en el proceso de negocio asociado a la aplicación. Por lo tanto, el proceso de negocio resultante de la aplicación de este nivel sería idéntico al del nivel 0.



### 7.5.2.3 Modelado del proceso utilizando el nivel 2

La aplicación de la extensión de eventos supone la modificación del proceso de negocio en dos puntos del mismo para incluir dos eventos de mensaje saliente. El primer evento de mensaje incluido en el proceso hace que, tras la inserción de un producto en el carrito de la compra, el usuario reciba por pantalla un mensaje que le confirma la inclusión del producto en su lista de compra. El segundo evento de mensaje saliente es el que se genera tras la realización del pedido; sin importancia del tipo de pedido que se genere (pago desde la aplicación o creación de un pedido sin pago), el usuario recibirá un mensaje de confirmación del pedido que le indicará que su pedido ha sido realizado con éxito.

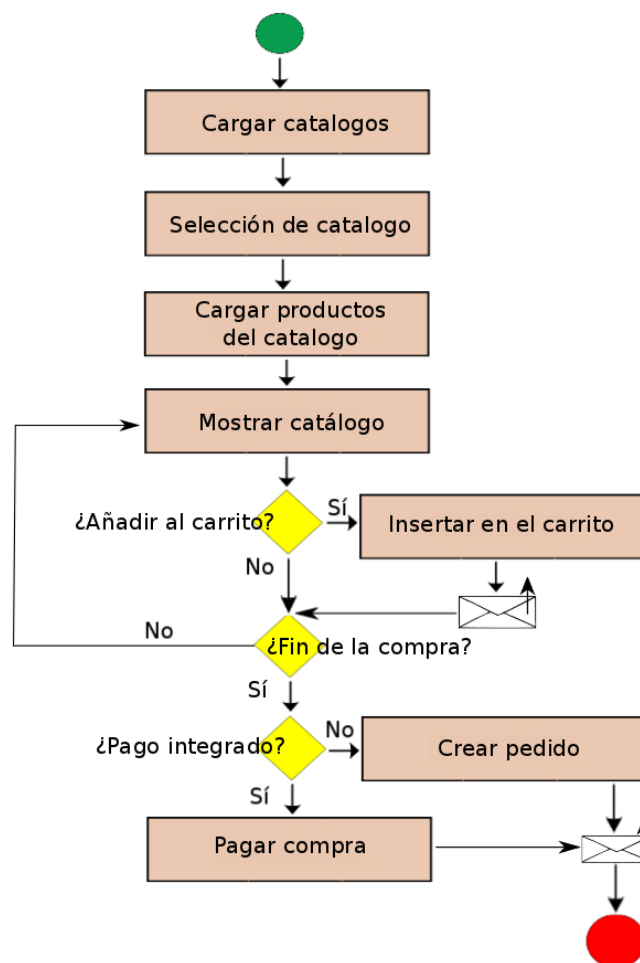


Figura 144: Proceso de negocio de aplicación de catálogos según el nivel 2

### 7.5.2.4 Modelado del proceso utilizando el nivel 3

La aplicación de la extensión de actividades de la metodología de modelado por niveles supone la inclusión de un nuevo tipo de elemento en el modelo de proceso de la aplicación de catálogos: la actividad de llamada.

Este tipo de actividad sirve para representar aquellas actividades comunes que pueden ser reutilizadas en diversos procesos de negocio, evitando la necesidad de configurar diversas actividades para la realización de la misma tarea en distintos dominios. En este caso, se introduce este tipo de construcción en los siguientes lugares: la carga de los catálogos incluidos en la aplicación, la carga de los productos incluidos en el catálogo seleccionado y la inserción de un producto en el carrito. El proceso de negocio resultante de la aplicación de este nivel de la metodología es el mostrado en la siguiente imagen.

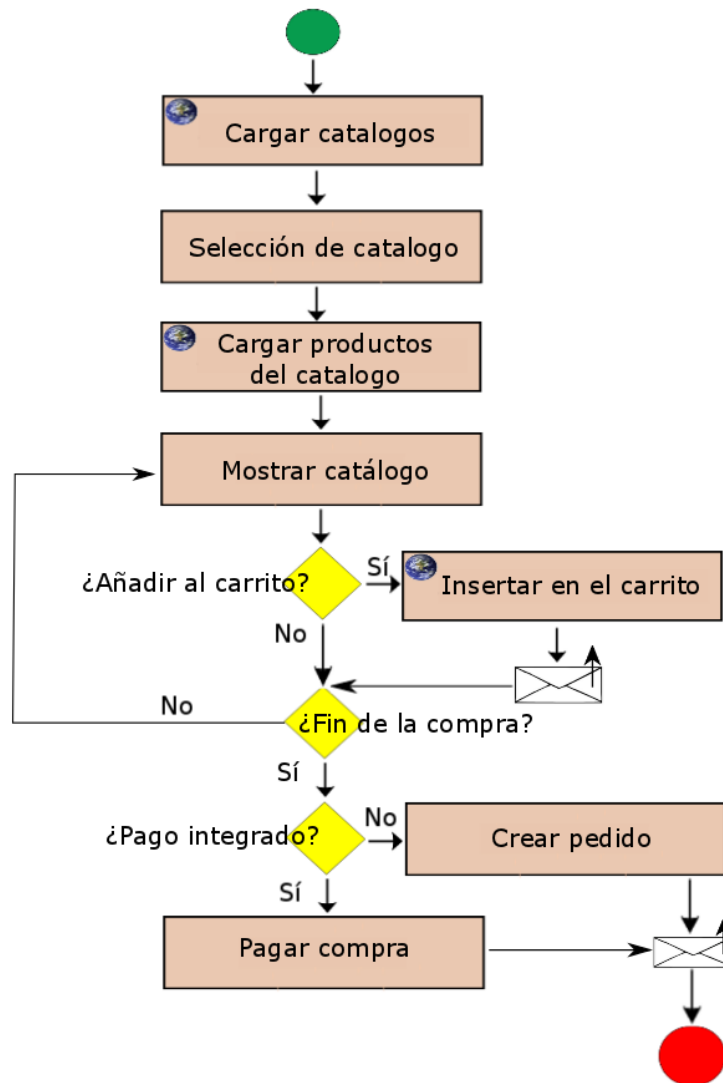


Figura 145: Proceso de negocio de aplicación de catálogos según el nivel 3

#### 7.5.2.5 Modelado del proceso utilizando el nivel 4

La aplicación del último nivel de la metodología, la extensión de datos, permite configurar la carga de los catálogos y de los productos a partir de la entidad denominada colección de objetos. Este tipo de entidad, destinada a aquellos expertos del dominio que dispongan de ciertos conocimiento informáticos, permite establecer el tipo de colección de elementos

que se quiere utilizar para el almacenaje en memoria de los datos que estén siendo utilizados en la aplicación.

A diferencia de otros modelos de proceso mostrados en este caso de estudio, no es necesario utilizar la entidad objeto ni la entidad almacén. Esto es debido a que una aplicación de este tipo siempre trabaja con colecciones de elementos (maneja diversos catálogos y varios productos por catálogo) y no necesita almacenar información a disco porque el pedido no se almacena en el dispositivo sino que se genera y se envía de forma telemática.

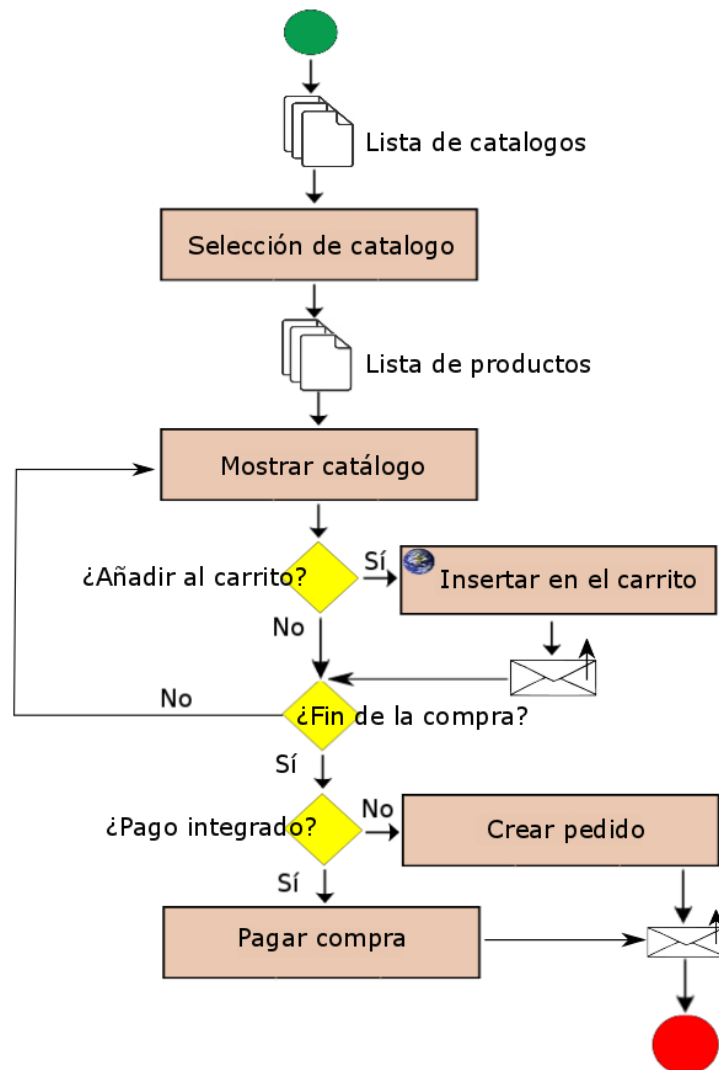


Figura 146: Proceso de negocio de aplicación de catálogos según el nivel 4

## **7.6 CONCLUSIONES**

Este capítulo muestra, a partir de diversos ejemplos, la aplicación de la metodología de modelado por niveles a distintos dominios de conocimiento. A partir de estos ejemplos se busca mostrar la capacidad de adaptación de la metodología propuesta a dominios de distinta naturaleza y complejidad, demostrando que la capacidad expresiva que posee es suficiente para modelar procesos con distintos grados de complejidad al igual que ocurre con las notaciones mostradas en el [capítulo 3](#) de este documento.

---

---

## PARTE IV

# DESARROLLO DE PROTOTIPOS Y EVALUACIÓN

---

---



# CAPÍTULO 8

## DESARROLLO DE PROTOTIPOS

---

---

### *8.1 INTRODUCCIÓN*

Una vez establecida la metodología que se propone en este trabajo es necesario realizar los prototipos adecuados para comprobar su idoneidad para el modelado de procesos de negocio y el desarrollo de software dirigido por modelos. Así, se han construido los siguientes prototipos: una herramienta de modelado y un generador de código adaptado a los modelos generados por la herramienta.

A continuación se razonará la selección de la plataforma escogida para el desarrollo de los prototipos y se explicará la estructura de aplicación de la metodología por niveles que acompaña a los prototipos. También se realizará un recorrido por la creación de los prototipos asociados a la metodología para después mostrar la utilización de los mismos para el modelado de los procesos y la generación de código asociados a dos de los procesos mostrados anteriormente.

## **8.2 SELECCIÓN DE LA PLATAFORMA PARA EL DESARROLLO DE LOS PROTOTIPOS**

El **capítulo 4** de esta memoria contiene un estudio o recorrido por algunas de las herramientas y plataformas que ofrecen soporte para el modelado y la ejecución de procesos de negocio. Dadas las características y los objetivos de este trabajo de investigación, se estableció que las plataformas con mayor grado de adaptación a dichas circunstancias son jBPM y Windows Workflow Foundation.

Por lo tanto, antes de comenzar la construcción de los prototipos es necesario realizar una comparación de las características de ambas para establecer cual de ellas es mejor para la construcción de los prototipos. Las características que se estudiarán son las siguientes: el coste de la licencia de uso de las plataformas, la relación con los estándares de ambas alternativas y el entorno de trabajo que se puede ofrecer al usuario con ambas.

### **8.2.1 ¿Software libre o plataforma de pago?**

El hecho de que jBPM sea un producto de software libre y que la plataforma WWF esté incluida dentro del entorno .NET de Microsoft supone una gran diferencia de operatividad. jBPM, que se encuentra integrada dentro del entorno de desarrollo Eclipse, necesita de la configuración y orquestación con otra serie de productos para crear artefactos funcionales a partir de los modelos creados por los expertos del dominio. Por ejemplo, en el caso de la generación de aplicaciones web se hace necesaria la existencia de un servidor web externo al entorno de Eclipse para poder realizar la publicación. Sin embargo, la plataforma WWF dispone de una integración total dentro de la plataforma .NET, donde es posible utilizar otros componentes, como los servidores, de manera transparente y sin necesidad de un proceso de configuración costoso. Esto es posible gracias a la comunicación directa existente entre todos los productos incluidos por Microsoft dentro de su plataforma de desarrollo .NET.

Por otra parte, otro de los aspectos relacionados con este punto es el coste de la licencia necesaria para la utilización de WWF, ya que es necesario disponer de una licencia de uso de Microsoft Visual Studio para poder utilizarla de forma completa y sin limitaciones.

### **8.2.2 Relación con los estándares**

Otro de los aspectos comparables de las herramientas es su relación con los estándares existentes dentro del entorno de BPM. En esta faceta es jBPM la herramienta que mejores resultados ofrece, ya que tanto la representación gráfica de las entidades utilizadas en el modelado como el fichero XML generado a partir del modelado del proceso se acercan mucho al estándar BPMN.

Por otro lado, WWF tiene una relación más lejana con los estándares ya que la representación gráfica de sus entidades se aleja más de la propuesta de BPMN, si bien sus similitudes son apreciables. Las mayores diferencias encontradas en WWF con respecto a los estándares radican en el fichero generado a partir de la definición gráfica de los modelos ya que Microsoft apuesta por unos ficheros con formato XAML y XAMLX, que



contienen tanto la información propuesta por el estándar como otra información adicional no contemplada en el mismo.

### 8.2.3 Entorno de trabajo del usuario

Uno de los objetivos primordiales de este proyecto es involucrar a los expertos de los procesos en el modelado de los mismos, por lo que gran parte del peso de la comparación entre las plataformas se destina al entorno en el que se realiza el modelado.

El entorno propuesto por jBPM se corresponde principalmente con el entorno de desarrollo integrado Eclipse, aunque en las últimas versiones se ofrece la posibilidad de crear una herramienta independiente de este. Sin embargo, el aspecto de esta herramienta sigue siendo muy cercano al de este entorno de desarrollo, circunstancia que no resulta muy recomendable para los expertos de los procesos a los que van orientadas estas herramientas. Además, el coste de construcción y configuración de una herramienta independiente dentro de esta plataforma es elevado en comparación con el de la plataforma WWF. También resulta un factor negativo para esta plataforma el nivel de abstracción que experimentan los usuarios, ya que en algunos casos sería necesario realizar cambios directamente en el fichero XML que representa al proceso, siendo esto contrario a los niveles de abstracción que se pretende alcanzar con las herramientas que se construyan.

Sin embargo, en el caso de WWF el entorno de trabajo que se puede proporcionar a los expertos de los procesos es mucho más adecuado para el cometido del proyecto. WWF ofrece la posibilidad de construir una herramienta de modelado independiente del entorno de trabajo del equipo informático. Dicha herramienta será construida de tal forma que los usuarios posean de una zona de modelado en la que puedan modelar sus procesos de negocio y habilitando las funciones típicas de las aplicaciones de escritorio para el manejo de los ficheros: nuevo proceso, cargar, guardar, guardar como e imprimir. Además, no será necesario que los expertos del proceso realicen ninguna modificación a nivel de ficheros ya que realizando el modelado de forma gráfica con las entidades propias de su dominio, la herramienta se encarga de construir, de forma completa y transparente, el fichero asociado al modelo de proceso construido.

### 8.2.4 Plataforma seleccionada

Los puntos comparativos presentados en este apartado nos permiten tomar una decisión en cuanto a la plataforma a adoptar para la realización del prototipo que acompaña a este proyecto.

En primer lugar, en lo referente al carácter de pago o gratuidad de las herramientas es necesario señalar que el entorno de trabajo existente en la Universidad de Oviedo, donde se cuenta con una licencia de Microsoft que permite la utilización de este componente de la plataforma sin coste adicional, nos permite obviar el coste de la licencia dentro de esta comparación. Además, los casos de estudio incluidos en esta documentación en el [capítulo](#)

7 disponían de un entorno de trabajo similar, por lo que este coste tampoco afectaba al uso de la propuesta dentro de su entorno.

Por otro lado, la cercanía a los estándares resulta un punto de comparación complicado de valorar. Si bien es verdad que la plataforma jBPM es más cercana al estándar en lo que a representación gráfica se refiere, este punto no resulta un factor diferenciador en la elección ya que se ha optado por definir una representación gráfica alternativa para las entidades de la metodología. Dentro de este aspecto de cercanía a los estándares se encuentra también el formato del fichero que almacena los modelos; en este caso, jBPM opta por un fichero XML convencional mientras que WWF almacena sus modelos en un formato XML extendido que se denomina XAMLX, por lo que la estructura de ambos ficheros es muy similar y tampoco resulta un factor determinante. Sin embargo es necesario resaltar que, si en algún momento fuera necesario hacer que los modelos construidos con la metodología BPLOM se almacenaran en el mismo formato que los modelos construidos con el estándar BPMN o con jBPM, no sería complicado construir un artefacto de transformación de los modelos que permitiera almacenarlos en el formato requerido.

En último lugar, el entorno de trabajo que es posible presentar utilizando WWF es mucho más acorde a las características de este proyecto. La posibilidad de construir, de manera sencilla, una herramienta de modelado independiente y altamente configurable tanto en lo referente a interfaz gráfico como a operaciones soportadas supone un gran punto a favor para esta plataforma. A pesar de que las últimas versiones de la plataforma jBPM también ofrecen soporte a esta posibilidad, la construcción y configuración de la herramienta es un proceso más costoso que en el caso de la plataforma WWF. Unido a estas circunstancias se encuentra el hecho de que el nivel de abstracción que experimentan los expertos de los procesos es más elevado si se usa una herramienta construida con la plataforma WWF.

Además de todo lo expuesto anteriormente, también se incluye cierto grado de elección personal en la selección de la plataforma para la construcción de los prototipos que acompañan a este trabajo de investigación. La comodidad que supone trabajar con una única herramienta que cuenta con integración completa en el sistema operativo Windows y el potencial que ofrece la herramienta Visual Studio han resultado atractivos a la hora de tomar la decisión.

A partir de todo esto, se cree que la plataforma adecuada para la realización de las herramientas asociadas a este proyecto es Windows Workflow Foundation [WWF].

### 8.3 ESTRUCTURA DE APLICACIÓN DE LA METODOLOGÍA POR NIVELES

Una vez que se ha formalizado la tecnología que se utilizará para la creación de los prototipos relacionados con este trabajo de investigación, se considera necesario realizar un pequeño recorrido por la estructura de aplicación de la propuesta a través de los prototipos que se presentarán en los siguientes apartados de este capítulo.

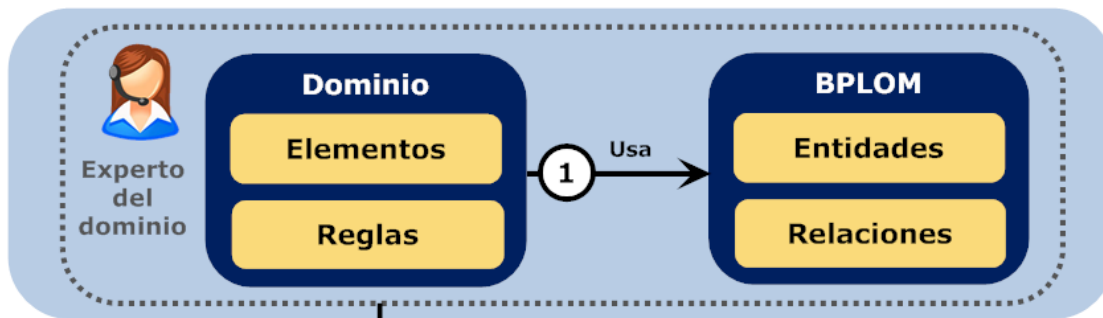
Tal y como se especificó anteriormente, uno de los objetivos de este trabajo de investigación se centra en la construcción de dos prototipos que permitan aplicar la metodología de modelado por niveles que se propone a los casos vistos en el capítulo anterior. Así, la estructura de prototipos que se considera oportuna a la vista de dichos objetivos está compuesta por:

- Una herramienta de modelado de procesos de negocio dirigida a los expertos del dominio del problema, denominada BPLevel Modeler.
- Una herramienta de generación de aplicaciones multigénero y multiplataforma dirigida a los expertos informáticos o usuarios técnicos, denominada BPLevel Generator.

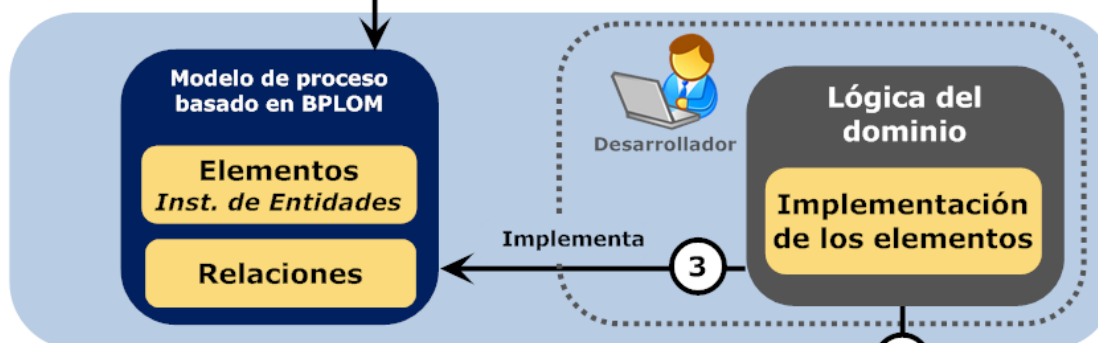
La imagen que se incluye a continuación muestra la estructura de aplicación de la metodología BPLOM para la construcción de aplicaciones personalizadas con soporte para los procesos de negocio descritos en los modelos. En el diagrama se han marcado cuatro fases diferenciadas:

- **P1:** El experto del dominio establece las equivalencias entre los elementos y las reglas del dominio y los tipos de entidades y las relaciones de BPLOM.
- **P2:** El experto técnico se encarga de implementar la lógica de negocio relacionada con cada uno de los elementos identificados por el experto del dominio en la primera fase. Además, este experto técnico se encargará de configurar la herramienta de modelado para que contenga únicamente aquellas entidades relacionadas con el dominio en cuestión.
- **P3:** En esta fase el experto del dominio utiliza la herramienta que le proporciona el experto técnico para modelar su proceso de negocio con aquellos elementos que identificó en la primera fase y que fueron implementados en la segunda fase. El resultado de esta tercera fase es la creación de un modelo del proceso de negocio utilizando la herramienta de modelado.
- **P4:** La última fase es la recoge la creación de una aplicación informática personalizada con soporte para el proceso de negocio a partir del modelo generado en P3. Para ello se utiliza la herramienta generadora de código, que recibe el modelo de proceso, lo analiza y lo traduce a código en base a los parámetros indicados para la ejecución del generador.

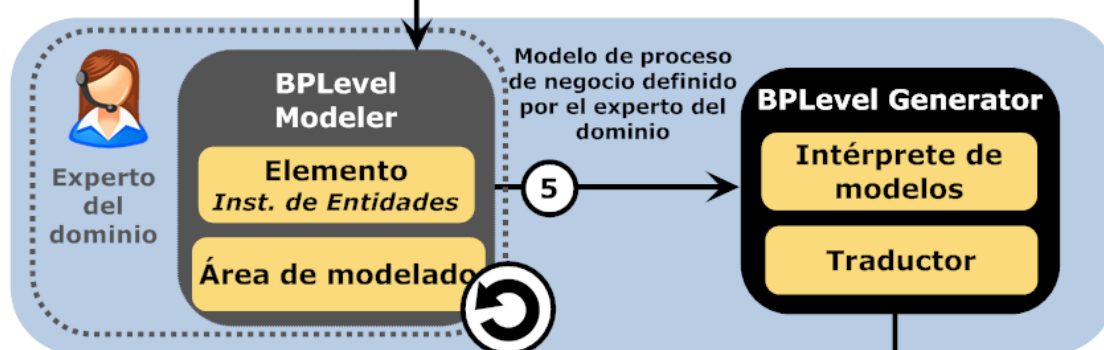
**P1: Definición del dominio específico**



**P2: Lógica de negocio**



**P3: Definición del modelo**



**P4: Proceso de negocio funcional**



Figura 147: Estructura de aplicación de BPLOM en un dominio específico

A partir de la estructura mostrada en la imagen anterior, el resultado de una comparación con la aplicación de BPMN, la notación estándar de modelado de procesos de negocio, sería la que se muestra en la siguiente imagen.

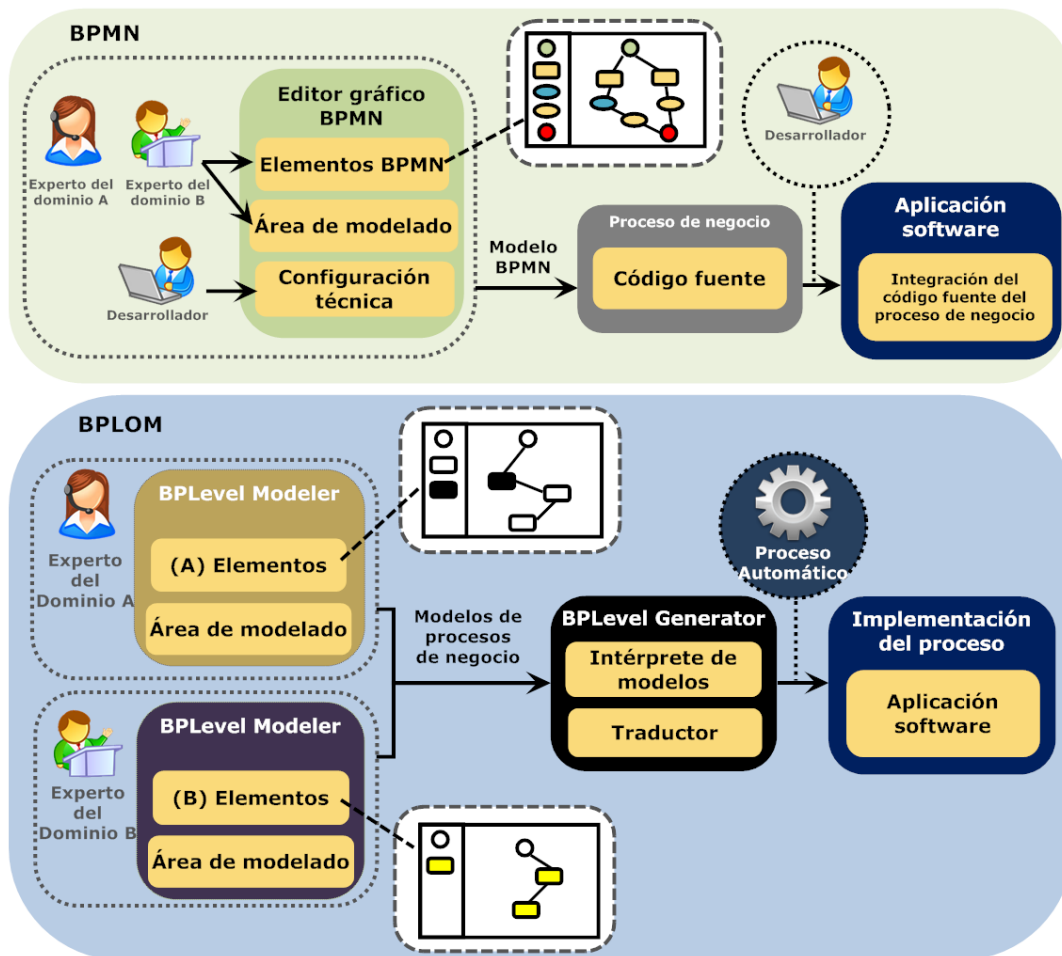


Figura 148: Comparación entre BPMN y BPLoM

Según se observa en el gráfico anterior, la principal diferencia es el grado de participación del experto o desarrollador técnico en el modelado de los procesos y la generación del código. Mientras que en el uso de BPMN su participación es necesaria tanto en el apartado del modelado de los procesos como en la generación de la aplicación con soporte para éstos, al aplicar la metodología por niveles propuesta en esta investigación se observa como una vez que se ha realizado la correspondencia entre los elementos del dominio y los de la metodología (P2 en la Figura 147) el experto no tiene que intervenir de manera activa, ya que será la herramienta generadora de código la que cree la aplicación correspondiente.

Estos gráficos anteriores muestran la aplicación de la metodología de forma abstracta, ya que no se determina el dominio de aplicación. Si bien esto se corresponde con la capacidad de adaptación de la metodología a cualquier dominio donde se utilicen aplicaciones informáticas, es conveniente mostrar un gráfico similar para uno de los dominios mostrados en el capítulo relativo a los casos de estudio. Así, la siguiente imagen muestra la aplicación de la metodología por niveles en el dominio de los videojuegos para dispositivos móviles.

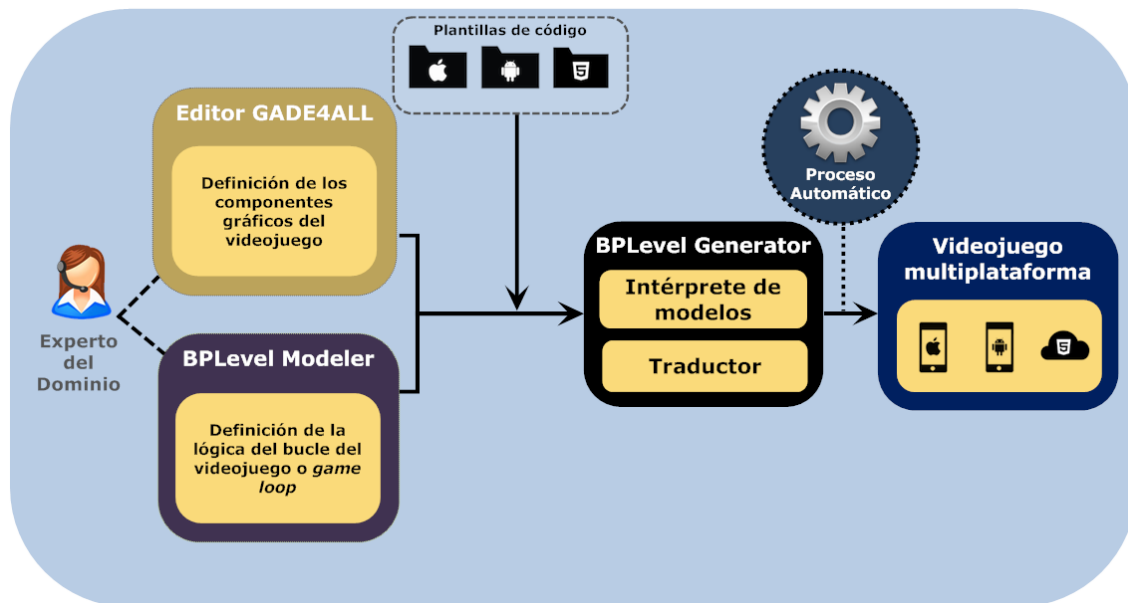


Figura 149: Aplicación de la propuesta al dominio de los videojuegos

En la imagen anterior se observa como el experto del dominio, en este caso el diseñador de videojuegos, interactúa tanto con BPLLevel Modeler como con el editor de videojuegos GADE4ALL, mencionado en el apartado del caso de estudio. Así, el editor de videojuegos le permite definir el aspecto gráfico de las pantallas y los personajes y los niveles que tendrá el juego y BPLLevel Modeler le permitirá definir el bucle del juego que establece el comportamiento de los elementos del juego durante una partida. Cada una de las herramientas anteriores genera sus ficheros de salida y después BPLLevel Generator será el encargado de interpretar dichos ficheros para crear un videojuego funcional para las siguientes plataformas: Android, iOS y HTML5.

### 8.4 BPLEVEL MODELER: HERRAMIENTA DE MODELADO

BPLEvel Modeler es una herramienta de modelado que ofrece soporte a BPLOM a través de las características aportadas al proyecto por Windows Workflow Foundation. Esta herramienta ha sido desarrollada aprovechando la posibilidad de realizar un rehosting del entorno de construcción de los diagramas de flujo existente en Visual Studio 2010; además, el hecho de que este rehosting sea generado en un archivo ejecutable (.exe) permite que sea ejecutado sin ningún problema en cualquier equipo informático que disponga de un sistema operativo Windows en el que esté instalada la versión 4.0 de .NET.

#### 8.4.1 Características de la herramienta

La característica más destacable de esta herramienta es su sencillez debido a la intención de ponerla a disposición de expertos del dominio del problema con pocos conocimientos tanto de informática como de modelado. Así, se ha optado por construir una interfaz con tres zonas claramente delimitadas y que ofrezca las acciones más comunes de cualquier software: nuevo proceso, cargar proceso y guardar proceso.

Otra característica destacable de este software de modelado es el alto nivel de adaptación y la sencillez con la que se logra ajustar su contenido a las necesidades existentes. Los mecanismos ofrecidos por la Windows Workflow Foundation para construir este tipo de herramientas permiten configurar, a través de sencillos cambios en las líneas de código de la aplicación, las entidades que aparecerán en el interfaz gráfico a disposición de los usuarios. Esto supone una ventaja en entornos donde se quiera disponer de distintas versiones de la herramienta de modelado para satisfacer las necesidades de un grupo de expertos para un dominio concreto.

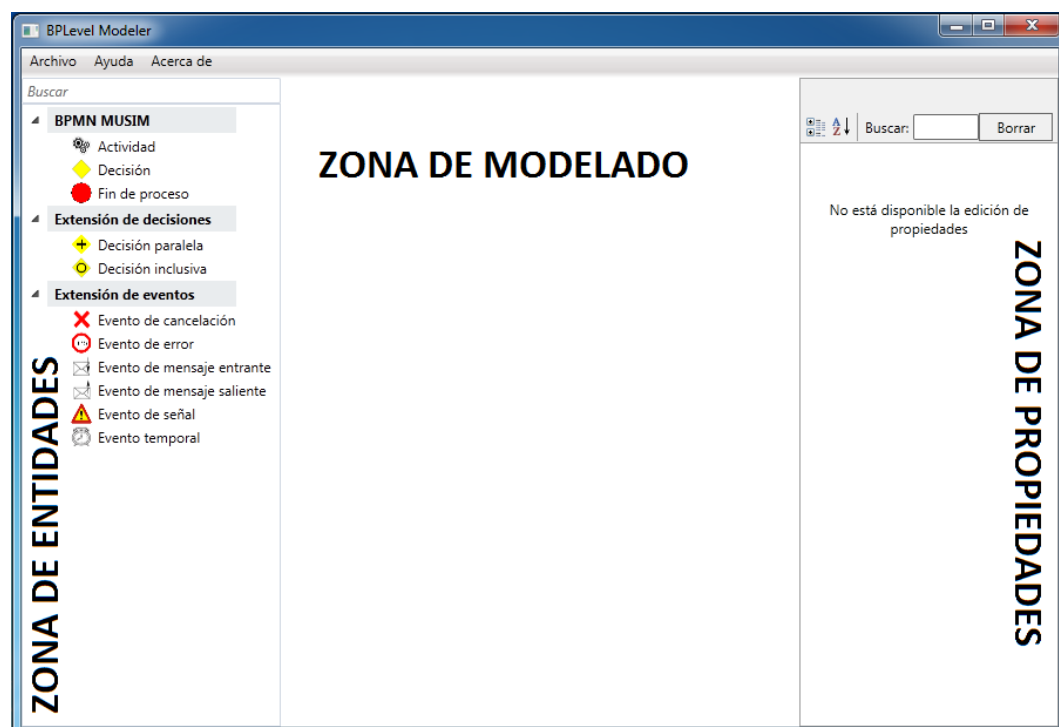


Figura 150: Interfaz gráfico de BPLEvel Modeler mostrando las zonas

#### *8.4.1.1 Zona de entidades*

La zona de entidades es la barra lateral situada más a la izquierda en el interfaz gráfico. Este área contiene todos aquellos elementos de los que disponen los usuarios del software para realizar los modelados de sus procesos. En esta zona será donde se apliquen los cambios más significativos de la herramienta ya que dependiendo tanto del nivel de BPLOM con el que se quiera modelar como del proceso que se vaya a modelar se cargarán en esta zona las entidades necesarias.

En la captura anterior se puede observar como esta zona se encuentra dividida en varias secciones. Las tres primeras secciones se corresponden con los niveles de BPLOM que el usuario ha indicado que quiere utilizar, en este caso los tres primeros niveles de la metodología. Debajo de las secciones correspondientes a los niveles de BPLOM a utilizar se pueden añadir otras secciones para actividades personalizadas desarrolladas para dar soporte a procesos de negocio concretos; este fue el caso de Isastur Modeler, donde aparte de las entidades de BPMN MUSIM se utilizaron dos secciones extra, una para las actividades propias del proceso de gestión de incidencias y otra para las del proceso de contratación de personal.

La definición y el uso de estas actividades personalizadas será abordado más adelante en esta sección, explicando las razones para la utilización de esta característica de Windows Workflow Foundation.

#### *8.4.1.2 Zona de modelado*

La zona de modelado es la parte central de esta aplicación de modelado y será la zona en la que los expertos del dominio irán componiendo el diagrama de su proceso de negocio a través de la utilización de los elementos de la zona de entidades.

La zona de modelado está compuesta por un diagrama de tipo Flowchart en el que los expertos arrastrarán los componentes de la zona de entidades que necesiten para completar su modelo. Además de arrastrar los elementos que compondrán el proceso de negocio los expertos necesitarán conectar estos elementos para establecer el flujo de ejecución de su proceso. Windows Workflow Foundation permite conectar las entidades de forma sencilla, ya que cada elemento que se encuentre en la zona de modelado dispondrá de un área de control que permitirá al experto añadir las transiciones entrantes y salientes de la entidad.

Debido a esta forma de conexión de las entidades se ha tenido que omitir la entidad transición de la sección BPMN MUSIM de la zona de entidades, tal y como se puede observar en la figura anterior.

#### *8.4.1.3 Zona de propiedades*

La zona situada en el extremo derecho del interfaz gráfico de esta herramienta es la zona de propiedades de la herramienta. Dentro de esta sección se muestran las propiedades de



la entidad que está seleccionada en el modelo del proceso que se está realizando. Las propiedades mostradas comprenden tanto el nombre de la entidad seleccionada como las posibles variables de entrada y salida u otra serie de atributos que las actividades puedan necesitar.

El área de propiedades es la zona de la herramienta dirigida a aquellos expertos de dominio que cuentan con un nivel de conocimientos técnicos alto, ya que la mayor parte de las propiedades que aparecen en esta zona se corresponden con funcionalidad técnica de las entidades correspondientes a los niveles más avanzados de la metodología. Por ejemplo, aquellos expertos que utilicen BPLLevel Modeler con las entidades de todos los niveles de BPLOM tendrán la opción de configurar detalles de la entidad Almacén como el conjunto de resultados esperado (un único resultado o una lista de resultados) o la base de datos con la que se quieren conectar de entre la lista de bases de datos disponibles.

## **8.4.2 Programación de la herramienta**

Después de haber expuesto la estructura del interfaz de BPLLevel Modeler es necesario establecer cómo se construye una herramienta de este tipo utilizando Visual Studio 2010. En esta sección se incluirán capturas de pantalla que muestren la especificación de las distintas zonas o áreas que se han presentado así como otra serie de detalles técnicos relacionados con BPLLevel Modeler.

### *8.4.2.1 Definición de la estructura*

El primer paso a dar es la configuración de los componentes que aparecerán en la pantalla de la aplicación, de tal forma que será necesario dividir la pantalla en las secciones que vaya a contener. Para ello, Visual Studio 2010 ofrece la posibilidad de definir la apariencia de la aplicación a través de un fichero XAML.

Según se observa en la siguiente captura, el fichero XAML que representa a la aplicación es un fichero de configuración en el que se indica, entre otros muchos aspectos, la disposición de filas y columnas de la pantalla de la herramienta.

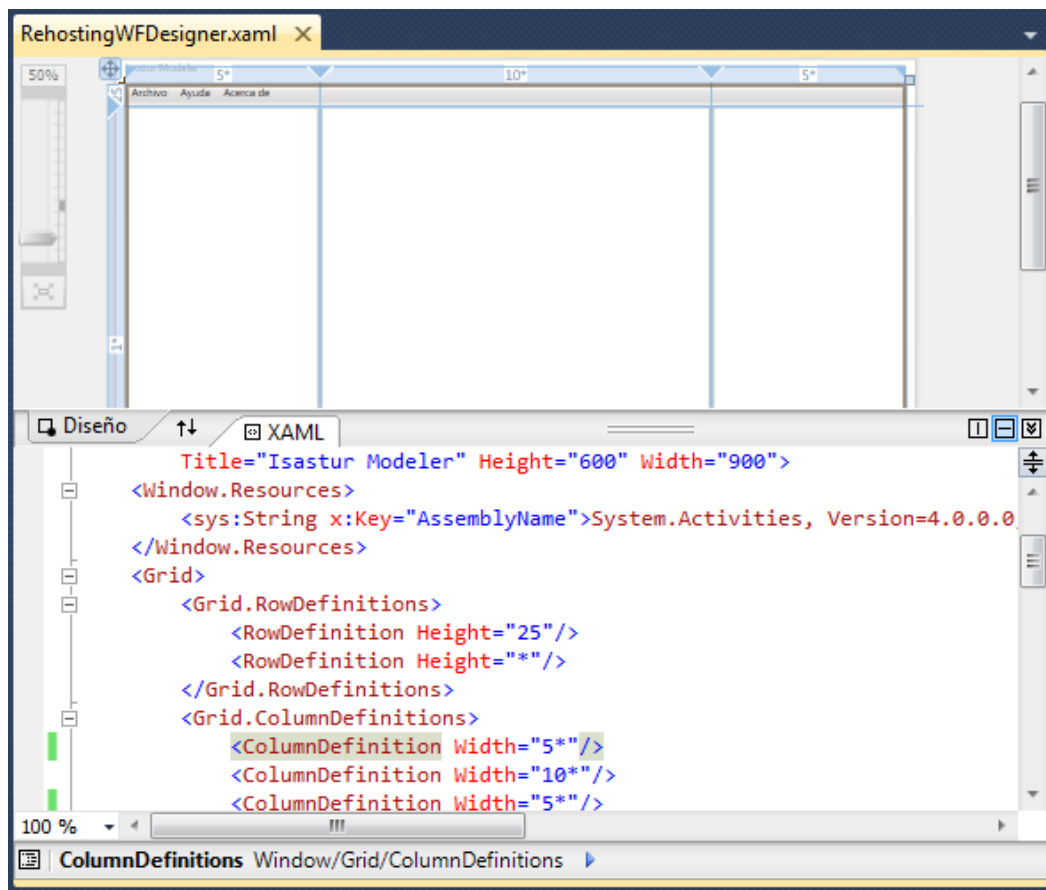


Figura 151: Entorno de definición del interfaz

En este caso la pantalla ha sido dividida en dos filas: una para albergar el menú que permite tener acceso a las acciones que se pueden realizar y otra para albergar las zonas de entidades, modelado y propiedades.

En la imagen anterior también se puede observar que además de definir el número de filas y columnas también se puede establecer el ancho de las mismas; en el caso de BPLLevel Modeler se ha optado por ofrecer el doble de ancho a la zona de modelado, ya que albergará los procesos de negocio y será la zona que reciba la mayor parte de la interacción del usuario.

#### 8.4.2.2 Definición del menú

Una vez definidas cada una de las filas y columnas de nuestro interfaz es necesario establecer sus contenidos.

En primer lugar se establece el contenido de la barra del menú, donde se establece que existirán cinco acciones disponibles en uno de sus componentes y en los otros dos se configura la aparición de unos cuadros de diálogo con los mensajes pertinentes. Esta circunstancia se muestra en la captura incluida a continuación.

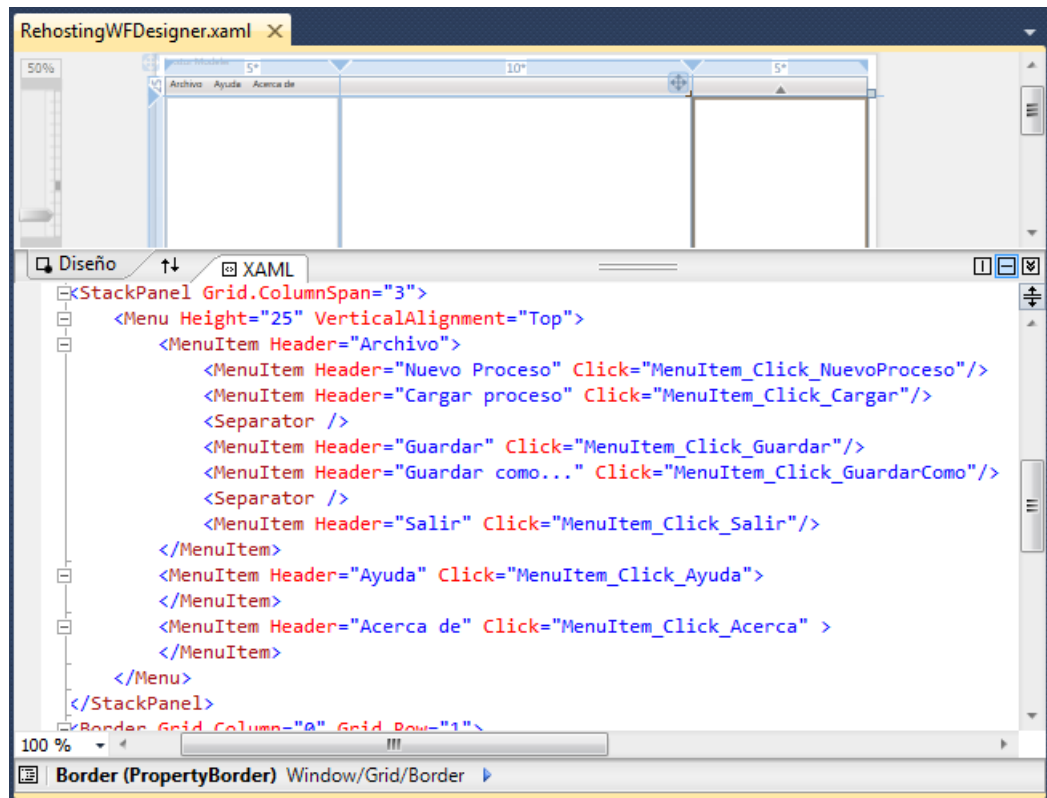


Figura 152: Definición de los elementos del menú

La definición de los elementos del menú conlleva la existencia de una serie de métodos en el fichero de código asociado a esta definición del interfaz. Así, cada vez que se selecciona un elemento del menú, se produce una llamada al método correspondiente, que ejecuta la acción asociada al elemento seleccionado. En la captura anterior también se puede observar cómo se utilizan unos separadores para diferenciar las opciones del menú Archivo, de tal forma que aparezcan separadas de una manera similar a la que aparecen en las aplicaciones informáticas comúnmente utilizadas.

A continuación se incluyen una tabla que permite conocer cuáles son las especificaciones de cada uno de los elementos de menú que se pueden ver en la figura anterior.

Elemento de Menú	Opción	Descripción
<b>Archivo</b>	Nuevo proceso	Esta opción de menú permite iniciar de nuevo el modelado de un proceso, eliminando el contenido del modelo actual en la zona de modelado
	Cargar proceso	Al seleccionar esta opción se le muestra al usuario un cuadro de diálogo en el que se le permite seleccionar un fichero de extensión XAMLX que represente a un modelo realizado con la herramienta para

		cargar su contenido en la zona de modelado.
	Guardar	Este elemento de menú permite guardar el progreso del trabajo actual en un fichero de formato XAMLX. En caso de que ya se hubiera guardado anteriormente se almacenará en el mismo lugar y con el mismo nombre mientras que si se guarda por primera vez se preguntará al usuario dónde y con qué nombre se procederá a guardar el modelo de proceso.
	Guardar como...	Al igual que en el caso anterior esta opción permite almacenar en un fichero el progreso del trabajo actual, aunque en este caso se preguntará al usuario por el nombre con el que se quiere guardar el fichero.
	Salir	Este comando permite abandonar la herramienta de modelado. Si el trabajo actual no se ha guardado se le preguntará al usuario si se desea guardar el progreso antes de salir.
<b>Ayuda</b>	----	Muestra un cuadro de diálogo donde se incluye un pequeño texto en el que se explica el cometido de la herramienta y como se pueden utilizar sus funciones.
<b>Acerca de</b>	----	Muestra un cuadro de diálogo con información sobre la versión de la herramienta.

Tabla 10: Elementos de menú y funcionalidad asociada

### 8.4.2.3 Definición de las zonas principales

Tal y como se mostró en la Figura 150, la zona central del interfaz se encuentra dividida en tres columnas que darán soporte a cada una de las áreas descritas en el apartado de definición de la estructura de la herramienta. Así, al igual que en el caso de los elementos de menú, será necesario definir cuáles serán los contenidos de cada una de estas zonas.

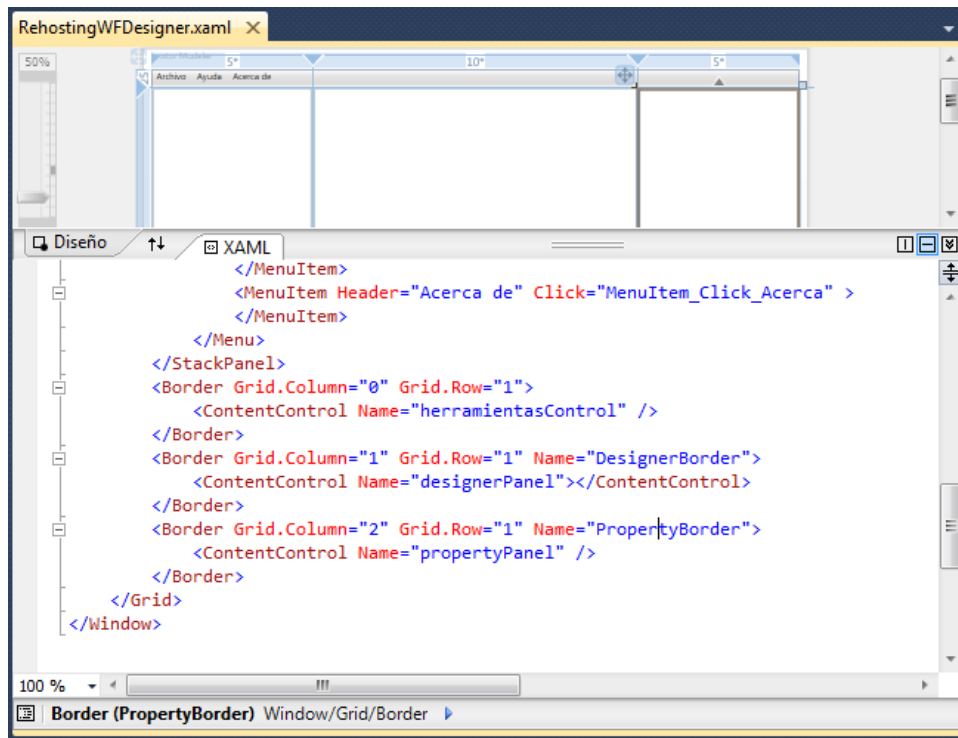


Figura 153: Definición de las zonas principales

La figura anterior muestra la definición de los contenidos de control que se asocian a cada una de las zonas. En este caso los contenidos se cargarán en el archivo de código asociado ya que es necesario declarar la construcción de objetos y hacer otras tareas que no es posible realizar desde este fichero de configuración.

## **8.5 BPLEVEL GENERATOR: HERRAMIENTA DE GENERACIÓN DE CÓDIGO**

Una vez modelados los distintos procesos de negocio es necesario poder utilizar dichos modelos como base para la construcción de aplicaciones. Para ello se ha desarrollado una herramienta que permite analizar y transformar los modelos generados con BPLLevel Modeler para construir aplicaciones informáticas con soporte para dichos modelos de proceso. Según el tipo de aplicación que se quiera esta herramienta será capaz de adaptar su funcionamiento para construir la aplicación correspondiente.

### **8.5.1 Estructura de la herramienta**

BPLLevel Generator es una aplicación de escritorio desarrollada en C# dentro del entorno Visual Studio 2010. Esta herramienta ha sido desarrollada con el fin de ofrecer al servicio de informática una forma sencilla de utilizar los modelados realizados por los usuarios para crear las aplicaciones que ofrezcan soporte a los mismos.

Así, esta aplicación cuenta con un interfaz gráfico sencillo en el que se pide al usuario que complete los campos de información necesarios para crear la estructura y el código de la futura aplicación. Es necesario señalar que algunas de las opciones posibles son excluyentes entre sí, ya que por ejemplo no es posible crear un videojuego o una aplicación de catálogos para las plataformas Java o .NET. La información que el usuario debe introducir para que BPLLevel Generator pueda acometer el proceso de generación es la siguiente:

- La ruta del archivo de proceso generado con BPLLevel Modeler que desea utilizar como base de la aplicación a generar.
- La ruta donde desea que se genere la aplicación.
- El tipo de aplicación que desea crear: web, catálogo o videojuego.
- La plataforma o plataformas para las que desea generar la aplicación: Java, .NET, iOS o Android.

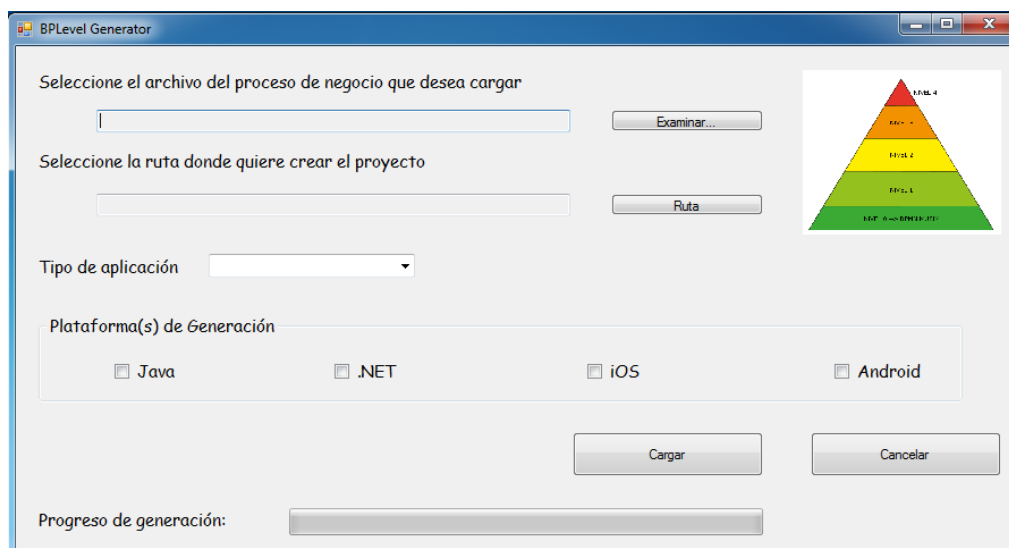


Figura 154: Interfaz gráfico de BPLLevel Generator

## 8.5.2 Funcionamiento de la herramienta

Una vez que el usuario ha introducido los datos correspondientes dentro del interfaz gráfico que se ha presentado anteriormente, BPLLevel Generator está listo para comenzar el proceso de análisis del modelo y la generación del código.

Así, cuando el usuario acciona el botón Cargar, la herramienta generadora de código comienza su funcionamiento y realiza las siguientes tareas:

- Prepara el entorno de creación de la aplicación, utilizando los datos introducidos por el usuario para crea la estructura interna de la aplicación en el lugar indicado usando las plantillas de código.
- Crea la aplicación personalizada según el contenido del modelo de proceso, introduciendo en la estructura creada en el primer paso todos los elementos personalizados que son necesarios.
- Analiza y transforma el modelo de proceso asociado a la aplicación creada, realizando un recorrido por el modelo de proceso para identificar las necesidades funcionales del proceso y configurar la aplicación para que represente dicha funcionalidad.

Estas acciones tienen una forma de ejecución distinta según el tipo de aplicación que se va a generar, tal y como se detallará en el apartado de ejemplo de uso de los prototipos para la construcción de aplicaciones.

## **8.6 EJEMPLOS DE USO DE LOS PROTOTIPOS PARA LA GENERACIÓN DE APLICACIONES**

En esta sección se incluyen detalles sobre la utilización de los prototipos para generar dos tipos de aplicaciones. En primer lugar se muestra como se realizaría la generación de una aplicación web que ofrezca soporte a un proceso de negocio convencional como los de gestión de incidencias o contratación de personal. El segundo ejemplo se centra en la utilización de BPLLevel Modeler y BPLLevel Generator para la generación de un videojuego en colaboración con el editor de videojuegos GADE4ALL.

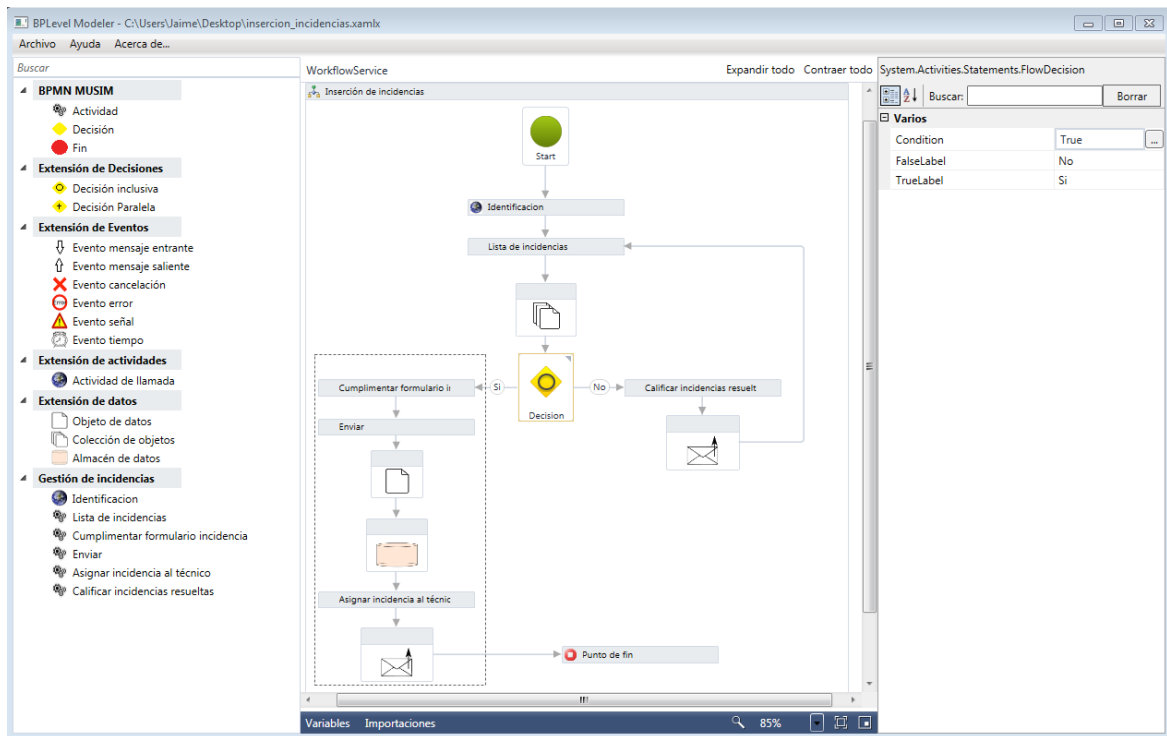
### **8.6.1 Generación de una aplicación web con un servicio asociado**

El experto de negocio utiliza BPLLevel Modeler para modelar su proceso de negocio. Según el nivel de conocimientos de modelado que tenga el experto, la herramienta se adaptará para mostrar únicamente aquellas entidades de la metodología que conoce y sabe utilizar. De esta forma se reducirá la posibilidad de introducción de errores en el modelo de proceso, minimizando la aparición de errores en la captura de requisitos y cumpliendo uno de los objetivos marcados en la investigación. Además, la herramienta mostrará en la zona de entidades las actividades personalizadas correspondientes al dominio del proceso que fueron identificadas en la fase P1 de la estructura de aplicación de BPLOM, de tal manera que el modelo que se genere pueda ser posteriormente utilizado para generar un servicio web ejecutable.

A continuación se muestra el aspecto de la herramienta BPLLevel Modeler con el modelo de nivel 4 correspondiente al subproceso de inserción de incidencias mostrado en uno de los apartados del [capítulo 7](#) de este documento. Tal y como se puede observar en dicha figura, la zona de entidades de la herramienta incluye no solo los elementos relacionados con cada uno de los niveles de la metodología BPLOM que el usuario haya indicado que quiere usar si no también un grupo de entidades relacionadas con el proceso en cuestión, que en este caso se encuentran dentro de la categoría denominada gestión de incidencias.

Estas entidades, específicamente diseñadas e implementadas durante la fase P2 de la estructura de aplicación de la metodología BPLOM contienen la lógica de negocio descrita por el experto del dominio y permiten que el modelo de proceso pueda ser analizado y transformado en un servicio web por la herramienta BPLLevel Generator. En este caso la herramienta de modelado incluye entidades que representan diversos pasos dentro del proceso: la identificación del usuario para la entrada a la aplicación web, la obtención de la lista de incidencias del usuario y la asignación de la incidencia al técnico encargado de resolverla, entre otras.





**Figura 155: BPL Level Modeler mostrando el subproceso de inserción de incidencias**

Una vez que el experto de negocio ha finalizado el modelado de su proceso, BPL Level Modeler le permitirá almacenar el modelo en formato electrónico. Dicho modelo será la base de ejecución de BPL Level Generator, la herramienta de generación de código asociada a la metodología por niveles. Esta herramienta, manejada por el técnico informático correspondiente, será la encargada de construir la aplicación y el servicio web que ofrecen soporte al proceso de negocio modelado por el experto del dominio.

### 8.6.1.1 Preparación del entorno de creación

Con los datos introducidos por el experto informático en la herramienta, BPL Level Generator crea la estructura de una solución de Visual Studio 2010 en la ruta indicada por el técnico como punto para el almacenaje de la aplicación generada. Esta solución de Visual Studio contiene dos proyectos distintos pero estrechamente relacionados: una aplicación web y un servicio web.

### 8.6.1.2 Creación de la aplicación personalizada

La segunda acción que realiza BPL Level Generator es la adición de los formularios necesarios a la estructura de la aplicación web generada. A través de un análisis de las actividades personalizadas que han sido incluidas en el modelo de proceso que se ofrece al generador como parámetro de entrada se determina qué formularios serán necesarios para componer la aplicación web que acompañe al modelo.

El contenido del fichero de modelado del proceso que se construye utilizando BPLLevel Modeler es muy similar a un fichero XML, tal y como se muestra en el fragmento incluido a continuación.

```
<p:FlowStep x:Name="__ReferencelD6">
  <sap:WorkflowViewStateService.ViewState>
    <scg:Dictionary x:TypeArguments="x:String, x:Object">
      <av:Point x:Key="ShapeLocation">200,109</av:Point>
      <av:Size x:Key="ShapeSize">200,22</av:Size>
      <av:PointCollection x:Key="ConnectorLocation">300,131 300,159</av:PointCollection>
    </scg:Dictionary>
  </sap:WorkflowViewStateService.ViewState>
  <cac:Login request="{x:Null}" sap:VirtualizedContainerService.HintSize="200,22" />
</p:FlowStep.Next>
```

A partir de esta característica y teniendo en cuenta el deseo de construir un generador que precise del menor mantenimiento posible, se ha decidido utilizar el lector de XML incluido en la plataforma .NET como el punto de partida para la realización de la inspección del fichero. Así, el primer paso necesario para la realización de este análisis es convertir el fichero en un archivo XML estándar; para conseguir este objetivo sin necesidad de realizar cambios en el fichero original, se ha optado por traspasar el contenido del archivo a una cadena de texto que comienza con la cabecera XML estándar. De esta forma, se logra simular un fichero XML que sigue la especificación oficial y se simplifica el análisis del fichero de modelado.

Una vez que el fichero es accesible a través del lector de XML de .NET es necesario comenzar el análisis del mismo. En el caso de un fichero de modelado como el que nos ocupa, el tipo de nodos XML que nos interesan para la inspección son los elementos; sin embargo, no nos interesan todos los elementos que pueda haber en el fichero sino solamente aquellos que se correspondan con las actividades personalizadas que se hayan programado. Para lograr la distinción de estos elementos frente a los demás se realizará un recorrido por todos los elementos del fichero y se inspeccionarán en detalle aquellos cuyo espacio de nombres se corresponda con el utilizado para la programación de las actividades personalizadas. Para realizar esta operación se utiliza una estructura de tipo *switch* en la que se busca los elementos XML que coincidan con las especificaciones citadas anteriormente.

Cada vez que se encuentra un elemento en el contenido del fichero que cumple la condición de pertenecer al espacio de nombres de las actividades personalizadas, se determina si dicha actividad dispone de un formulario ASPX asociado. Para ello, BPLLevel Generator busca en la ruta que contiene los formularios desarrollados uno que tenga por nombre el mismo que la actividad personalizada; en tal caso el generador copia los archivos relacionados con el formulario a la estructura de proyecto de la aplicación web de tal forma que al final del análisis del fichero todos los formularios necesarios hayan sido incluidos en la estructura de la aplicación web. Esta circunstancia se puede comprobar en la captura anterior.

Además de añadir estos ficheros a la estructura de directorios del proyecto también será necesario modificar uno de los ficheros del proyecto para conseguir la integración total de los formularios dentro del proyecto. Así, una vez que se han copiado todos los archivos

pertinentes a la estructura de directorios del proyecto se procede a abrir el fichero `aplicacionWeb.csproj` para editar su contenido; en concreto, será necesario añadir las siguientes sentencias para cada formulario añadido:

- En primer lugar añadir la sentencia `include` para cada formulario, lo que indica a Visual Studio 2010 que un formulario forma parte de un proyecto.
- En segundo término añadir las sentencias `compile` que indican al entorno de desarrollo los ficheros de código que contienen la funcionalidad de los formularios (los archivos de tipo `.cs` y `.designer.cs`).

En este caso la edición del fichero se realiza de una forma distinta a la manipulación realizada anteriormente: el fichero del proyecto se lee y su contenido se almacena en un vector de `String` para luego proceder a recorrer dicho vector y buscar los puntos del mismo donde se han de incluir las sentencias mencionadas anteriormente; según se va recorriendo el fichero se va almacenando su contenido en una lista a la que también se añaden las sentencias que se han de incluir, para finalmente volcar el contenido de la lista a un fichero que sustituye al fichero original del proyecto.

### 8.6.1.3 Análisis y transformación del modelo de proceso

Una de las principales ventajas del uso de la plataforma .NET es la posibilidad de publicar el modelo de proceso como un servicio web de forma semiautomática. Así, una vez creada en la ruta indicada por el usuario la estructura de la solución necesaria para crear la aplicación web y el servicio con el modelo del proceso, solamente es necesario introducir el fichero que incluye el modelo del proceso en dicha estructura.

El usuario de BPLLevel Generator establece en un primer momento el nombre y la ruta del fichero de modelado que se va a utilizar para la generación, por lo que a partir de éste se procede a copiar dicho archivo. Una vez copiado se utiliza la ruta especificada por el usuario como base para la estructura vacía con el fin de introducir el fichero de proceso en el lugar que le corresponde dentro de ésta.

## 8.6.2 Generación de un videojuego multiplataforma

A la hora de realizar la generación de un videojuego utilizando la metodología descrita en este documento el esquema de funcionamiento de los prototipos es el que se muestra en la figura que representa la aplicación de la propuesta al dominio de los videojuegos, incluida en el [epígrafe 8.3](#) de este documento.

Tal y como se observa en dicho esquema, el experto del dominio de los videojuegos tendrá que utilizar tanto la herramienta GADE4ALL como la herramienta BPLLevel Modeler. Con la primera diseñará gráficamente todos los elementos del videojuego que desea generar mientras que con la segunda modelará la funcionalidad del bucle del juego. En la imagen incluida a continuación se puede observar una captura de pantalla de la aplicación BPLLevel Modeler mostrando el proceso de negocio que describe la lógica de un videojuego

de estrategia por turnos. Al igual que ocurriera en el caso anterior, BPLLevel Modeler muestra las entidades de los cuatro niveles de la metodología BPLOM pero, además, muestra las entidades relacionadas con los videojuegos de estrategia por turnos dentro de una categoría homónima. Estas entidades son las que han sido identificadas por el experto del dominio en el paso P2 de la estructura de aplicación de la metodología y son las entidades que permiten configurar el juego que se va a generar a partir de la plantilla de código correspondiente.

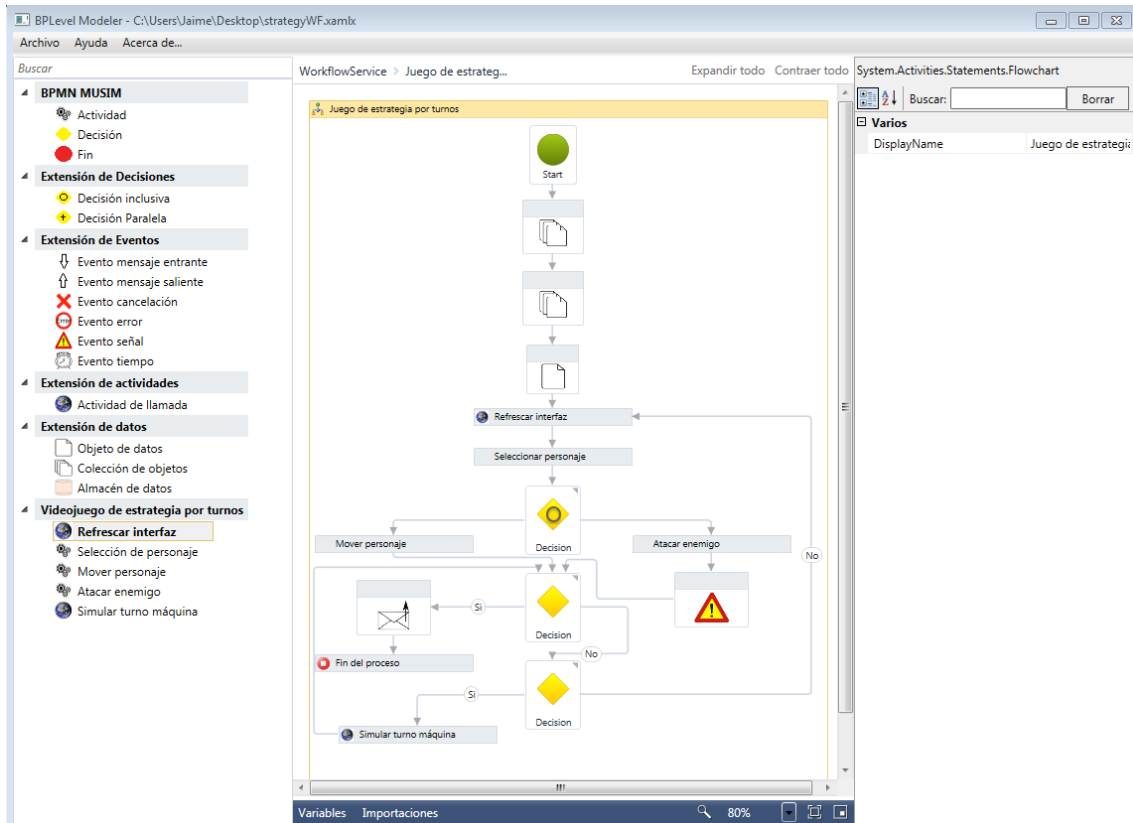


Figura 156: BPLLevel Modeler mostrando el proceso del videojuego de estrategia por turnos

Una vez que el usuario dé por finalizado el diseño gráfico del juego y haya modelado el proceso que representa la lógica del mismo, se procederá a la generación del código del videojuego para las plataformas seleccionadas. La generación, igual que en el caso anterior, se realiza en tres fases, tal y como se muestra en la siguiente captura.

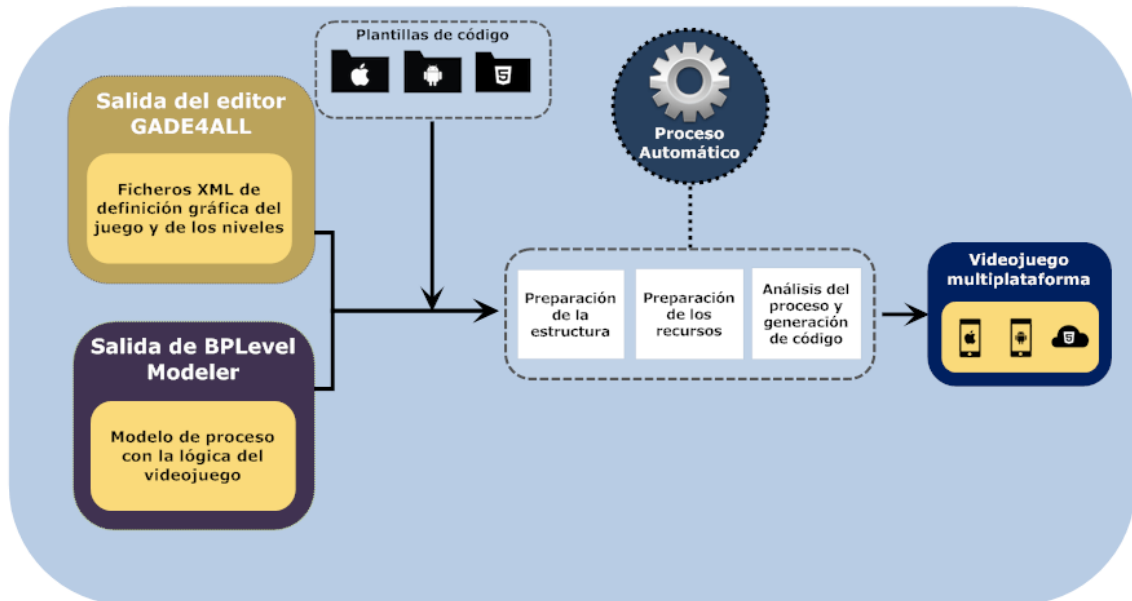


Figura 157: Funcionamiento interno de BPLLevel Generator al generar videojuegos

A continuación se incluye una descripción más detallada de los pasos que BPLLevel Generator acomete en cada una de las tres acciones anteriores para el caso de los videojuegos.

### 8.6.2.1 Preparación de la estructura

La preparación de la estructura se realiza teniendo en cuenta dos de los datos que el usuario introduce en el interfaz gráfico de BPLLevel Generator: la ruta de generación de código y la ruta donde se almacenó el resultado del diseño realizado con la herramienta GADE4ALL.

BPLLevel Generator toma ambos datos y, extrayendo el tipo de juego modelado desde uno de los ficheros generados por la herramienta GADE4ALL realiza la copia de la plantilla de código correspondiente en la ruta indicada.

### 8.6.2.2 Preparación de los recursos

La preparación de los recursos está directamente relacionada con el contenido generado por la herramienta GADE4ALL, donde se encuentran todos los ficheros de recursos necesarios para el funcionamiento del juego y el fichero dsl.XML con los datos de uso para dichos recursos.

Así, BPLLevel Generator realiza en esta fase la copia de los recursos desde la carpeta donde fueron almacenados por la herramienta GADE4ALL a la carpeta que el usuario ha proporcionado como destino del proceso de generación. Una vez que se ha realizado dicha copia, BPLLevel Generator se encarga de analizar el fichero dsl.XML para extraer la configuración de los elementos gráficos e introducir dicha configuración en el archivo de código correspondiente dentro del proyecto de código generado.

### *8.6.2.3 Análisis del proceso y generación de código*

La última fase de BPLLevel Generator a la hora de realizar la generación del videojuego es el análisis del proceso de negocio modelado con la herramienta de modelado para configurar el código del videojuego de tal forma que su funcionamiento sea el descrito en el modelo de proceso generado.

Para ello BPLLevel Generator se encarga de procesar el modelo de proceso desde el nodo inicial hasta el nodo final para conocer el orden en el que han de realizarse todas las operaciones dentro del bucle del juego. Una vez analizado el modelo, BPLLevel Generator configura el código del juego de tal forma que todas las operaciones se realicen en el mismo orden que se ha indicado en el modelo y con la configuración que se indica en el modelo del proceso.

## *8.7 CONCLUSIONES*

Este capítulo incluye la explicación de los prototipos que acompañan a la metodología de modelado BPLOM: BPLevel Modeler, una herramienta de modelado de procesos de negocio, y BPLevel Generator, una herramienta de generación de aplicaciones basada en los modelos de negocio construidos con BPLevel Modeler. A través del uso de estas dos herramientas es posible modelar un sistema utilizando la metodología BPLOM y construir la aplicación informática que ofrezca soporte a dicho sistema.

Ambas aplicaciones han sido desarrolladas bajo la plataforma .NET para aprovechar la integración total ofrecida por Microsoft para la plataforma Windows Workflow Foundation y han sido diseñadas de tal forma que su uso fuera el más sencillo posible. Así, se consigue que los expertos del dominio sin conocimientos informáticos puedan utilizar la herramienta de modelado sin limitaciones en base a sus conocimientos técnicos y que la generación de aplicaciones también pueda acometerse de forma sencilla.

Éstos prototipos serán utilizados en el proceso de pruebas que se describe en el siguiente capítulo. Dicho escenario de pruebas se centra en comprobar la idoneidad de las herramientas para acometer las tareas para las que fueron diseñadas así como para realizar mediciones sobre la mejora que supone la propuesta en el proceso de construcción de artefactos software.





# CAPÍTULO 9

## EVALUACIÓN

---

### 9.1 INTRODUCCIÓN

Una vez que se ha realizado la introducción de la propuesta de metodología por niveles para BPM a través de la definición de sus componentes, del caso de estudio y de los prototipos que la acompañan, llega el momento de describir el proceso de evaluación seguido para evaluar el impacto que supone la definición de la metodología.

Este proceso de evaluación ha sido dividido en dos partes claramente diferenciadas. Por un lado se han realizado pruebas para conocer la idoneidad y el grado de usabilidad de los componentes del nivel inicial de la metodología, BPMN MUSIM, a través de una serie de test de usabilidad a los que fueron sometidos los expertos del dominio. En segundo lugar se realizaron pruebas con los prototipos introducidos en el capítulo anterior para establecer los siguientes detalles:

- La usabilidad de BPLLevel Modeler, la herramienta de modelado propuesta, en comparación con la herramienta de modelado de procesos de negocio incluida en la plataforma jBPM.
- La reducción de esfuerzo en el desarrollo que supone la utilización de una herramienta generadora como BPLLevel Generator a la hora de construir aplicaciones informáticas.

Algunos de los resultados que se presentan en este apartado ya han sido incluidos en algunas de las publicaciones científicas relacionadas con el trabajo de investigación descrito en este documento [Solís-Martínez et al. 2013] [Solís-Martínez et al. 2014].

## 9.2 EVALUACIÓN DE LA METODOLOGÍA POR NIVELES

El primer paso acometido durante la evaluación de la propuesta realizada en este trabajo de investigación se basa en conocer el grado de usabilidad de la metodología de modelado por niveles. Para ello se realizaron una serie de test de usabilidad a los expertos del dominio que utilizaron la metodología BPLOM para modelar sus procesos de negocio.

Teniendo en cuenta la naturaleza incremental de la metodología BPLOM y que la definición de sus componentes se realizó en dos fases diferenciadas, definiendo en un primer lugar el nivel inicial de la metodología para conocer si este se ajustaba a las necesidades de los expertos y completando la definición de la metodología en una fase posterior, las pruebas con los niveles se realizaron también por etapas. Así, los expertos del dominio relacionados con la definición de los procesos de negocio tradicionales fueron los encargados de cumplimentar los test de usabilidad relacionados con BPMN MUSIM, el nivel inicial de la metodología mientras que los expertos relacionados con los procesos de definición de la lógica de los videojuegos fueron los encargados de afrontar el proceso de pruebas del resto de niveles.

### 9.2.1 Proceso de pruebas de BPMN MUSIM

A continuación se incluye una captura imagen en la que se muestra el test de usabilidad que fue completado por los expertos del dominio que realizaron las pruebas relacionadas con el nivel inicial de la metodología BPLOM. Este test de usabilidad también contenía una serie de preguntas relacionadas con una versión de la herramienta de modelado BPLLevel Modeler que serán mostradas y discutidas en el siguiente apartado de este capítulo.

#### Preguntas Sobre la Notación

*A la hora de modelar, ¿ha echado en falta algún símbolo?. En caso afirmativo indique cual.*

.....

.....

*De los símbolos presentados a continuación, ¿cuáles considera más sencillos de comprender?*




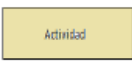




Símbolo	BPMN	BPMN MUSIM	Opción Escogida (BPMN/BPMN MS)
Inicio			
Actividad			
Decisión			
Fin			

Figura 158: Test de usabilidad sobre BPMN MUSIM

Tal y como se puede observar en la imagen anterior, el test de usabilidad relacionado con BPMN MUSIM tenía dos objetivos claramente definidos.

- Por un lado, era preciso conocer si existía la necesidad de incluir algún símbolo adicional a este nivel básico de la metodología para que pudiera ser utilizado para definir procesos de negocio de cualquier dominio. Es por esto que los expertos debían indicar si habían echado en falta algún símbolo durante la utilización de BPMN MUSIM para modelar sus procesos de negocio.
- En segundo lugar el test de usabilidad contenía una tabla en la que se mostraba una comparación de la representación gráfica de los símbolos de BPMN MUSIM con los de la notación estándar de modelado BPMN. Con las respuestas a esta comparación se intentaba corroborar que la inclusión de un código de color en los símbolos de la metodología resultaba en un beneficio a la hora de comprender los modelos construidos.

Los resultados que los expertos de negocio concedieron a estas preguntas sobre la notación BPMN MUSIM son los incluidos en las tablas que se muestran a continuación. En esta fase inicial de pruebas se realizaron un total de 31 test de usabilidad.

¿Echa en falta algún símbolo en BPMN MUSIM?		
Nº resp. afirmativas (Sí)	Nº resp. negativas (No)	Nº resp. NS/NC
3	27	1

Tabla 11: Respuestas sobre falta de símbolos en BPMN MUSIM

Facilidad de comprensión de símbolos (BPMN vs BPMN MUSIM)			
Símbolo analizado	BPMN	BPMN MUSIM	IGUAL
Punto de inicio	9	22	--
Actividad	9	19	3
Decisión	7	24	--
Punto de fin	9	22	--

Tabla 12: Respuestas sobre facilidad de comprensión de símbolos

En base a las respuestas que se muestran en las tablas anteriores, se han construido dos gráficos que permiten realizar un resumen de las respuestas proporcionadas por los expertos del dominio.

En primer lugar se muestra un gráfico circular de porcentajes donde se puede observar que el 87% de los expertos del dominio consultados en esta primera fase de pruebas no consideran necesaria la inclusión de ningún otro símbolo en el nivel inicial o básico de la metodología, frente al 10% de los usuarios que sí consideran necesario incluir algún otro

símbolo y al 5% de expertos que no saben o no contestan la pregunta. Es necesario señalar que prácticamente la totalidad de los expertos que respondieron afirmativamente a la pregunta establecieron que consideraban necesario disponer de una entidad que permitiera modelar la ejecución en paralelo dentro de los procesos.

### ¿Echa en falta algún símbolo en BPMN MUSIM?

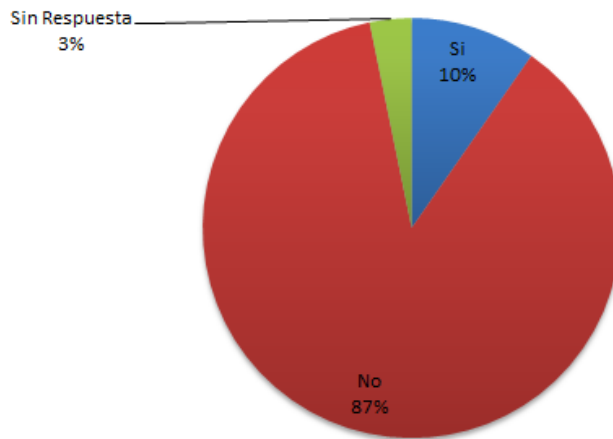


Figura 159: Gráfico de porcentaje sobre falta de símbolos en BPMN MUSIM

En lo referente a la facilidad de comprensión de los símbolos que representan a los elementos incluidos en la definición de BPMN MUSIM, el nivel inicial de la metodología, se observa que para todos los casos al menos un 61% de los encuestados considera que los símbolos de BPMN MUSIM son más fáciles de comprender que los símbolos usados por la notación estándar de modelado BPMN, que no llegan a superar el 30% en ninguno de los casos. En este gráfico también se puede observar que el símbolo que representa a la actividad es el que menor diferencia obtiene en la facilidad de comprensión, existiendo aproximadamente un 10% de usuarios que no encuentran diferencia alguna entre los símbolos.

### ¿Qué símbolos comprende mejor?

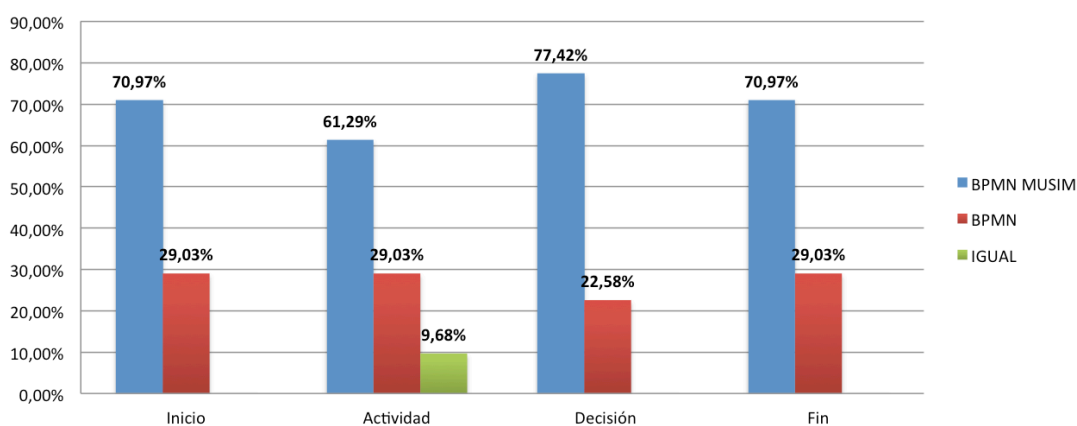


Figura 160: Gráfico de barras sobre comprensión de símbolos

## 9.2.2 Proceso de pruebas del resto de niveles

Una vez realizadas las pruebas sobre BPMN MUSIM, el nivel introductorio de la metodología de modelado por niveles BPLOM, se realizaron unos test de usuario para conocer la diferencia de complejidad entre los niveles definidos y la idoneidad de orden en el que éstos se presentan al usuario. Con esta prueba se pretende establecer si existe un grado de proporcionalidad entre el aumento de la complejidad que los usuarios experimentan al pasar de un nivel a otro de la metodología.

Además, se aprovechó este escenario de pruebas para preguntar a los usuarios acerca de la idoneidad de las entidades escogidas para la composición de la metodología, preguntando la importancia que éstos le dan a cada una de las entidades. De esta forma se intentó hacer una encuesta similar a la realizada por Recker en uno de sus trabajos [Recker 2008], donde se establecía la importancia que los usuarios de BPMN otorgaban a algunas de las entidades incluidas en dicha notación de modelado (*véase epígrafe 3.4.1.4*).

Los sujetos escogidos para la realización de estos test de usuarios fueron expertos del dominio con conocimientos técnicos y de modelado altos, ya que era necesario que pudieran utilizar todos los niveles de la metodología para poder realizar la prueba de forma satisfactoria.

### 9.2.2.1 Evaluación de los niveles de la metodología

Uno de los aspectos principales de la metodología de modelado BPLOM es su adaptación al nivel de conocimiento técnico y de modelado que poseen los expertos del dominio que la utilizan. Esta capacidad de adaptación viene determinada por la existencia de niveles de entidades que añaden complejidad y expresividad de forma incremental.

Para favorecer la implicación de los expertos del dominio en el proceso de captura de requisitos a través del modelado de procesos de negocio se considera importante ofrecer un nivel inicial de la metodología suficientemente simple para que pueda ser utilizado por expertos sin conocimientos técnicos y aumentar la complejidad y expresividad del lenguaje de modelado de forma proporcional. De esta manera es posible introducir a los usuarios de forma gradual y sostenida a los beneficios del modelado de procesos de negocio, obteniendo conocimientos de modelado en base al uso que hacen de la metodología y las herramientas que se les proporciona. Este escenario es distinto al mencionado por White y Miers en su guía de uso de BPMN [White and Miers 2008], donde se establece que los expertos del dominio que quieran utilizar la metodología estándar para el modelado de sus procesos de negocio deberán recibir algún tipo de curso formativo antes de poder comenzar a utilizarla.

Esta prueba se realizó en dos fases diferenciadas: en primer lugar los encuestados respondieron a la idoneidad de ordenación de los niveles y en segundo término indicaron su percepción sobre el aumento proporcional de la complejidad entre los niveles de la metodología. Las cuestiones incluidas en este test fueron *“Indique cómo de adecuada*

considera la ordenación de los niveles de la metodología BPLOM” y “Establezca el grado de complejidad resultante del cambio de nivel en cada caso”. Para la realización de estos test se optó por la utilización de una escala de Likert [Likert 1932], donde los usuarios califican las preguntas que se les realizan según su grado de aceptación de las mismas. En este caso se ha optado por utilizar una escala de Likert de 5 puntos, donde los usuarios podían calificar los puntos mencionados anteriormente de la siguiente forma: 1- muy adecuado/a, 2- adecuado/a, 3- normal, 4- inapropiado/a y 5- muy inapropiado/a.

Las tablas incluidas a continuación muestran los resultados dados por los 30 expertos que realizaron el test. En primer lugar se observan los datos relacionados con la idoneidad de ordenación de los niveles incluidos en la metodología mientras que la segunda tabla incluye una entrada con resultados para cada salto entre niveles.

Idoneidad de la ordenación de los niveles de la metodología BPLOM				
1- Muy adecuada	2- Adecuada	3- Normal	4- Inapropiada	5- Muy inapropiada
8	13	5	5	--

Tabla 13: Idoneidad de la ordenación de los niveles de la metodología BPLOM

Aumento de complejidad resultante del cambio de nivel					
Cambio de nivel que se produce	1- Muy adecuado	2- Adecuado	3- Normal	4- Inapropiado	5- Muy inapropiado
Cambio del nivel 0 al nivel 1	10	15	6	--	--
Cambio del nivel 1 al nivel 2	12	10	8	1	--
Cambio del nivel 2 al nivel 3	8	8	12	3	--
Cambio del nivel 3 al nivel 4	14	10	7	--	--

Tabla 14: Aumento de complejidad resultante del cambio de nivel

Tal y como se observa en Tabla 13, los expertos del dominio consideran que la ordenación de los niveles de la metodología BPLOM es muy adecuada o adecuada en más del 60% de los casos, por lo que se puede considerar que la ordenación propuesta para los niveles de

la metodología responde de forma satisfactoria a las expectativas de los expertos del dominio con conocimientos técnicos y de modelado suficientes para la utilización de todos los niveles de la metodología BPLOM. Además, es posible destacar que únicamente el 16% de los encuestados ha calificado la propuesta de inapropiada, por lo que el porcentaje de expertos que considera la propuesta como razonable es del 84%. Los datos pueden observarse de manera más clara en el siguiente gráfico de porcentajes.

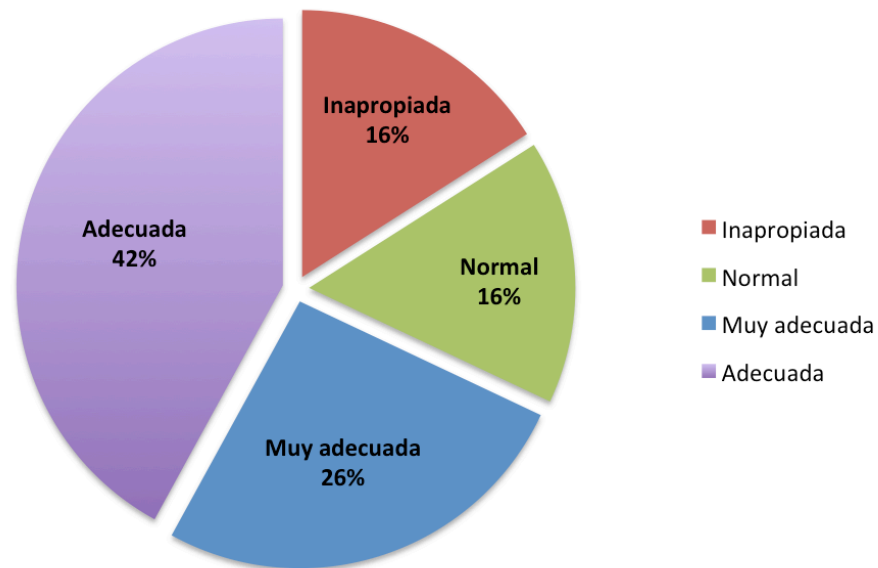


Figura 161: Gráfico de porcentajes sobre la idoneidad de la ordenación de los niveles

En lo relativo al aumento de la complejidad con el cambio entre niveles se observa que el salto de niveles más criticado por los expertos del dominio es el que se produce desde la extensión de eventos (nivel 2 de BPLOM) a la extensión de actividad (nivel 3 de BPLOM); los expertos del dominio han calificado este salto como normal o inapropiado con casi un 50%, un valor bastante superior al de todos los demás cambios entre niveles. En el resto de casos se observa una clasificación más o menos equitativa, siendo todos los casos calificados como muy apropiados o apropiados por un porcentaje entre el 75% y el 80%. Es reseñable que en ninguno de los casos se ha producido una calificación de inapropiado mayor del 10% y que la respuesta muy inapropiado no ha sido utilizada por ninguno de los expertos del dominio en ningún caso.

A modo de resumen gráfico de los datos de la Tabla 14 se ha compuesto el siguiente gráfico de barras donde se puede observar el porcentaje, por cada cambio de nivel, de cada una de las valoraciones posibles. Es necesario señalar que se ha omitido la barra para la columna relacionada con la respuesta número 5, muy inapropiado, ya que ninguno de los usuarios ha utilizado dicha respuesta en ninguno de los casos. En la gráfica se puede observar el predominio de las barras azules (muy apropiado) y morado (apropiado) en 3 de los 4 grupos de datos, mientras que en el grupo restante predomina la barra verde (normal) frente al resto. Es necesario señalar como la barra roja, relacionado con la respuesta inapropiado, aparece solamente en 2 de los 4 grupos de datos y con porcentajes reducidos.

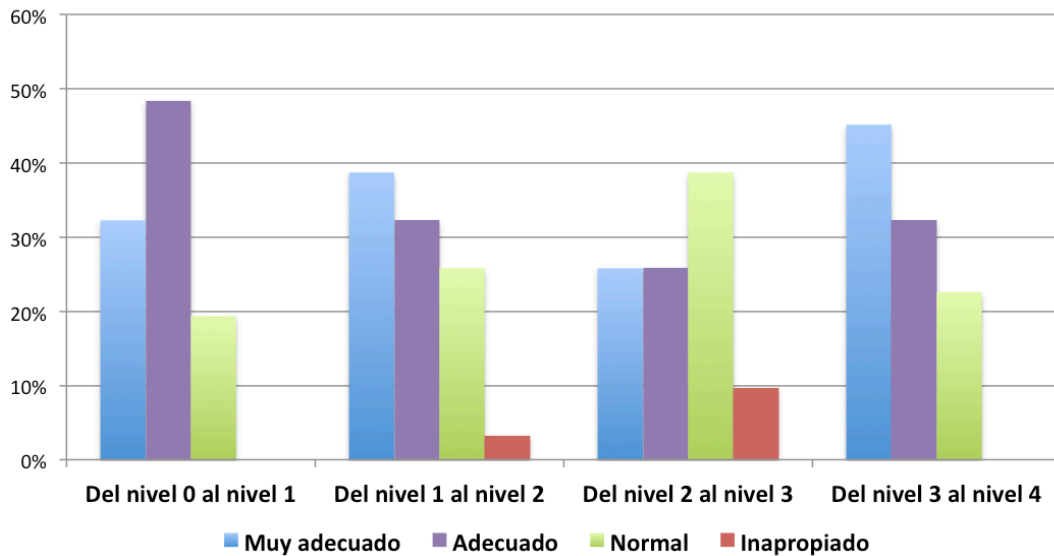


Figura 162: Gráfico de barras sobre aumento de complejidad entre niveles

### 9.2.2.2 Evaluación de la importancia de las entidades incluidas en BPLOM

Tal y como se vio en el [capítulo 3](#) de este documento, donde se realiza un recorrido por algunas de las notaciones de modelado de procesos de negocio existentes, uno de los resultados principales de algunas de las investigaciones referenciadas **[Recker 2008]** **[Chinosi and Trombetta 2012]** está relacionado con la falta de uso que experimentan algunas de las entidades incluidas en la notación estándar de modelado de procesos de negocio BPMN. El grado de importancia que los usuarios de dicha notación le otorgan a las entidades existentes en esta notación está estrechamente relacionado con esta circunstancia, pudiendo observarse una tendencia de falta de importancia en aquellas entidades que experimentan tasas de uso más bajas.

Aprovechando el escenario de pruebas existente se decidió realizar una encuesta de valoración de cada una de las entidades incluidas en la metodología de modelado BPLOM, con el fin de comprobar cuáles eran las entidades que los usuarios consideraban menos importantes dentro de la definición de la metodología. Los resultados relativos a la importancia que le otorgan los expertos del dominio a cada una de las entidades incluida en la metodología BPLOM se pueden consultar en la siguiente tabla, donde se califica cada entidad de la siguiente forma: muy importante, normal y poco importante. Estas categorías pueden considerarse análogas a las utilizadas por Recker en su estudio **[Recker 2008]** que tuvo como resultado la gráfica incluida en el [epígrafe 3.4.1.4](#)



Importancia de las entidades incluidas en la metodología BPLOM			
Entidad analizada	Muy importante	Normal	Poco importante
Punto de inicio	100%	--	--
Punto de fin	100%	--	--
Actividad	100%	--	--
Decisión	100%	--	--
Relación	100%	--	--
Decisión inclusiva	40%	40%	20%
Decisión paralela	64%	30%	6%
Punto de convergencia	62%	19%	19%
Evento de mensaje	72%	28%	--
Evento temporal	23%	31%	46%
Evento de error	77%	--	23%
Evento de cancelación	77%	--	23%
Evento de señal	63%	37%	--
Actividad de llamada	84%	16%	--
Subproceso	20%	32%	48%
Subproceso de evento	20%	32%	48%
Objeto de datos	100%	--	--
Colección de objetos	100%	--	--
Almacén	77%	--	23%

Tabla 15: Importancia de las entidades incluidas en la metodología BPLOM

Los resultados mostrados en la tabla anterior muestran como los expertos del dominio consideran que la mayor parte de las entidades incluidas en la metodología BPLOM llevan asociado un grado de importancia alto o normal, existiendo solamente 3 entidades (evento temporal, subproceso y subproceso de evento) consideradas como poco importantes por

más de un 40% de los usuarios. Con porcentajes de poca importancia alrededor del 20% se encuentran entidades como la decisión inclusiva, el punto de convergencia o el almacén, entre otros. Realizando un gráfico similar al incluido en el trabajo de Recker [Recker 2008] mencionado anteriormente en este apartado de la evaluación, se obtiene una tabla como la que se muestra a continuación. Según se observa en esta figura, existe un predominio de los colores beige y granate en la mayor parte de las entidades de las entidades de la metodología BPLOM lo que indica que los usuarios consideran que estas tienen una importancia alta o normal. En cuanto a los casos de tasas de importancia baja cabe destacar los resultados obtenidos por las entidades evento temporal, subproceso y subproceso de evento, que obtienen calificaciones de poca importancia en un porcentaje de usuarios cercano al 50%.

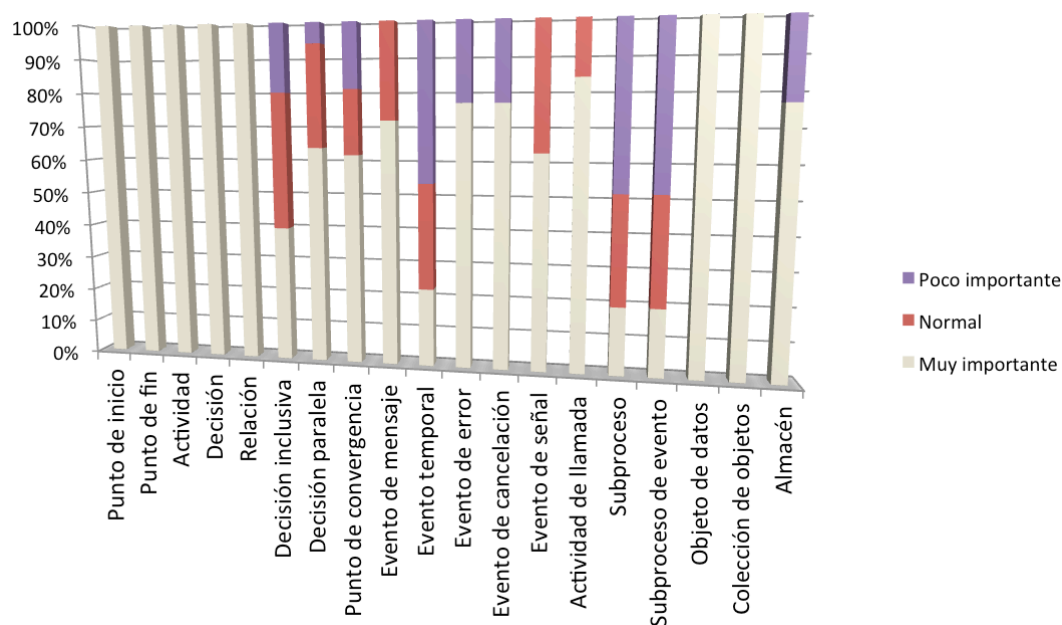


Figura 163: Gráfico de importancia de las entidades de la metodología BPLOM

Aprovechando la introducción de la metodología BPLOM en distintos dominios de conocimiento se realizó un análisis de los resultados anteriores para determinar la posibilidad de que existiera algún tipo de relación entre el dominio y el tipo de artefacto que se va a generar a partir del modelo y la percepción de la importancia de las entidades de modelado disponibles en la metodología. Separando los resultados anteriores según los dominios de conocimiento (procesos de negocio tradicionales y videojuegos para dispositivos móviles) representados por los expertos del dominio que hicieron la prueba, se pudieron observar las siguientes tendencias:

- Las decisiones inclusivas tienen un mayor impacto en los procesos de negocio tradicionales que en los procesos que representan la lógica de un videojuego. Por el contrario, la decisión paralela y los puntos de convergencia son importantes en ambos dominios.
- Los eventos de mensaje y de señal son utilizados con cierto grado de importancia en ambos dominios, ya que la utilización de mensajes y el paso de avisos entre componentes son una constante en ambos casos. Sin embargo, los eventos de

cancelación y error parecen más indicados para los procesos de negocio tradicionales, donde las aplicaciones generadas necesitan responder ante circunstancias como las que especifican este tipo de eventos. El único evento que parece poco importante en ambos dominios es el evento temporal.

- Los subprocesos, tanto los normales como los de evento, obtienen tasas de importancia prácticamente nulas cuando son evaluados por los expertos del dominio de los videojuegos. Esto puede ser debido a que no existe el concepto de error a la hora de la ejecución del bucle del juego mientras que una aplicación informática tradicional si tiene que saber responder ante este tipo de circunstancias.
- Los objetos de datos y las colecciones de objetos son importantes para todos los expertos del dominio que respondieron al test, ya que tanto los videojuegos como los procesos tradicionales manejan este tipo de construcciones. Por el contrario, la entidad almacén parece carecer de importancia en el dominio de los videojuegos, ya que no existe necesidad de almacenar ningún tipo de información a disco.

Este análisis de los datos en función del dominio y del tipo de aplicación que se quiere generar en base al modelo que se construye abre un nuevo punto de investigación que se incluye como uno de los puntos de trabajo futuro mencionados en el epígrafe correspondiente dentro del [capítulo 10](#) de este documento.

### 9.3 EVALUACIÓN DE LOS PROTOTIPOS PRESENTADOS

El segundo apartado del proceso de evaluación establecido para la propuesta presentada en este trabajo de investigación se centra, como ya se mencionó en la introducción de este capítulo, en la realización de pruebas con los prototipos presentados.

Estas pruebas permitirán conocer el grado de idoneidad de la herramienta BPLLevel Modeler para la construcción de modelos de proceso por parte de expertos con pocos conocimientos técnicos y establecer la reducción de esfuerzo en el proceso de desarrollo que supone la utilización de una herramienta generadora como BPLLevel Generator.

#### 9.3.1 Evaluación realizada sobre BPLLevel Modeler

BPLLevel Modeler es una herramienta de modelado de procesos de negocio construida para que los expertos del dominio puedan construir modelos de proceso de forma sencilla y rápida. Estos expertos del dominio disponen de conocimientos avanzados sobre el dominio del problema que los expertos técnicos no poseen, de la misma forma que los expertos técnicos disponen de conocimientos técnicos de los que los expertos del dominio no disponen.

La evaluación de BPLLevel Modeler está dividida en dos apartados. Por un lado se comprobó la dificultad que los expertos encontraban para acometer unas tareas consideradas como básicas para el modelado de procesos de negocio. Por el otro, se realizó una comparación del prototipo propuesto en este trabajo de investigación y una herramienta de modelado asociada a la propuesta jBPM. Se escogió esta herramienta para la comparación porque jBPM fue la plataforma con soporte para el modelado y la ejecución de procesos que fue considerada como la alternativa más viable, junto con WWF, para la realización del trabajo asociado a esta investigación.

##### 9.3.1.1 Evaluación de la dificultad de las operaciones básicas

Para asegurarse de que los expertos del dominio puedan construir sus modelos de proceso de forma sencilla, BPLLevel Modeler ha sido construido de tal forma que el conjunto de operaciones básicas para el modelado de procesos de negocio puedan ser realizadas de la forma más sencilla posible. El conjunto de operaciones consideradas como básicas para el modelado de procesos de negocio es el siguiente: inicio del modelado, añadir una actividad al modelo, conectar dos actividades existentes en el modelo, cambiar de nombre el modelo de proceso y guardar el modelo.

A la hora de calificar estas operaciones se ha vuelto a utilizar una escala de tipo Likert [Likert 1932], donde los encuestados califican los elementos de la encuesta con valores numéricos en función de su grado de aceptación del elemento. Al igual que en el caso anterior, se ha utilizado una escala de Likert de 5 puntos donde las respuestas posibles al grado de dificultad de estas tareas básicas del modelado son las siguientes: 1-muy fácil, 2-

fácil, 3-normal, 4-difícil y 5-muy difícil. Las calificaciones otorgadas por los expertos a estas operaciones se muestran en la siguiente tabla.

<b>Dificultad de realización de las operaciones básicas de modelado con BPLevel Modeler</b>					
<b>Operación analizada</b>	<b>Nº resp. muy fácil (1)</b>	<b>Nº resp. fácil (2)</b>	<b>Nº resp. normal (3)</b>	<b>Nº resp. difícil (4)</b>	<b>Nº resp. muy difícil (5)</b>
<b>Inicio del modelado</b>	15	6	10	--	--
<b>Añadir una actividad</b>	30	--	--	1	--
<b>Conectar dos actividades</b>	9	11	10	1	--
<b>Cambiar el nombre del modelo</b>	25	5	1	--	--
<b>Guardar el modelo</b>	24	--	7	--	--

Tabla 16: Dificultad de las operaciones básica del modelado

Los datos de la tabla anterior, donde se muestra el número de respuestas de cada grado de la escala Likert que los expertos le dieron a cada una de las operaciones básicas del modelado, se encuentran resumidos en el gráfico de barras que se muestra a continuación. Dicho gráfico representa la dificultad media de cada una de las operaciones en base a las respuestas concedidas por los expertos. Tal y como se observa, la dificultad media de 4 de las operaciones no supera el valor 2, por lo que estas cuatro operaciones son consideradas entre fáciles y muy fáciles por los expertos del dominio.

### Dificultad Media de las Operaciones

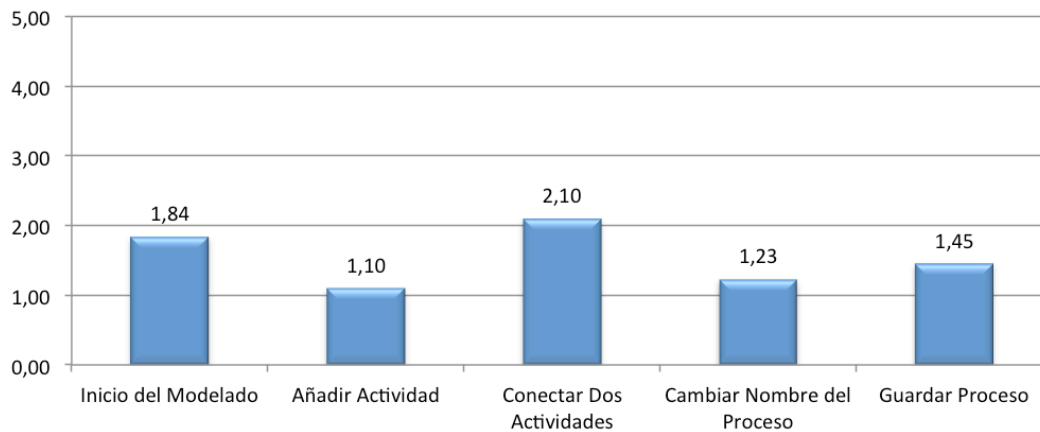


Figura 164: Dificultad media de las operaciones básicas en BPLLevel Modeler

La única operación básica que supera el grado de dificultad fácil, con un valor medio de dificultad de 2.1, es la conexión de dos actividades. Esto es debido a que la herramienta BPLLevel Modeler no dispone de una entidad de conexión de actividades en la zona de entidades si no que permite conectar dos actividades utilizando unas áreas de control que se muestran cuando el experto posiciona el puntero del ratón en el borde de las actividades.

#### 9.3.1.2 Comparación de BPLLevel Modeler con otra herramienta de modelado

La otra prueba realizada con el prototipo BPLLevel Modeler fue la realización de una comparación entre esta herramienta y la herramienta de modelado existente en la propuesta de jBPM, comentada en el [capítulo 4](#) de este documento. Se escogió esta herramienta para la realización de la comparación por las siguientes causas:

- La ausencia de limitaciones funcionales en contrapunto a algunas de las herramientas comerciales mencionadas anteriormente, cuyas versiones de prueba incluyen ciertas limitaciones a la hora de realizar el modelado.
- Las similitudes encontradas en la disposición de los elementos utilizados para el modelado en ambas herramientas. Al igual que en el caso de BPLLevel Modeler, esta herramienta dispone de una zona de entidades situada a la izquierda del interfaz gráfico y un área de modelado similar.
- La utilización de un código de color de similares características al incluido en la notación BPLM, por lo que el cambio de utilización de una herramienta a otra no supone una diferencia drástica a la hora de conocer los elementos que van a ser utilizados.

El proceso de comparación de las herramientas consistió en la construcción del mismo modelo de proceso utilizando ambas alternativas mientras una aplicación de medición de eventos llamada Mousotron **[Mousotron]** monitorizaba los siguientes valores: los metros

recorridos por el ratón del usuario durante el modelado del proceso, el tiempo empleado para completar el modelo de proceso de negocio y el número de eventos de teclado necesarios para cumplimentar el modelo.

Además, se supervisó todo el proceso de pruebas para establecer el número de errores que cometidos por el usuario durante el uso de cada una de las herramientas. Al finalizar la prueba, los usuarios fueron preguntados por la dificultad de uso de las alternativas presentadas en base a la siguiente escala de Likert: 1- muy fácil, 2- fácil, 3- normal, 4- difícil y 5- muy difícil.

Los resultados obtenidos durante la realización de esta comparación de herramientas se muestran en la siguiente tabla, donde se incluyen los valores medios de cada uno de los parámetros monitorizados así como de la dificultad expresada por el usuario a la finalización de la prueba.

Valores medios de los parámetros monitorizados durante la comparativa		
Parámetro	Valor obtenido con BPLLevel Modeler	Valor obtenido con herramienta jBPM
Tiempo empleado en la prueba (minutos)	2,12	3,84
Número de errores cometidos	0,64	2,21
Metros recorridos por el ratón	5,01	8,13
Eventos de teclado recogidos	1,71	132,50
Dificultad según escala de Likert	1,29	3,57

Tabla 17: Valores medios de los parámetros monitorizados en la comparativa

Tal y como se observa en la tabla anterior, la herramienta BPLLevel Modeler permite realizar el modelado de procesos de forma más rápida, con un menor número de eventos de teclado y con una distancia menor del ratón. Además, el número medio de errores cometidos con BPLLevel Modeler es casi 4 veces inferior al asociado a la propuesta jBPM y la dificultad media según la escala de Linkert es más del doble en el caso de la herramienta asociada a jBPM.

Los gráficos que se incluyen a continuación muestra un resumen de estos valores donde es posible observar con mayor claridad las diferencias encontradas entre ambas herramientas.

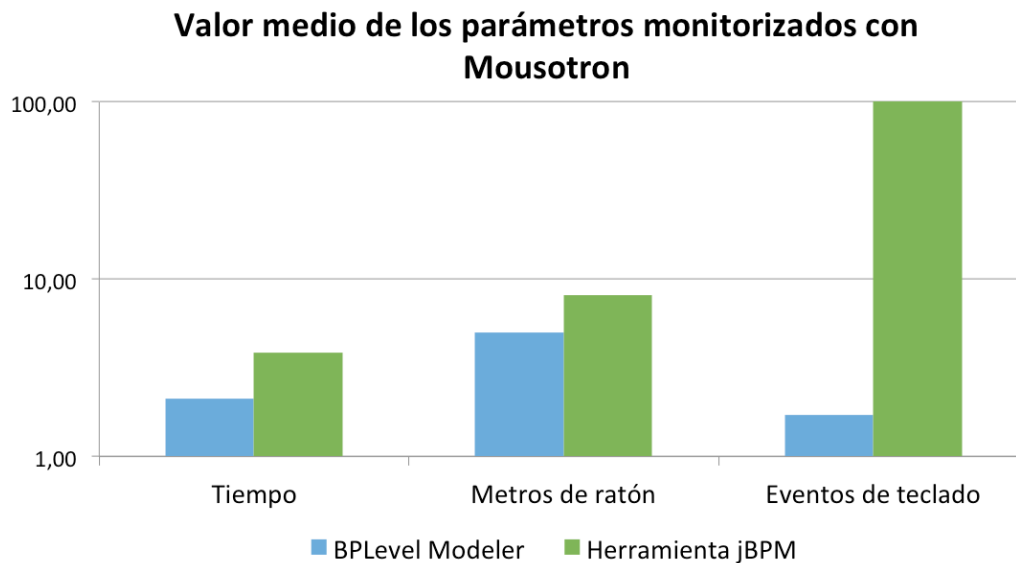


Figura 165: Comparativa de parámetros monitorizados

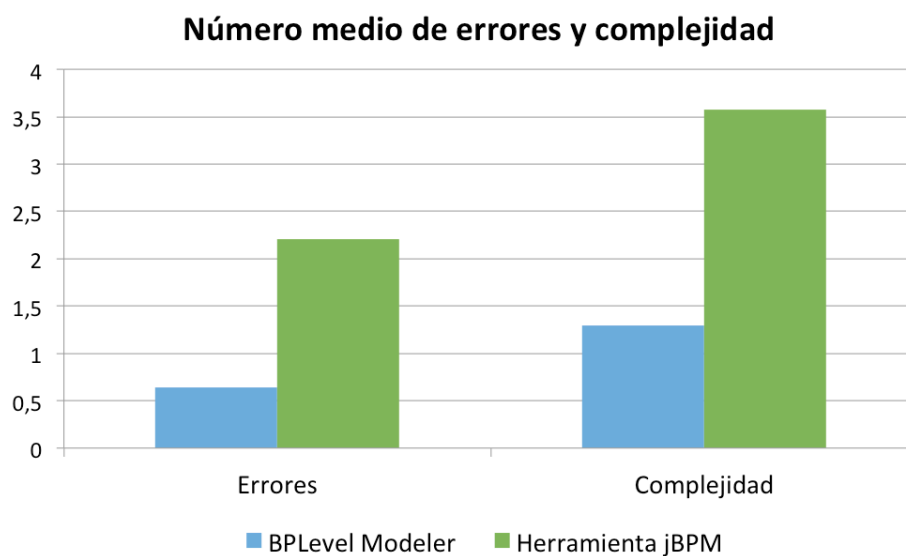


Figura 166: Comparativa de errores y complejidad

Durante la realización de esta comparación entre herramientas de modelado se estableció que sería interesante cuantificar la capacidad de adaptación de los usuarios a la herramienta BPLLevel Modeler. Es por esto que aprovechando el entorno de pruebas existente se decidió que los usuarios repitieran el modelado del mismo proceso con dicha herramienta una vez hubieran concluido el proceso de la prueba comparativa. De esta forma se obtendría una idea aproximada del beneficio obtenido por el uso repetitivo de la herramienta que acompaña a la metodología BPLOM aquí propuesta.

Los parámetros medidos durante esta segunda ronda de modelado con BPLLevel Modeler fueron los mismos a excepción de la dificultad según la escala Likert, obteniéndose los datos que se muestran en la siguiente tabla bajo la columna “Valor obtenido con BPLLevel Modeler (repetición)”.



Repetición del modelado con BPLLevel Modeler		
Parámetro	Valor obtenido con BPLLevel Modeler	Valor obtenido con BPLLevel Modeler (repetición)
Tiempo empleado en la prueba (minutos)	2,12	1,14
Número de errores cometidos	0,64	0,14
Metros recorridos por el ratón	5,01	3,73
Eventos de teclado recogidos	1,71	0,00

Tabla 18: Comparativa de repetición del modelado con BPLLevel Modeler

### Valor medio de los parámetros monitorizados

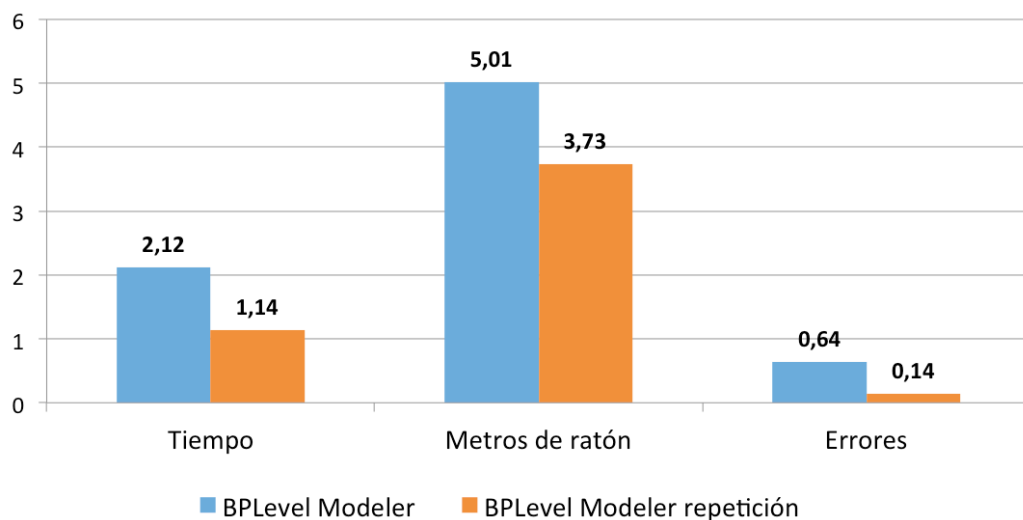


Figura 167: Gráfico comparativo de repetición del modelado con BPLLevel Modeler

Tal y como se puede observar en la figura anterior, donde se incluye un resumen gráfico de los resultados obtenidos al repetir el modelado del proceso con BPLLevel Modeler, existe una reducción del valor medio de los parámetros al repetir el modelado con esta herramienta. El parámetro que mayor reducción experimenta es el de los errores, donde existe una reducción del 80% aproximadamente, mientras que los parámetros del tiempo y de los metros de ratón ven reducidos sus valores medios en unos porcentajes que rondan el 47% y el 25% respectivamente.

### 9.3.2 Evaluación realizada sobre BPLLevel Generator

BPLLevel Generator, la herramienta de generación de aplicaciones multigénero incluida en

la propuesta presentada en este trabajo, permite generar el código de las aplicaciones informáticas en base a los modelos de proceso que describen la funcionalidad de estas.

Para establecer la reducción de esfuerzo que se experimenta al utilizar esta herramienta en vez de optar por un ciclo de desarrollo tradicional, se ha optado por analizar una serie de parámetros tanto en los modelos como en el código y comparar dichos valores en base al peso de importancia que se le otorga a cada uno de los parámetros. Este tipo de análisis ha sido utilizado en otros trabajos del grupo de investigación MDE-RG para medir los beneficios obtenidos de la aplicación de algunas propuestas [Espada et al. 2013]. A partir de los pesos que se otorgan a cada parámetro se puede realizar un cálculo del coste de desarrollo de la siguiente forma: se multiplica el valor del parámetro en cada caso por el peso que se le otorga y después se realiza el sumatorio de todos los valores obtenidos con las multiplicaciones.

Las tablas que se incluyen a continuación muestran los parámetros que serán analizados para establecer el beneficio obtenido a partir del proceso de generación de código que se propone en este trabajo.

Parámetros analizados en los modelos de proceso		
Letra identificativa	Nombre del parámetro	Peso de importancia
E	Número de entidades que tiene el modelo de proceso	8
R	Número de relaciones existentes entre las entidades del modelo	10
M	Número de metros recorridos con el ratón para construir el modelo	3
K	Número de clics de ratón necesarios para construir el modelo	3
O	Entidades del modelo que necesitan configuración por parte del usuario	20
D	Número de entidades distintas existentes en el modelo	20
I	Entidades del modelo que necesitan una implementación asociada	<i>valor variable</i> <sup>1</sup>

Tabla 19: Parámetros analizados en los modelos de proceso

<sup>1</sup> El valor del peso asociado a las entidades del modelo que necesitan una implementación asociada depende del tipo de aplicación que se vaya a generar a partir de dicho modelo.

Parámetros analizados en el código		
Letra identificativa	Nombre del parámetro	Peso de importancia
L	Líneas de código	4
W	Número de palabras	2
C	Número de caracteres	0,5
G	Número de interfaces gráficos necesarios	8

Tabla 20: Parámetros analizados en el código

El cálculo del peso asociado al parámetro **I**, que es un número que varía según el tipo de aplicación que se vaya a construir, se realiza de la siguiente forma. Para cada una de las entidades del modelo que necesitan una implementación asociada, se obtienen los valores relacionados con cada uno de los parámetros a analizar en el código (**L**, **W**, **C** y **G**). Una vez obtenidos estos valores se realiza la media geométrica para los valores de las entidades correspondientes a cada uno de los parámetros a analizar. Finalmente, se realiza el sumatorio de las medias geométricas obtenidas anteriormente y se multiplica el número obtenido en el sumatorio por un factor corrector de 0,8. De esta forma se obtiene un peso para el parámetro **I** que depende de dos factores: la complejidad de implementación de las entidades asociadas a cada modelo concreto y la plataforma para la que se esté creando la aplicación.

Una vez que se haya calculado el peso asociado al parámetro **I** según el tipo de aplicación que se va a generar, es posible calcular el coste medio de desarrollo de dicha aplicación. Así, el cálculo del coste medio se realizará de la siguiente forma:

- En el caso de que la alternativa de desarrollo utilice modelos para la construcción de la aplicación correspondiente, el coste medio de desarrollo será el sumatorio obtenido a partir de la multiplicación de los valores obtenidos para los parámetros a analizar en los modelos (**E,R,M,K,O,D,I**) por los pesos de cada parámetro.
- En el caso de que la alternativa de desarrollo no implique la utilización de modelos, el coste medio de desarrollo será el sumatorio obtenido a partir de la multiplicación de los valores obtenidos para los parámetros a analizar en el código (**L,W,C,G**) por los pesos de cada parámetro.

### 9.3.2.1 Coste medio de la construcción de una aplicación web

Utilizando los parámetros incluidos en estas tablas se realizará la estimación del esfuerzo necesario para construir una aplicación web con soporte para el proceso de inserción de incidencias descrito en el apartado correspondiente del [capítulo 7](#). Esta estimación se realizará comparando las siguientes alternativas:

- BPMN MUSIM, el nivel inicial de la metodología. En este escenario los expertos utilizan BPLLevel Modeler para la construcción de los modelos y disponen de actividades preconfiguradas para el proceso de negocio que necesitan construir.
- jBPM con asociación, donde se usa jBPM para la creación de los modelos y se cuenta con un conjunto de actividades preconfiguradas para el proceso a modelar.
- jBPM con implementación, escenario en el que se usa jBPM para el modelado de los procesos pero sin existir actividades preconfiguradas específicas para el modelo de proceso. Así, la implementación de las actividades específicas del modelo ha de realizarse al mismo tiempo que se modela.
- Desarrollo tradicional de una aplicación web con C#, sin la utilización de modelos de proceso.

En este escenario los valores de cada uno de los parámetros a analizar en los modelos son los siguientes:

Valores de los parámetros para los modelos de proceso			
Letra identificativa del parámetro	1- BPMN MUSIM	2- jBPM con asociación	3- jBPM con implementación
E	9	9	9
R	9	9	9
M	3,9	3,6	3,6
K	24	50	68
O	0	3	3
D	4	5	5
I	0	0	3

Tabla 21: Valores de los parámetros para los modelos de proceso de inserción de incidencias (I)

A la hora de construir una aplicación web para que ofrezca soporte al proceso de negocio de inserción de incidencias, se utilizan 3 clases de código personalizadas en la aplicación que representan a las pantallas necesarias para el funcionamiento de la misma: la pantalla de login, la pantalla que muestra las incidencias del usuario que trabaja con la aplicación y la pantalla que permite añadir una incidencia. Los parámetros de código a analizar para este caso concreto son los siguientes:

Valores de los parámetros para el código					
Letra identificativa	Login	Incidencias usuario	Nueva incidencia	Media geométrica	Valor total del parámetro
L	69	184	102	108,9989619	355
W	110	395	212	209,6244232	717
C	1784	6592	3262	3372,603879	11638
G	1	1	1	---	3

Tabla 22: Valores de los parámetros para el código del proceso de inserción de incidencias (I)

Teniendo en cuenta los valores totales de los parámetros de código incluidos en la tabla anterior, se hace un cálculo para extraer el peso asociado al parámetro I en este caso concreto. Dicho cálculo, tal y como se explicó anteriormente, se basa en multiplicar los valores de la columna “Media geométrica” de la tabla anterior por el peso asociado al parámetro, realizar el sumatorio de los valores resultantes de esta multiplicación y multiplicar el valor del sumatorio por 0,8. Así, el valor para peso de importancia del parámetro I en este caso es de 2033,237307.

Una vez que se ha obtenido el peso para el parámetro I es posible calcular el valor medio del coste resultante de la aplicación de cada una de las alternativas para la construcción de la aplicación web que ofrece soporte al modelo de proceso de inserción de incidencias. Los valores numéricos obtenidos del cálculo de dicho coste se observan en el gráfico que se incluye a continuación.

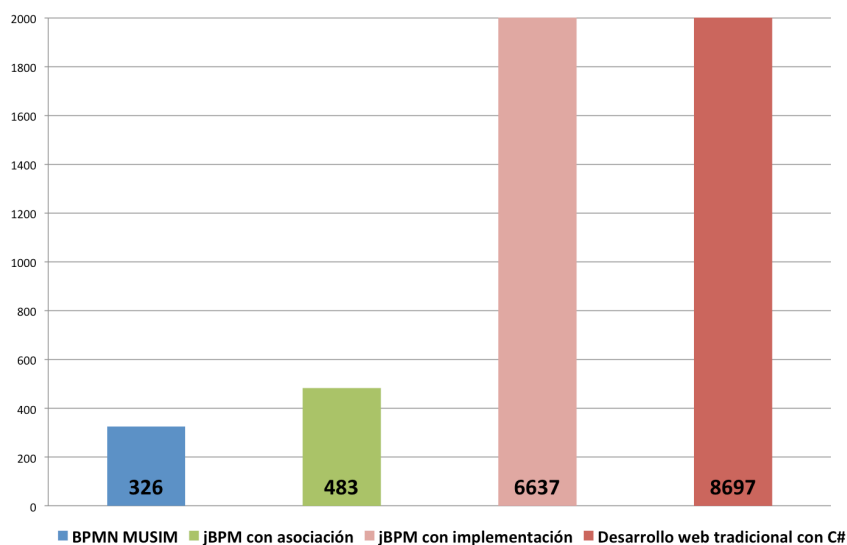


Figura 168: Coste medio del desarrollo de la aplicación web de inserción de incidencias (I)

La gráfica anterior muestra como el coste del desarrollo utilizando el nivel inicial de la metodología propuesta es de 326, siendo el valor más cercano a este el coste del desarrollo con la alternativa jBPM con asociación, que obtiene una puntuación de 483. A igual número de entidades y relaciones entre estas en el modelo, el nivel inicial de la metodología BPLOM se beneficia de la reducción en los valores de los parámetros **M** (metros de ratón), **K** (clics de ratón), **O** (entidades que necesitan configuración) y **D** (número de entidades distintas en el modelo). El coste de la alternativa número 3, jBPM con implementación, se ve marcado por la necesidad de implementación que tienen las entidades propias del proceso (login, incidencias de usuario y añadir incidencia), lo que hace que su valor se acerque más al del desarrollo tradicional de aplicaciones web con C# que al de las otras alternativas.

Si se introdujeran todas las entidades disponibles en la metodología BPLOM para realizar el modelado del mismo proceso de inserción de incidencias, los valores de los parámetros a analizar en los modelos se verían alterados, ya que el número de entidades distintas en el modelo y el número de entidades y de relaciones existentes en el mismo se verían incrementados. La siguiente tabla muestra los valores de dichos parámetros al utilizar todos los niveles de la metodología BPLOM.

Valores de los parámetros para los modelos de proceso			
Letra identificativa del parámetro	1- BPLOM	2- jBPM con asociación	3- jBPM con implementación
E	15	15	15
R	14	14	14
M	3,9	3,6	3,6
K	24	50	68
O	0	3	3
D	10	11	11
I	0	0	3

Tabla 23: Valores de los parámetros para los modelos de proceso de inserción de incidencias (II)

Este nuevo escenario no supone un cambio en los valores de código a analizar, ya que siguen siendo necesarias el mismo número de clases personalizadas para el funcionamiento de la aplicación web asociada al proceso. Es por esto que el valor del parámetro I no sufriría variación alguna, ya que este se calcula en base a los valores numéricos de los parámetros del código. Con este escenario, el coste del desarrollo de la aplicación web con soporte para el proceso de inserción de incidencias según cada una de las alternativas planteadas es el que se muestra en el gráfico incluido a continuación.

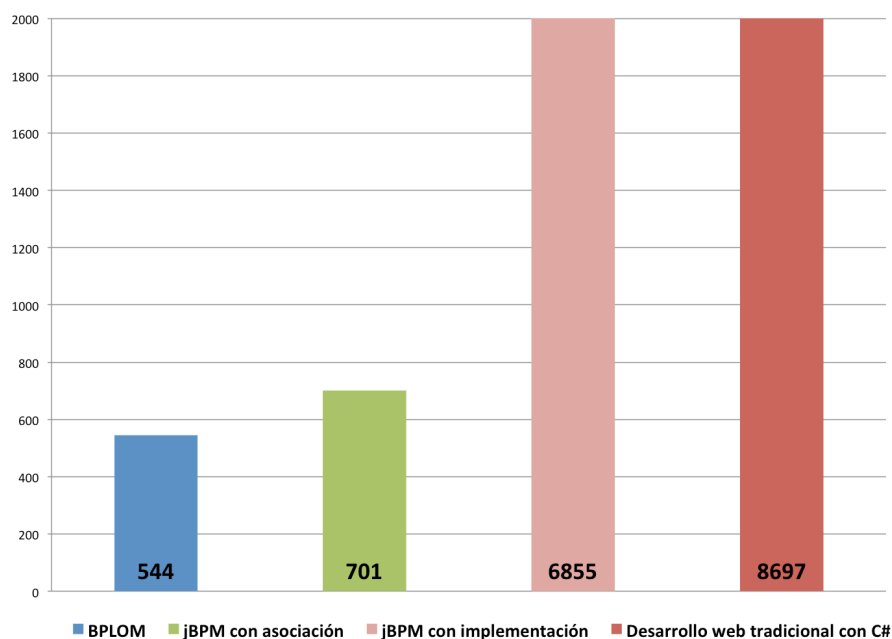


Figura 169: Coste medio del desarrollo de la aplicación web de inserción de incidencias (II)

Tal y como se observa en la ilustración anterior, el coste medio del desarrollo se incrementa para las 3 alternativas relacionadas con el modelado de procesos de negocio, al aumentar el valor de alguno de los parámetros medidos. Comparando los valores obtenidos en este escenario con los obtenidos en el anterior, se observa que el coste medio del desarrollo con BPLOM es un 66% mayor que el coste medio obtenido con BPMN MUSIM, el nivel inicial de la metodología. De igual forma, el uso de jBPM con asociación supone un aumento del 45% en el coste medio del desarrollo y el coste medio resultante del uso de jBPM con implementación experimenta una subida del 3%.

A pesar de la diferencia existente en el porcentaje de aumento del coste medio del desarrollo en las dos primeras alternativas, BPLOM todavía muestra un coste medio menor al experimentado con la alternativa jBPM con asociación, con unos valores de 544 y 701 respectivamente. Estos valores siguen siendo muy inferiores a los relacionados con las alternativas jBPM con implementación y desarrollo web tradicional con C#, donde la necesidad de aportar una implementación para las actividades propias del proceso de negocio supone una penalización demasiado elevada.

Repitiendo el análisis anterior para el proceso de negocio de resolución de incidencias, que es el proceso que completa el sistema de gestión de incidencias descrito en el [capítulo 7](#) de esta documentación, se observa un patrón similar. Como el modelo relacionado con este proceso es más simple, al disponer de menos entidades y poder modelarse de forma más sencilla con BPLLevel Modeler, el coste medio de desarrollo también es menor. Teniendo en cuenta los valores de todos los parámetros de código analizados para este nuevo escenario de pruebas, el valor del parámetro I en este caso sería de 1638,978207.

Las gráficas que se incluyen a continuación muestran el coste medio de desarrollo de la aplicación web correspondiente a este modelo usando en primer lugar BPMN MUSIM y después todos los niveles de la metodología.

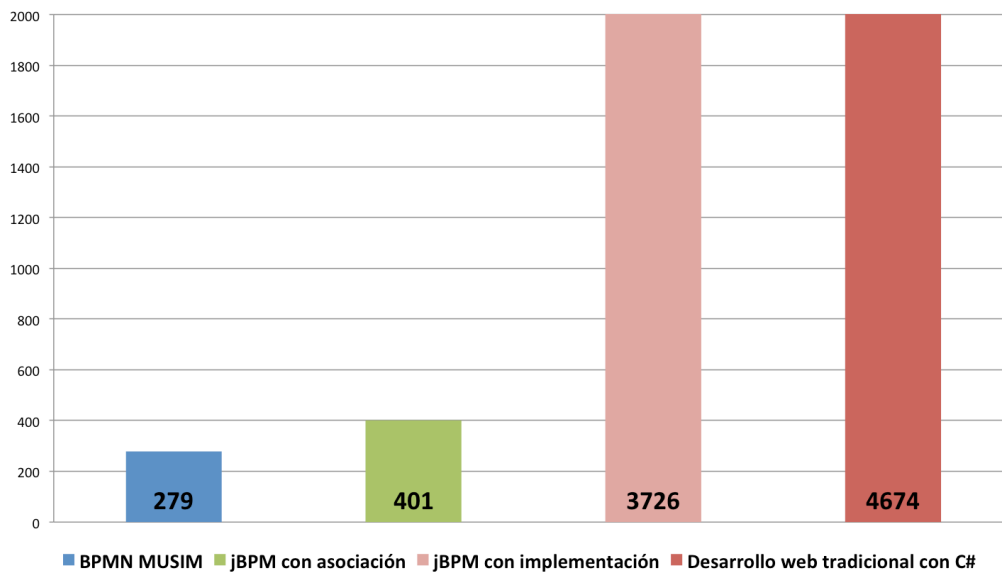


Figura 170: Coste medio del desarrollo de la aplicación web de resolución de incidencias (I)

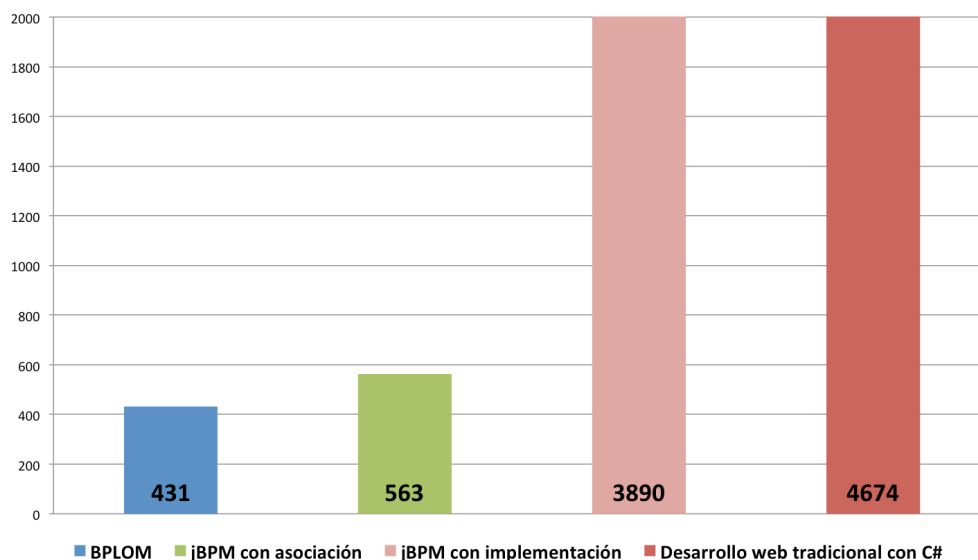


Figura 171: Coste medio del desarrollo de la aplicación web de resolución de incidencias (II)

De manera análoga a lo sucedido en el escenario anterior, se observa que el coste asociado a las dos primeras alternativas es muy inferior al de las alternativas que necesitan aportar una implementación. También se puede comprobar que el coste medio resultante del uso de BPMN MUSIM (279) es algo inferior al obtenido con el uso de todos los niveles de la metodología (431), existiendo un incremento en el coste medio de un 54% aproximadamente. El incremento del coste medio para la alternativa de jBPM con asociación es de un 40%, valor cercano al incremento visto en el caso del proceso de inserción de incidencias.



### 9.3.2.2 Coste medio de la construcción de un videojuego

A través de la medición de los parámetros anteriores se realizará el cálculo para establecer el coste medio de desarrollo de un videojuego para dispositivos móviles. A pesar de que la combinación del editor GADE4ALL con los prototipos que acompañan a este trabajo de investigación permite generar videojuegos para múltiples plataformas en un solo intento, la comparativa se realizará para una única plataforma: la plataforma iOS.

En este caso se han escogido, a modo de ejemplo, dos tipos de juegos para realizar los cálculos del coste medio de desarrollo: el juego de plataformas y el juego de tipo *touch*. Las alternativas de desarrollo con las que se realizará el cálculo del coste medio son:

- 1- BPMN MUSIM, el nivel inicial de la metodología. En este escenario los expertos utilizan BPLevel Modeler para la construcción de los modelos y disponen de actividades preconfiguradas para el proceso de negocio que necesitan construir.
- 2- jBPM con asociación, donde se usa jBPM para la creación de los modelos y se cuenta con un conjunto de actividades preconfiguradas para el proceso a modelar.
- 3- jBPM con implementación, escenario en el que se usa jBPM para el modelado de los procesos pero sin existir actividades preconfiguradas específicas para el modelo de proceso. Así, la implementación de las actividades específicas del modelo ha de realizarse al mismo tiempo que se modela.
- 4- Desarrollo tradicional de un videojuego de plataformas con Objective C, sin la utilización de modelos de proceso.

Las tablas incluidas a continuación muestran los valores de los parámetros a medir en el código y en los modelos para el juego de plataformas para las alternativas anteriores.

Valores de los parámetros para los modelos de proceso			
Letra identificativa del parámetro	1- BPMN MUSIM	2- jBPM con asociación	3- jBPM con implementación
E	12	12	12
R	15	15	15
M	4,1	4,3	4,3
K	33	60	72
O	0	6	6
D	4	5	5
I	0	0	6

Tabla 24: Valores de los parámetros del modelo de proceso para el videojuego de plataformas (I)

Valores de los parámetros para el código								
Letra identificativa	Cargar elementos	Mover personaje	Coger elemento	Realizar disparo	Simular turno máquina	Refrescar interfaz	Media geométrica	Valor total
L	63	73	24	8	33	93	37,33879123	294
W	212	108	51	36	139	323	111,1667569	869
C	1980	1989	636	419	1776	3392	1359,810278	10192
G	---	---	---	---	---	---	---	1

Tabla 25: Valores de los parámetros de código para el videojuego de plataformas

En el caso anterior, donde se medían los datos asociados a la construcción de aplicaciones web, cada actividad personalizada que se incluía en el modelo disponía de un interfaz gráfico y una clase de código asociadas. Sin embargo, en el caso de uso de los videojuegos, las actividades se corresponden con llamadas a métodos de código existentes, por lo que los valores de los parámetros serán los de los métodos y solamente existirá un único interfaz gráfico: la pantalla del juego; de ahí que el valor del parámetro **G** esté vacío en todos los casos salvo en la columna del valor total, donde se refleja que existe un único interfaz gráfico asociado. Tomando todos los datos anteriores, el valor numérico del parámetro I para el videojuego de plataformas es: 841,2750543.

Una vez obtenido el valor del parámetro I, es posible calcular el coste medio del desarrollo de un videojuego de plataformas para iOS utilizando las distintas alternativas señaladas. Los resultados de dicho cálculo se muestran en la siguiente gráfica.

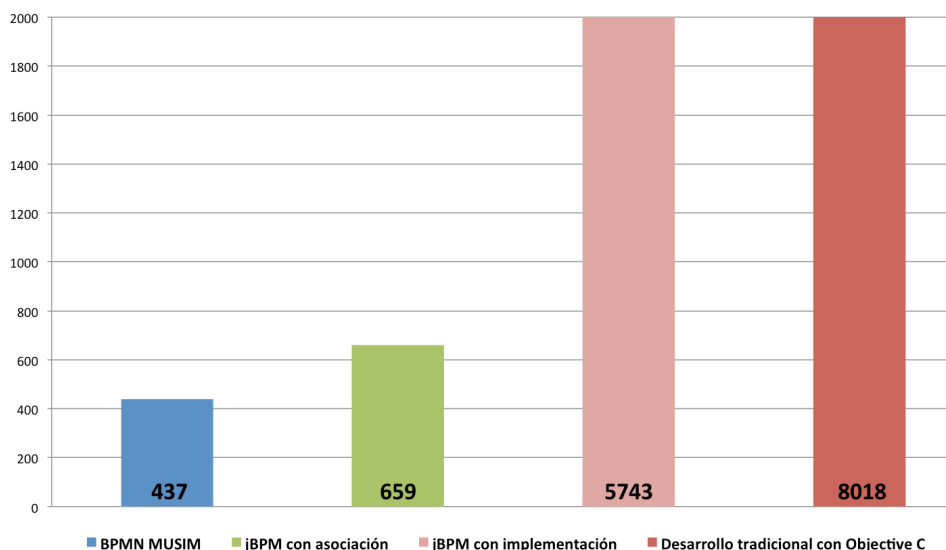


Figura 172: Coste medio del desarrollo de un videojuego de plataformas para iOS (I)

Tal y como se observa en la ilustración anterior, el coste medio del desarrollo de un videojuego de plataformas para iOS es de 437 utilizando el nivel de inicial de la metodología frente al valor de 659 obtenido con la alternativa de jBPM con asociación, la

alternativa más cercana. Esto supone que el coste medio de desarrollo de este tipo de videojuegos utilizando BPMN MUSIM es alrededor de un 40% menos costoso que utilizando jBPM con asociación. Al igual que ocurriera en el caso de las aplicaciones web, las alternativas jBPM con implementación y desarrollo tradicional obtienen valores de coste medio de desarrollo mucho más elevados, en este caso de 5743 y 8018 respectivamente.

En caso de utilizar toda la potencia expresiva de la metodología BPLOM para modelar la lógica del videojuego con los 5 niveles, las cifras relativas a los parámetros a analizar en los modelos del proceso asociado al videojuego serían las mostradas en la tabla incluida a continuación.

Valores de los parámetros para los modelos de proceso			
Letra identificativa del parámetro	1- BPLOM	2- jBPM con asociación	3- jBPM con implementación
E	22	22	22
R	24	24	24
M	4,5	4,9	4,9
K	33	60	72
O	0	6	6
D	13	13	13
I	0	0	6

Tabla 26: Valores de los parámetros del modelo de proceso para el videojuego de plataformas (II)

Con los parámetros mostrados en la tabla anterior, el cálculo del coste medio arroja los resultados que se pueden ver en la siguiente figura. El coste obtenido al utilizar la propuesta pasa a ser de 609, lo que supone un aumento del coste de un 39% con respecto al coste experimentado al utilizar solamente el nivel inicial de la metodología. Este aumento vuelve a repetirse en el caso de jBPM con asociación; esta alternativa obtiene un coste medio de 831, lo que supone un incremento del 26%.

El incremento porcentual que experimenta el coste medio del desarrollo para las dos primeras alternativas mantiene la tendencia vista en el caso de la construcción de la aplicación web, si bien los porcentajes son algo mayores en el caso de la construcción de los videojuegos.

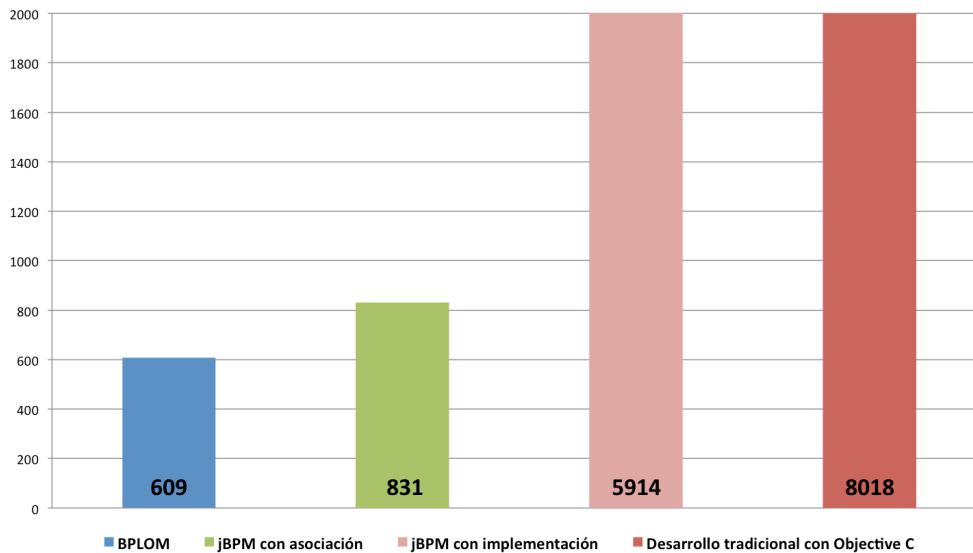


Figura 173: Coste medio del desarrollo de un videojuego de plataformas para iOS (II)

Una vez calculado el coste medio de desarrollo de un juego de plataformas, considerado el más complejo, se va a analizar el coste medio de desarrollo del tipo de juego más sencillo: el juego de tipo touch. Las tablas que se incluyen a continuación muestran los valores de los parámetros para el código y los modelos de proceso relativos a este tipo de videojuego.

Valores de los parámetros para los modelos de proceso			
Letra identificativa del parámetro	1- BPMN MUSIM	2- jBPM con asociación	3- jBPM con implementación
E	9	9	9
R	11	11	11
M	3,1	3,4	3,4
K	28	35	35
O	0	4	4
D	4	5	5
I	0	0	4

Tabla 27: Valores de los parámetros del modelo de proceso para el videojuego touch (I)

Valores de los parámetros para el código						
Letra identificativa	Cargar elementos	Mover personaje	Coger elemento	Refrescar interfaz	Media geométrica	Valor total del parámetro
L	40	65	6	28	25,70814007	139
W	102	135	12	49	53,34304666	298
C	1121	2122	173	678	726,7862416	4094
G	---	---	---	---	---	1

Tabla 28: Valores de los parámetros de código para el videojuego touch

Teniendo en cuenta los valores de la tabla anterior, el peso asociado al parámetro I para los videojuegos de tipo touch es de 458,3294195. Utilizando dicho valor es posible calcular el coste medio asociado al desarrollo de un videojuego de este tipo para la plataforma iOS, obteniendo los valores que se muestran en la gráfica incluida a continuación.

El coste medio del desarrollo de un juego de este tipo utilizando BPMN MUSIM, el nivel inicial de la metodología propuesta es de 355 mientras que el de la alternativa jBPM con asociación es de 477, siendo esta la alternativa más cercana. El coste medio usando jBPM con asociación asciende a 2310 y el del desarrollo tradicional con el lenguaje Objective C se queda en 3207.

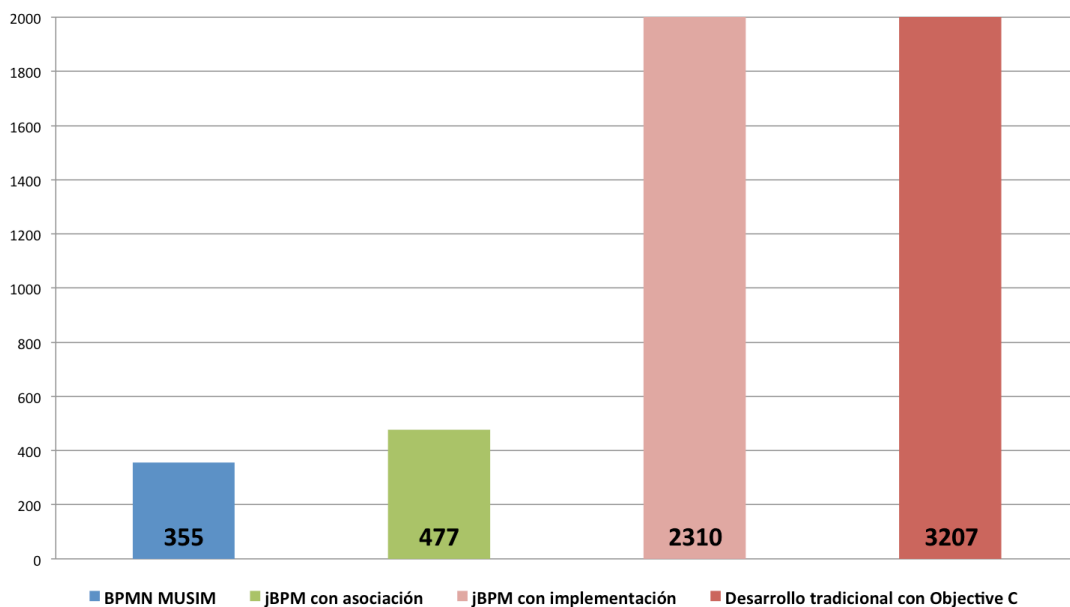


Figura 174: Coste medio del desarrollo de un videojuego touch para iOS (I)

Al utilizar todos los niveles de la metodología BPLOM varían los valores de los parámetros que se analizan en los modelos, ya que el número de entidades utilizadas es mayor. Los valores de dichos parámetros para el caso del videojuego touch son los que están recogidos en la siguiente tabla.

Valores de los parámetros para los modelos de proceso			
Letra identificativa del parámetro	1- BPLOM	2- jBPM con asociación	3- jBPM con implementación
E	13	13	13
R	14	14	14
M	3,1	3,4	3,4
K	28	35	35
O	0	4	4
D	12	12	12
I	0	0	4

Tabla 29: Valores de los parámetros del modelo de proceso para el videojuego touch (II)

A partir de estos nuevos valores, el coste medio de desarrollo del videojuego touch sufre las modificaciones que pueden verse en el gráfico que se muestra a continuación. El coste medio al utilizar la metodología BPLOM es de 577, alrededor de un 62% mayor que al utilizar solamente el nivel inicial de la metodología. El coste medio asociado a la alternativa jBPM con implementación asciende a 679, con un incremento del 42% aproximadamente. A pesar del aumento en el coste de ambas alternativas, éstas siguen siendo las que obtienen un coste medio mucho menor que las otras alternativas.

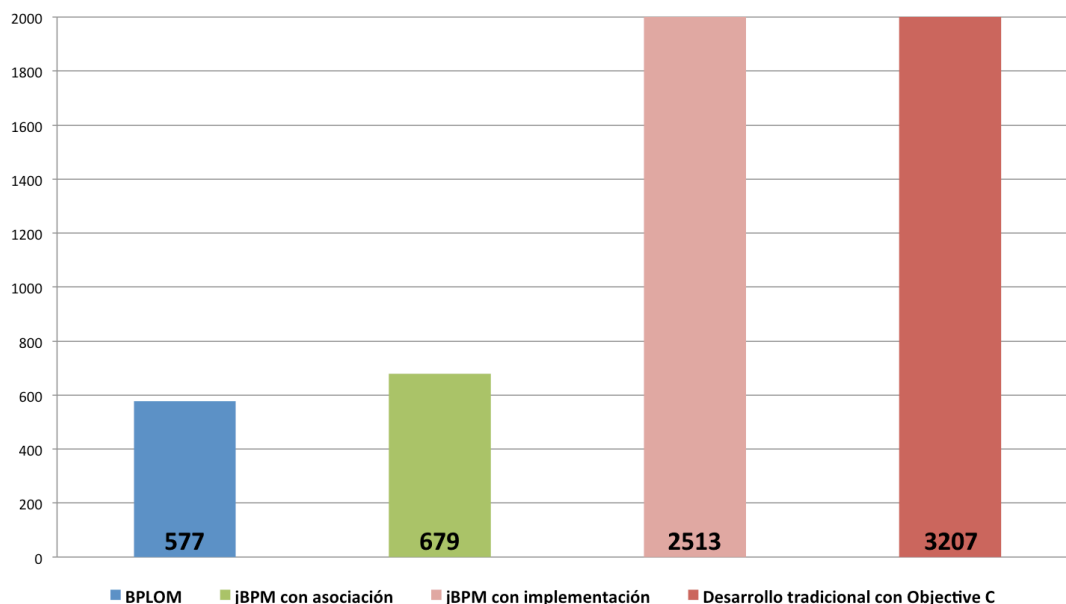


Figura 175: Coste medio del desarrollo de un videojuego touch para iOS (II)

## 9.4 CONCLUSIONES

El apartado de evaluación de la propuesta realizada en este trabajo de investigación está diseñado para mostrar varias de las características de la misma.

En primer lugar se realizaron varios test de usabilidad para conocer la idoneidad del nivel inicial de la metodología, para comprobar que se ajustaba a las necesidades de los expertos del dominio que no disponen de conocimientos técnicos suficientes para utilizar una notación o lenguaje de modelado ya existente. Los resultados revelaron que los expertos consideraron que no necesitaban ningún elemento más para modelar sus procesos de forma completa y además indicaron que comprendían mejor los símbolos de BPMN MUSIM que los de BPMN.

A partir de los resultados obtenidos en dichas pruebas se expandió el test para conocer la opinión de los usuarios acerca del incremento de complejidad experimentado con cada nivel de la metodología así como para conocer aquellas entidades que consideraban menos importantes. Los niveles de la metodología BPLOM parecen ser adecuados, tanto en ordenación como en diferencias de complejidad entre ellos, para la mayor parte de los usuarios que realizaron las pruebas. Además, las tasas de importancia revelaron que algunas de las entidades incluidas en la metodología BPLOM son más relevantes según el dominio al que corresponda el experto de negocio, por lo que se abre una nueva vía de investigación a tener en cuenta en el trabajo futuro (ver el punto primero del trabajo futuro en el [epígrafe 10.6](#)).

La segunda parte del proceso de evaluación realizado se centró en la realización de pruebas sobre las herramientas propuestas. BPLLevel Modeler demostró ofrecer un interfaz sencillo de utilizar para los expertos que realizaron las pruebas y además en la comparación realizada con otra herramienta de modelado de procesos de negocio basada en la plataforma Eclipse obtuvo mejores resultados, en base a los parámetros medidos, a la hora de modelar el mismo proceso. Por su parte, BPLLevel Generator fue utilizada para comparar el coste medio de desarrollo de diversas aplicaciones a partir de una serie de parámetros medidos tanto en los modelos utilizados como en el código de las aplicaciones. Los resultados obtenidos permiten establecer que el coste medio de desarrollo de una aplicación es bastante inferior cuando se utiliza un enfoque basado en la utilización de modelos, y en este caso el coste asociado al uso de la propuesta realizada en esta investigación es menor al coste resultante de la aplicación de la propuesta de jBPM.





---

---

## PARTE V

## CONCLUSIONES

---

---



# CAPÍTULO 10

## CONCLUSIONES Y TRABAJO FUTURO

---

---

### 10.1 SÍNTESIS DEL TRABAJO DESARROLLADO

Esta tesis propone una metodología de modelado de procesos de negocio por niveles, promoviendo la adaptación del conjunto de símbolos o entidades disponibles para la construcción de los modelos de proceso al nivel de conocimiento técnico y de modelado del que dispone el experto del dominio que la utiliza. Además, los modelos de proceso construidos con la herramienta de modelado que acompaña a la metodología, BPLLevel Modeler, pueden ser analizados y traducidos de forma automática por la herramienta generadora de artefactos BPLLevel Generator para construir una aplicación informática que ofrezca soporte al modelo de proceso construido por el experto del dominio.

El contenido de los [capítulos 3 y 4](#) es un recorrido por algunas de las notaciones de modelado existentes en la actualidad junto con herramientas y plataformas que ofrecen soporte para ellas. Se realiza este resumen para mostrar las características de todas ellas a modo de establecer los puntos interesantes que deberían ser cubiertos por la metodología de modelado propuesta en este trabajo. El [capítulo 5](#) es el encargado de analizar la iniciativa MDE, que propone la utilización de modelos para la solución de algunos de las limitaciones y problemas comunes que acompañan al desarrollo de software tradicional.

La metodología por niveles que compone la parte fundamental de este trabajo de investigación, mostrada en el [capítulo 6](#), parte de un primer acercamiento al modelado de procesos de negocio por parte de expertos del dominio sin conocimientos técnicos. Este primer paso en la definición de la metodología, que posteriormente se convertiría en el nivel inicial de la misma, fue concebido como una notación de modelado de procesos de negocio muy simplificada, que incluyera solamente aquellas entidades estrictamente necesarias para acometer, de forma completa, el modelado de procesos de negocio de distintos dominios sin que los expertos del dominio del problema tuvieran que disponer de conocimientos técnicos. Este acercamiento fue denominado BPMN MUSIM: BPMN MUy SIMplificada.

A pesar de que esta notación de procesos de negocio muy simplificada era capaz de modelar los procesos de negocio de distintos ámbitos o dominios de forma completa, tenía una gran desventaja con respecto al resto de notaciones de modelado existentes: una capacidad expresiva muy limitada. Es por esto que se decidió proponer la construcción de una metodología por niveles, aumentando la expresividad de la notación con el paso de cada nivel hasta llegar a una capacidad expresiva parecida a la ofrecida por las notaciones de modelado de procesos existentes. BPMN MUSIM se convirtió en este momento en el nivel más básico de la metodología de modelado BPLM, un nivel de modelado básico destinado a los expertos del dominio con conocimientos técnicos limitados y con poca

experiencia en el modelado de procesos de negocio. Los otros 4 niveles de la metodología constituyen extensiones incrementales de este nivel inicial, introduciendo conceptos de modelado más avanzado: decisiones complejas, eventos, actividades de diversos tipos y entidades para el manejo de datos. Los niveles fueron dispuestos de tal forma que el nivel de conocimiento técnico y de modelado fuera creciendo de forma gradual hasta el último nivel, donde los modelos están compuestos por detalles avanzados de carácter técnico o informático. Este contenido está incluido en el [capítulo 6](#) de esta memoria.

El [capítulo 7](#) incluye los ámbitos de aplicación de la metodología que fueron estudiados durante este trabajo de investigación. Para demostrar la capacidad de adaptación de la metodología BPLOM a diversos dominios y a procesos de negocio con diversos niveles de complejidad se seleccionaron los siguientes procesos para el caso de estudio: procesos de negocio tradicionales como la gestión de incidencias y la contratación de personal, procesos que describen la lógica que describe el funcionamiento de un videojuego y el proceso de utilización de una aplicación de catálogos. Todos estos procesos fueron modelados utilizando todos los niveles de la metodología, mostrando como afectaba la aplicación de cada nuevo nivel a la estructura del modelo de proceso. La descripción de los prototipos que permiten utilizar la metodología para construir los modelos y las aplicaciones que ofrecen soporte a estos se realiza en el [capítulo 8](#). El contenido de este capítulo incluye tanto un desglose de la funcionalidad de las herramienta BPLLevel Modeler y BPLLevel Generator como ejemplos de la misma a la hora de generar modelos y aplicaciones para dos de los casos incluidos en el caso de estudio.

El [capítulo 9](#) sirve para mostrar el marco de evaluación escogido para medir diversos valores relacionados tanto con la metodología BPLOM como con las herramientas. Así, se muestran los resultados obtenidos en test de usabilidad para medir la facilidad de comprensión de las representaciones gráficas de las entidades incluidas en la metodología. Además, también se incluyen resultados relativos a la usabilidad de la herramienta BPLLevel Modeler, mostrando la sencillez de realización de las tareas consideradas como básicas para el modelado de procesos de negocio y comparando su rendimiento a la hora de construir un modelo con una herramienta de modelado similar. Por último, las pruebas realizadas con la herramienta generadora BPLLevel Generator permiten obtener una estimación del coste medio de creación de una aplicación utilizando diversas alternativas que van desde el uso de la metodología BPLOM al seguimiento de un ciclo de desarrollo tradicional.

## 10.2 VERIFICACIÓN Y EVALUACIÓN DE LOS OBJETIVOS

En el [capítulo 1](#) de este documento se plantearon una serie de objetivos específicos que han sido cubiertos por el trabajo realizado en esta investigación. El cumplimiento de estos objetivos específicos ha significado la consecución de los objetivos generales de la investigación.

A continuación se incluye una lista con los objetivos mencionados anteriormente y como se ha llegado a su cumplimiento.

- **Realizar un análisis global de los lenguajes y notaciones de modelado de procesos de negocio existentes en la actualidad, estudiando sus características e identificando sus puntos fuertes y débiles.** Durante este análisis, mostrado en el [capítulo 3](#), se realizó un recorrido por las notaciones y lenguajes de modelados más relevantes que existen, desde la notación estándar de modelado de procesos de negocio, denominada BPMN, hasta otras alternativas como las redes de Petri o los diagramas de actividad de UML. Al realizar este análisis se identificaron tanto las ventajas como los inconvenientes de las notaciones y los lenguajes a fin de establecer el conjunto de características deseables para la definición de la metodología.
- **Realizar un estudio de las plataformas y aplicaciones con soporte para la utilización de modelos de proceso, haciendo distinción entre aquellas que solamente permiten modelar los procesos y otras que permiten modelar y ejecutar los procesos.** A través del estudio de algunas de las aplicaciones y plataformas con soporte para el modelado de procesos de negocio y/o la ejecución de modelos de proceso se estableció el conjunto de necesidades que era preciso cubrir a la hora de ofrecer una aplicación de modelado de procesos de negocio para el uso de la metodología a definir.
- **Realizar un recorrido por los géneros de aplicaciones más utilizados en la actualidad, intentando establecer cuáles serían apropiados para la introducción de la propuesta y si en estos géneros existe alguna solución actual similar a la que se propone.** En el segundo apartado del [capítulo 7](#) se realizó un pequeño resumen de los ámbitos en los que se podría aplicar la metodología de modelado por niveles. Este resumen incluye la identificación de algunos de los géneros de aplicaciones más utilizados en la actualidad (aplicaciones web y videojuegos y aplicaciones de catálogo para dispositivos móviles) así como la mención de algunas herramientas comerciales existentes en la actualidad con funcionalidades similares a las que se puede obtener al utilizar la metodología BPLOM y las herramientas propuestas en este trabajo de investigación.
- **Establecer los niveles de la metodología que permitan la adaptación de ésta al nivel de conocimiento del experto del dominio.** En el [capítulo 6](#) se presenta la metodología de modelado por niveles para procesos de negocio llamada BPLOM. Para realizar esta presentación se realiza un recorrido por todos los niveles de la

metodología para mostrar las entidades que ofrece cada uno y su representación gráfica. También se incluye un razonamiento sobre aquellas entidades de otras notaciones y lenguajes de modelado que fueron descartadas para esta primera versión de la metodología, quedando abierta la posibilidad de aumentar el conjunto de entidades ofrecidas si fuese necesario.

- **Crear las aplicaciones software que permitan modelar los procesos de negocio y generar aplicaciones multigénero y multiplataforma con soporte para los procesos de negocio utilizando la metodología propuesta.** El [capítulo 8](#), dedicado a los prototipos que acompañan a la metodología, presenta la herramienta de modelado BPLLevel Modeler y la herramienta de generación de aplicaciones BPLLevel Generator que permiten usar la metodología propuesta en esta investigación para modelar los procesos de negocio y crear aplicaciones informáticas con soporte para los dichos modelos de proceso.
- **Utilizar la metodología y las herramientas para modelar un conjunto de aplicaciones de distintas características, mostrando el grado de adaptación de la propuesta a los distintos ámbitos de aplicación y generando el código que ofrezca soporte a la funcionalidad descrita en los modelos de proceso.** La segunda parte del [capítulo 7](#) ofrece muestras de aplicación de la metodología BPLOM para el modelado de diversos procesos de negocio relacionados con distintos tipos de aplicaciones informáticas. Además, el [capítulo 8](#) incluye una explicación detallada del funcionamiento de las aplicaciones que acompañan a la metodología, para mostrar como se realiza el modelado del proceso y los pasos que se siguen para la creación automática de las aplicaciones asociadas a los modelos de proceso.
- **Comparar las aplicaciones modeladas y generadas utilizando la metodología propuesta con otras aplicaciones de similares funcionalidades desarrolladas con otras tecnologías, con el fin de analizar el coste asociado a la creación de las mismas en base a una serie de parámetros.** Durante el proceso de evaluación mostrado en el [capítulo 9](#) de este documento se muestra el análisis que se realiza de ciertos parámetros asociados a los modelos y al código de las aplicaciones que permiten realizar un cálculo aproximado del coste medio de creación de las aplicaciones con diversas alternativas. En dicho proceso de evaluación se puede comprobar que la metodología de modelado BPLOM permite modelar y crear aplicaciones con un coste medio asociado menor en comparación con el de las otras alternativas estudiadas.

### 10.3 ÁMBITOS DE APLICACIÓN

La metodología de modelado por niveles que se propone en este trabajo de investigación junto con los prototipos que permiten su utilización tienen los posibles ámbitos de aplicación que se especifican a continuación:

1. **Reducción de errores durante la fase de captura de requisitos.** Tal y como se menciona en el trabajo de Sommerville [Kotonya and Sommerville 1996], el proceso de captura de requisitos es una de las fases más importantes del desarrollo de aplicaciones informáticas. Los errores cometidos en esta fase, que normalmente vienen dados por la falta de conocimiento del dominio del problema que posee el técnico informático que se encarga de esta tarea, tienen un coste alto durante el resto de fases del desarrollo y del ciclo de vida de una aplicación. A través del uso de una notación de modelado adaptable al conocimiento del usuario que la utiliza, se obtiene un mayor grado de implicación de los expertos del dominio en la definición de las necesidades y funcionalidades asociadas a la aplicación. Si además se logra utilizar, como es el caso, los modelos construidos por los expertos para generar el código de la aplicación descrita en el modelo, se reduce de forma considerable el número de imprecisiones que pueda cometerse en esta fase tan crítica del desarrollo.
2. **Construcción de prototipos y/o desarrollos finales con rapidez.** La construcción de aplicaciones informáticas en base a las necesidades del cliente supone un coste temporal demasiado elevado en aquellos casos en los que el cliente necesita obtener una solución de forma rápida. A través del uso combinado de una metodología de modelado y una herramienta generadora de aplicaciones como las propuestas en este documento es posible disminuir, de forma considerable, el tiempo necesario para la construcción de una aplicación que cumpla las necesidades del cliente.
3. **Ahorro en los costes de desarrollo para aplicaciones multiplataforma.** A la hora de construir aplicaciones informáticas que puedan ser utilizadas bajo múltiples plataformas, existe el problema de disponer de un equipo de desarrollo que disponga de conocimientos para realizar el desarrollo bajo todas estas tecnologías. La utilización de modelos, como los resultantes del uso de la metodología BPLOM en combinación con la herramienta BPLLevel Modeler, y su transformación automática en código permite obviar los detalles y las necesidades de cada plataforma y generar varias aplicaciones nativas de forma rápida.

#### **10.4 PRINCIPALES APORTACIONES**

La lista incluida a continuación representa el conjunto de aportaciones y beneficios más destacados que se desprenden de este trabajo de investigación:

- Reducción del número de fases necesarias a la hora de aplicar las técnicas de modelado de procesos de negocio para modelar un sistema y generar código a partir de los modelos que se construyan.
- Creación de una metodología de modelado de procesos de negocio que permite adaptar la notación de modelado al nivel de conocimiento técnico y de modelado del que disponga el experto del dominio que la utiliza. Esta metodología facilita el modelado de procesos por parte de los expertos del dominio y permite promover la implicación directa de los expertos en el proceso de captura de requisitos a partir de la construcción de modelos.
- Establecimiento de un ecosistema de modelado de procesos de negocio y creación automatizada de aplicaciones compuesto por una herramienta de modelado de procesos de negocio y una herramienta de generación de código. La herramienta de modelado es capaz de construir modelos de proceso independientes de la plataforma objetivo para la construcción de los artefactos mientras que la herramienta de generación de artefactos permite transformar, de manera automática, los modelos de procesos construidos por la otra herramienta en aplicaciones informáticas funcionales.
- Reducción del coste medio de desarrollo de las aplicaciones informáticas. Utilizando las herramientas que acompañan a este trabajo se ha logrado reducir el coste medio asociado al desarrollo de los siguientes tipos de aplicaciones: aplicaciones web y videojuegos para plataformas móviles.



### 10.5 PRINCIPALES RESULTADOS DE LA INVESTIGACIÓN

A continuación se incluye una lista con los trabajos que han sido presentados en diversos medios para exponer los resultados obtenidos en la investigación y recibir retroalimentación de la comunidad científica:

- Jaime Solís-Martínez, Jordán Pascual Espada, B. Cristina Pelayo G. Bustelo and Juan Manuel Cueva Lovelle (*En proceso de revisión*). “VGPM: Using Business Process Modeling for Videogame Modeling and Code Generation in Multiple Platforms”. *Computer Standards & Interfaces*. **JCR: 0.978 (año 2012)**.
- Jaime Solís-Martínez, Jordán Pascual Espada, B. Cristina Pelayo G. Bustelo and Juan Manuel Cueva Lovelle (*Publicado*). “BPMN MUSIM: Approach to improve the domain expert’s efficiency in business processes modeling for the generation of specific software applications”. *Expert Systems with Applications*. **JCR: 1.854 (año 2012)**.
- Jaime Solís-Martínez, Natalia García-Menéndez, B. Cristina Pelayo G-Bustelo and Juan Manuel Cueva Lovelle (*Publicado*). “BPLOM: BPM Level-Oriented Methodology for Incremental Business Process Modeling and Code Generation on Mobile Platforms”. *International Journal of Interactive Multimedia and Artificial Intelligence*. (Vol 2- num 2, año 2013).
- Jaime Solís. Generación automática de aplicaciones a partir de modelos de procesos de negocio. Póster presentado en las II Jornadas Doctorales de la Universidad de Oviedo (Noviembre 2012).
- Jaime Solís. Metodología por pasos para la aplicación de BPMN 2.0. Póster presentado en las I Jornadas Doctorales de la Universidad de Oviedo (Diciembre 2011).
- Jaime Solís Martínez, Vicente García Díaz, Begoña Cristina Pelayo García-Bustelo y Juan Manuel Cueva Lovelle. “Isastur Modeler: A tool for BPMN MUSIM”. 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI). 15-18 de Junio de 2011 en Chaves, Portugal.
- Jaime Solís Martínez, Vicente García Díaz, Begoña Cristina Pelayo García-Bustelo y Juan Manuel Cueva Lovelle. “BPMN MUSIM: Notación BPMN muy simplificada”. Simposio doctoral en la 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI). 15-18 de Junio de 2011 en Chaves, Portugal.

## 10.6 LÍNEAS DE INVESTIGACIÓN Y TRABAJO FUTURO

A pesar de que los objetivos planteados para este trabajo de investigación y mencionados en el [capítulo 1](#) han sido cumplidos, existe la posibilidad de realizar ampliaciones al trabajo realizado durante esta investigación. A continuación se incluye una enumeración de las líneas de investigación y el trabajo futuro que se contemplan.

- ✓ **Introducción de la metodología de modelado en nuevos dominios de conocimiento para comprobar su capacidad de adaptación a distintos escenarios.** La introducción de la metodología a nuevos dominios implica la realización de pruebas para conocer la reducción del esfuerzo necesario para el desarrollo de las aplicaciones que se obtiene a partir de la aplicación de la metodología en comparación al desarrollo tradicional ligado al dominio de conocimiento en cuestión. Además también se contempla seguir recabando datos acerca de la importancia que le otorgan los expertos del dominio a las entidades incluidas en la metodología, continuando con la recopilación de información relativa a este aspecto iniciada en el proceso de pruebas descrito en el [capítulo 9](#). A través de estos datos se pretende investigar si sería adecuado proponer modificaciones en la metodología en base a las tasas de importancia que le otorgan los expertos a las entidades y también en relación a la posibilidad de incluir el dominio como un factor de adaptación de la metodología, proponiendo que según el dominio escogido el conjunto de entidades disponible para la construcción del modelo cambie.
- ✓ **Desarrollo de nuevos test de usabilidad.** Estos test estarán enfocados a conocer la opinión de los expertos del dominio acerca de las dificultades y/o facilidades que experimentan al utilizar la metodología BPLOM para modelar sus procesos de negocio. En base a los resultados obtenidos en estos test de usabilidad podría darse el caso de realizar modificaciones en el conjunto de entidades que ofrece la metodología para modelar los procesos, ya sea añadiendo nuevos elementos o eliminando aquellos que experimenten tasas de uso bajas.
- ✓ **Diseñar y construir una herramienta colaborativa que permita a los expertos del dominio definir los elementos personalizados que necesitan para la construcción de sus modelos.** El uso de una herramienta de este tipo podría lograr un mayor nivel de implicación por parte de los expertos del dominio y una transferencia de conocimiento entre los expertos de los distintos dominios.
- ✓ **Dotar a la herramienta BPL Level Modeler de la capacidad necesaria para conocer los patrones de modelado comunes, de tal forma que ésta pueda predecir errores habituales de modelado cometidos por los expertos del dominio.** Utilizando este mecanismo la herramienta sería capaz de sugerir una estructura alternativa para el modelo de proceso que esté construyendo el experto. Esta alternativa será el resultado del análisis de modelos de proceso similares que hayan sido desarrollados con la herramienta y hayan servido para generar aplicaciones.

- ✓ **Integrar un mecanismo de validación de modelos en la herramienta BPLLevel Modeler para que se validen los modelos que se crean en base al tipo de aplicación que se está modelando y al nivel de la metodología que está siendo utilizado por el experto del dominio.** En base a los modelos construidos previamente para un dominio y un tipo de aplicación concretos, la herramienta de modelado validaría el modelo actual en base a la estructura de los modelos similares creados anteriormente.
- ✓ **Añadir a la estructura de BPLLevel Modeler un repositorio de modelos, de tal forma que el experto del dominio pueda consultar un historial de versiones del modelo en el que está trabajando.**
- ✓ **Incluir en BPLLevel Modeler una funcionalidad de generación de reportes que permita utilizar el repositorio de modelos para construir un documento de evolución de los procesos.** Aprovechando la funcionalidad de repositorio mencionada anteriormente, BPLLevel Modeler sería capaz de crear documentación relativa a la evolución del proceso a través de las versiones del modelo creadas.
- ✓ **Ampliar el número de plataformas y tecnologías soportadas por BPLLevel Generator para la generación de las aplicaciones.** A través de una monitorización de los géneros y plataformas de aplicaciones que cuenten con mayor uso, se estudiará la posibilidad de ampliar la oferta de tipos de aplicaciones y plataformas que estén soportadas por la herramienta de generación de código. Una de las opciones que se espera incluir en primer lugar es la generación de aplicaciones de escritorio aprovechando la tecnología .NET, ya que a pesar del auge de las aplicaciones web en los entornos profesionales se siguen utilizando aplicaciones de escritorio en muchos casos.
- ✓ **Introducir la posibilidad de despliegue automático de las aplicaciones en la nube y las tiendas de aplicaciones.** Los servicios de la nube, también conocidos como *cloud computing*, y las tiendas de aplicaciones promovidas por las distintas compañías suponen una revolución para el despliegue de las aplicaciones. La generación de aplicaciones propuesta por BPLLevel Generator es local, ya que los artefactos se generan en la propia máquina sobre la que se ejecuta la herramienta generadora y esto implica la necesidad de transportar los artefactos generados a los canales de distribución correspondientes. Por lo tanto, sería interesante ofrecer la publicación directa tanto en la nube como en las tiendas de aplicaciones pertinentes para facilitar la distribución de los artefactos generados y automatizar aún más el proceso de generación propuesto en este trabajo.
- ✓ **Desarrollo de documentación y tutoriales para facilitar la programación de los elementos personalizados y la configuración de las herramienta BPLLevel Modeler y BPLLevel Generator.** A la hora de brindar acceso al uso de la metodología es necesario poner a disposición de los usuarios la documentación que permita ampliar la funcionalidad de las herramientas que acompañan a la metodología para que estas puedan ser adaptadas a los casos de uso que surjan.



---

---

## BIBLIOGRAFÍA Y REFERENCIAS

---

---



**[Appian BPM]** Appian BPM Suite: <http://www.appian.com/bpm-software/bpm-suite.jsp>

**[ARIS]** Aris Platform:  
[http://www.softwareag.com/es/products/aris\\_platform/default.asp](http://www.softwareag.com/es/products/aris_platform/default.asp)

**[Bonita Open Sol.]** Bonita Open Solution: <http://www.bonitasoft.com/>

**[BP-VA]** Business Process Visual Architect: <http://www.visual-paradigm.com/product/bpva/>

**[BPDM]** Business Process Definition Metamodel: <http://www.omg.org/spec/BPDM/1.0/>

**[BPM Soft. Suite]** BPM Software Suite: <http://www.ultimus.com/>

**[BPMI]** Business Process Modeling Initiative: <http://www.bpmi.org/>

**[BPMN Editor]** Yaoqiang BPMN Editor: <http://sourceforge.net/projects/bpmn/>

**[BPMN]** Business Process Modeling Notation <http://www.bpmn.org>

**[Chinosi and Trombetta 2012]** Chinosi, M. & Trombetta, A. (2012) BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

**[Czarnecki and Eisenecker 2000]** Czarnecki, K. & Eisenecker, U. (2000). *Generative Programming*. Addison-Wesley.

**[Dijkstra 1972]** Dijkstra, E. W. (1972). The humble programmer. *Communications of the ACM*, 15(10):859-866.

**[Eloranta et al. 2006]** Eloranta, L.; Kallio, E. & Terho, I. (2006). A Notation Evaluation of BPMN and UML Activity Diagrams.

**[Elrad et al. 2001]** Elrad, T.; Filman, R. E. & Bader, A. (2001). Aspect-oriented programming: Introduction. *Communications of the ACM*, 44(10):29-32.

**[Ent. Architect]** Enterprise Architect: <http://www.sparxsystems.com/>

**[Espada et al. 2013]** Espada, J. P.; Díaz, V. G.; Crespo, R. G.; Martínez, O. S.; Cristina Pelayo G-Bustelo, B. & Lovelle, J. M. C. (2013). Using extended web technologies to develop Bluetooth multi-platform mobile applications for interact with smart things. *Information Fusion*.

**[Facilis BPMN]** Facilis BPMN: <http://www.statum.biz/web/guest/facilis-bpmn>

**[Fernández 2008]** Héctor Fernández Fernández (2008). SBPD – Una arquitectura dirigida por modelos basada en procesos de negocio.

**[Fernández et al. 2009]** Fernandez-Fernandez, H.; Palacios-González, E.; García-Díaz, V.; Pelayo G-Bustelo, B. Cristina; Sanjuán Martínez, O. & Cueva Lovelle, J. M (2009). Developing a Business Application with BPM and MDE. *International Journal of Artificial Intelligence and Interactive Multimedia (IJIMAI)*.

**[Fernández et al. 2010]** Fernández, H. F.; Palacios-González, E.; García-Díaz, V.; Pelayo GBustelo, B. C.; Sanjuán Martínez, O. & Cueva Lovelle, J. M. (2010). SBPMN — An easier business process modeling notation for business users. *Computer Standards & Interfaces*, 32(1-2), 18–28.

**[Fernández González 2012]** Fernández González, Pablo. (2012). *Redes de Petri y su aplicación al modelado de procesos de negocio*. Proyecto Fin de Carrera de la Escuela de Ingeniería Informática de la Universidad de Oviedo.

**[García-Díaz et al. 2010]** García-Díaz, V.; Fernández-Fernández, H.; Palacios-González, E.; G-Bustelo, B. C. P.; Sanjuan-Martínez, O. & Lovelle, J. M. C. (2010). Talisman mde. mixing mde principles. *Journal of Systems and Software*, 83(7):1179-1191.

**[García-Díaz et al. 2011]** García-Díaz, V.; G-Bustelo, B. C. P.; Sanjuan-Martínez, O.; Valdez, E. R. N. & Lovelle, J. M. C. (2011). MCTest: Towards an improvement of match algorithms for models. *IET Software* 6 (2), 127-139.

**[Greenfield et al. 2004]** Greenfield, J.; Short, K.; Cook, S. & Kent, S. (2004). *Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools*. John Wiley & Sons.

**[Havey 2005]** Havey, M. (2005). *Essential Business Process Modeling*. O'Reilly Media, Inc.

**[Intalio]** Intalio|BPMS: <http://www.intalio.com/bpms/designer>

**[Isastur]** Grupo Isastur: <http://www.grupoisastur.com>

**[JBoss]** JBoss Community: <http://www.jboss.org/>

**[jBPM]** jBPM: <http://www.jboss.org/jbpm>

**[Jensen and Kristensen 2009]** Jensen, K. & Kristensen, Lars M. (2009). *Coloured Petri Nets: Modeling and Validation of Concurrent Systems*. Springer.

**[Kent 2002]** Kent, S. (2002). Model driven engineering. IFM '02: Proceedings of the Third International Conference on Integrated Formal Methods, p. 286-298. Springer-Verlag.

**[Kleppe et al. 2003]** Kleppe, A. G.; Warmer, J. & Bast, W. (2003). *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., Inc.

**[Ko 2009]** Ko, Ryan K. L. (2009). A computer scientist's introductory guide to business process management (BPM). *Crossroads* v.15 n.4, p.11-18.

**[Kotonya and Sommerville 1996]** Kotonya, G. & Sommerville, I. (1996). Requirements engineering with viewpoints. *BCS/IEE Software Engineering Journal* 11,1, 5–18.

**[Laue and Gadatsch 2010]** Laue, R. & Gadatsch, A. (2010). Measuring the Understandability of Business Process Models - Are We Asking the Right Questions? *Business Process Management Workshops - BPM 2010 International Workshops and Education Track*.



**[Lauesen and Vinter 2001]** Lauesen, S. & Vinter, O. (2001). Preventing Requirement Defects: An Experiment in Process Improvement. *Requirements Engineering*, 6, 37-50.

**[Likert 1932]** Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*. 22 1-55.

**[Lu and Sadiq 2007]** Lu, R. & Sadiq, S. (2007). A Survey on Comparative Modelling Approaches. *Proceedings BIS'07*.

**[M. Draw]** Magic Draw: <https://www.magicdraw.com/home>

**[MDE-RG]** Grupo de investigación de ingeniería dirigida por modelos: <http://grupos.uniovi.es/web/mderg>

**[Mousotron]** Mousotron: <http://www.blacksunsoftware.com/mousotron.html>

**[Muehlen and Recker 2008]** zur Muehlen, M. & Recker, J. (2008). How much language is enough? Theoretical and Practical Use of the Business Process Modeling Notation. 20th International Conference on Advanced Information Systems Engineering.

**[OMG 2007]** OMG (2007). Uml 2.0 superstructure speci\_cation. Technical report, OMG. [www.omg.org/docs/formal/07-11-02.pdf](http://www.omg.org/docs/formal/07-11-02.pdf).

**[OMG]** Object Management Group <http://www.omg.org>

**[Petri 1962]** Petri, C.A. (1962). Kommunikation mit Automaten. PhD thesis, Institut fiir instrumentelle Mathematik, Bonn. (año 1962).

**[Recker 2008]** Recker, Jan C. (2008). BPMN Modeling – Who, Where, How and Why. *BPTrends*.

**[Recker 2010]** Recker, Jan C. (2010). Opportunities and Constraints: The Current Struggle with BPMN. *Business Process Management Journal* 15.

**[Recker et al. 2010]** Recker, J.; Safrudin, N. & Rosemann, M. (2010). How Novices Model Business Processes. *Business Process Management – 8th International Conference*.

**[Russell et al. 2006]** Russell, N.; van der Aalst, W.M.; ter Hofstede, A.H. & Wohed, P. (2006). On the suitability of UML 2.0 activity diagrams for business process modeling. *Proceedings of the 3rd Asia-Pacific Conference on Conceptual Modeling - Volume 53, ACM*.

**[Seidewitz 2003]** Seidewitz, E. (2003). What models mean. *IEEE Software*, 20(5):26-32.

**[Selic 2003]** Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, 20(5):19-25.

**[Selic 2008]** Selic, B. (2008). Mda manifestations. *The European Journal for the Informatics Professional (UPGRADE)*, 9(2):12-16.

**[Solís Martínez et al. 2011a]** Solís Martínez, J.; García Díaz, V.; Pelayo García-Bustelo, B. C. & Cueva Lovelle, J. M. (2011). BPMN MUSIM: Notación BPMN muy simplificada. 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI).

**[Solís Martínez et al. 2011b]** Solís Martínez, J.; García Díaz, V.; Pelayo García-Bustelo, B. C. & Cueva Lovelle, J. M. (2011). Isastur Modeler: A tool for BPMN MUSIM. 6ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI).

**[Solís-Martínez et al. 2013]** Solís-Martínez, J.; García-Menéndez, N.; Pelayo G-Bustelo, B. C. & Cueva Lovelle, J. M. (2013). BPLOM: BPM Level-Oriented Methodology for Incremental Business Process Modeling and Code Generation on Mobile Platforms. *International Journal of Interactive Multimedia and Artificial Intelligence*. v. 2- n. 2, p. 13-27.

**[Solís-Martínez et al. 2014]** Solís-Martínez, J.; Pascual Espada, J.; Pelayo G-Bustelo, B. C. & Cueva Lovelle, J. M. (2014). BPMN MUSIM: Approach to improve the domain expert's efficiency in business processes modeling for the generation of specific software applications. *Expert Systems with Applications* v. 41 Issue 4 Part 2 p. 1864-1874.

**[Sztipanovits and Karsai 1997]** Sztipanovits, J. and Karsai, G. (1997). Modelintegrated computing. *Computer*, 30(4): 110-111.

**[UML]** Unified Modeling Language. <http://www.uml.org>

**[UModel]** Altova UModel: <http://www.altova.com/umodel.html>

**[van der Aalst 1994]** van der Aalst, W.M.P. (1994). Putting high-level Petri nets to work in industry. *Computers in Industry*, 25(1): 45-54.

**[van der Aalst et al. 1998]** van der Aalst, W.M.P et al. (1998) The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1): 21-66.

**[van Deursen 1997]** van Deursen, A. (1997). Domain-specific languages versus object-oriented frameworks: A financial engineering case study. *Proceedings of Smalltalk and Java in Industry and Academia*, p. 35-39.

**[van Deursen et al. 2000]** van Deursen, A.; Klint, P. & Visser, J. (2000). Domain-specific languages: an annotated bibliography. *SIGPLAN Not.*, 35(6):26-36.

**[Visio]** Microsoft Visio: <http://office.microsoft.com/es-es/visio/pagina-principal-de-visio-2010-FX010048786.aspx>

**[Völter and Stahl 2006]** Völter, M. & Stahl, T. (2006). *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons.

**[Wahl and Sindre 2005]** Wahl, T. & Sindre, G. (2005). An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. *CAiSE'05 Workshops*. v. 1, p. 533-544.

**[Ward 1994]** Ward, M. (1994). Language oriented programming. *Software-Concepts and Tools*, 15:147-161.

**[White 2004]** White, Stephen A. (2004). *Introduction to BPMN*.

**[White and Miers 2008]** White, Stephen A. & Miers, D. (2008). *BPMN Modeling and Reference Guide: Understanding and Using BPMN*. Future Strategies Inc.

**[WoPeD]** Workflow Petri Net Designer (WoPeD): <http://woped.org/>

**[WWF]** Windows Workflow Foundation: <http://msdn.microsoft.com/en-us/netframework/aa663328>

**[Zhiqiang et al. 2010]** Zhiqiang, Y.; Reijers, Hajo A. & Dijkman, Remco M. (2010). An evaluation of BPMN Modeling Tools. Business Process Modeling Notation - Second International Workshop.



---

---

# APÉNDICES

---

---



---

---

# APÉNDICE I

## MODELO DE PROCESO DE NEGOCIO

---

---





## MODELO DE PROCESO DE NEGOCIO

A continuación se incluye, a modo de ejemplo, el contenido de un fichero XAMLX que representa a uno de los procesos de negocio incluido en el caso de estudio que compone el [capítulo 7](#) de este documento. En concreto, este fichero representa el modelo de proceso de inserción de incidencias que compone el sistema de gestión de incidencias descrito en dicho caso de estudio. Este fichero ha sido construido utilizando el nivel 0 de la metodología BPLOM, denominado BPMN MUSIM.

Tal y como se puede observar el contenido del fichero está compuesto por etiquetas XML en las que se incluye información adicional relacionada con la posición de los elementos en la representación gráfica del proceso y con las conexiones que existen entre las distintas entidades que conforman la estructura del modelo.

Este tipo de ficheros son generados por la herramienta BPLLevel Modeler en base al modelado de los procesos que realizan los expertos de forma gráfica y son utilizados, como uno de sus *inputs*, por BPLLevel Generator para la creación de la aplicación personalizada que ofrezca soporte al proceso de negocio descrito en el modelo.

```
<WorkflowService mc:Ignorable="sap" ConfigurationName="servicio"
sap:VirtualizedContainerService.HintSize="704,666" Name="servicio"
mva:VisualBasic.Settings="Assembly references and imported namespaces serialized as XML
namespaces" xmlns="http://schemas.microsoft.com/netfx/2009/xaml/servicemodel"
xmlns:av="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:cac="clr-
namespace:ComponentesWorkflow.Actividades.Comunes;assembly=BPLLevel Modeler"
xmlns:cag="clr-
namespace:ComponentesWorkflow.Actividades.GestionIncidencias;assembly=BPLLevel Modeler"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:mva="clr-
namespace:Microsoft.VisualBasic.Activities;assembly=System.Activities"
xmlns:p="http://schemas.microsoft.com/netfx/2009/xaml/activities"
xmlns:sap="http://schemas.microsoft.com/netfx/2009/xaml/activities/presentation" xmlns:scg="clr-
namespace:System.Collections.Generic;assembly=microsoftlib"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <p:Flowchart DisplayName="Nuevo Proceso" sap:VirtualizedContainerService.HintSize="674,636"
mva:VisualBasic.Settings="Assembly references and imported namespaces serialized as XML
namespaces">
    <sap:WorkflowViewStateService.ViewState>
      <scg:Dictionary x:TypeArguments="x:String, x:Object">
        <x:Boolean x:Key="IsExpanded">False</x:Boolean>
        <av:Point x:Key="ShapeLocation">270,2.5</av:Point>
        <av:Size x:Key="ShapeSize">60,75</av:Size>
        <av:PointCollection x:Key="ConnectorLocation">300,77.5 300,107.5 300,109</av:PointCollection>
        <x:Double x:Key="Width">660</x:Double>
      </scg:Dictionary>
    </sap:WorkflowViewStateService.ViewState>
    <p:Flowchart.StartNode>
      <x:Reference>__ReferenceID6</x:Reference>
    </p:Flowchart.StartNode>
    <p:FlowStep x:Name="__ReferenceID6">
      <sap:WorkflowViewStateService.ViewState>
        <scg:Dictionary x:TypeArguments="x:String, x:Object">
          <av:Point x:Key="ShapeLocation">200,109</av:Point>
          <av:Size x:Key="ShapeSize">200,22</av:Size>
          <av:PointCollection x:Key="ConnectorLocation">300,131 300,159</av:PointCollection>
        </scg:Dictionary>
      </sap:WorkflowViewStateService.ViewState>
      <cac:Login request="{x:Null}" sap:VirtualizedContainerService.HintSize="200,22" />
    </p:FlowStep.Next>
    <p:FlowStep x:Name="__ReferenceID1">
```

```

<sap:WorkflowViewStateService.ViewState>
  <scg:Dictionary x:TypeArguments="x:String, x:Object">
    <av:Point x:Key="ShapeLocation">200,159</av:Point>
    <av:Size x:Key="ShapeSize">200,22</av:Size>
    <av:PointCollection x:Key="ConnectorLocation">300,181 300,209</av:PointCollection>
  </scg:Dictionary>
</sap:WorkflowViewStateService.ViewState>
<cag:IncidenciasUsuario request="{x:Null}" sap:VirtualizedContainerService.HintSize="200,22" />
<p:FlowStep.Next>
  <p:FlowStep x:Name="__ReferenceID0">
    <sap:WorkflowViewStateService.ViewState>
      <scg:Dictionary x:TypeArguments="x:String, x:Object">
        <av:Point x:Key="ShapeLocation">200,209</av:Point>
        <av:Size x:Key="ShapeSize">200,22</av:Size>
        <av:PointCollection x:Key="ConnectorLocation">300,231 300,261
300,262.5</av:PointCollection>
      </scg:Dictionary>
    </sap:WorkflowViewStateService.ViewState>
    <cag:IncidenciasPendientes lista="{x:Null}" sap:VirtualizedContainerService.HintSize="200,22"
/>
    <p:FlowStep.Next>
      <p:FlowDecision x:Name="__ReferenceID2"
sap:VirtualizedContainerService.HintSize="60,75">
        <sap:WorkflowViewStateService.ViewState>
          <scg:Dictionary x:TypeArguments="x:String, x:Object">
            <av:Point x:Key="ShapeLocation">270,262.5</av:Point>
            <av:Size x:Key="ShapeSize">60,75</av:Size>
            <av:PointCollection x:Key="TrueConnector">270,300 210,300</av:PointCollection>
            <av:PointCollection x:Key="FalseConnector">330,300 470,300
470,319</av:PointCollection>
          </scg:Dictionary>
        </sap:WorkflowViewStateService.ViewState>
        <p:FlowDecision.True>
          <p:FlowStep x:Name="__ReferenceID3">
            <sap:WorkflowViewStateService.ViewState>
              <scg:Dictionary x:TypeArguments="x:String, x:Object">
                <av:Point x:Key="ShapeLocation">10,289</av:Point>
                <av:Size x:Key="ShapeSize">200,22</av:Size>
                <av:PointCollection x:Key="ConnectorLocation">110,289 110,220
200,220</av:PointCollection>
              </scg:Dictionary>
            </sap:WorkflowViewStateService.ViewState>
            <cag:CalificarIncidencia lista="{x:Null}" request="{x:Null}"
sap:VirtualizedContainerService.HintSize="200,22" />
            <p:FlowStep.Next>
              <x:Reference>__ReferenceID0</x:Reference>
            </p:FlowStep.Next>
          </p:FlowStep>
        </p:FlowDecision.True>
        <p:FlowDecision.False>
          <p:FlowStep x:Name="__ReferenceID4">
            <sap:WorkflowViewStateService.ViewState>
              <scg:Dictionary x:TypeArguments="x:String, x:Object">
                <av:Point x:Key="ShapeLocation">370,319</av:Point>
                <av:Size x:Key="ShapeSize">200,22</av:Size>
                <av:PointCollection x:Key="ConnectorLocation">470,341 470,369</av:PointCollection>
              </scg:Dictionary>
            </sap:WorkflowViewStateService.ViewState>
            <cag:NuevaIncidencia request="{x:Null}"
sap:VirtualizedContainerService.HintSize="200,22" />
            <p:FlowStep.Next>
              <p:FlowStep x:Name="__ReferenceID5">
                <sap:WorkflowViewStateService.ViewState>
                  <scg:Dictionary x:TypeArguments="x:String, x:Object">
                    <av:Point x:Key="ShapeLocation">370,369</av:Point>
                    <av:Size x:Key="ShapeSize">200,22</av:Size>
                  </scg:Dictionary>
                </sap:WorkflowViewStateService.ViewState>
              </p:FlowStep>
            </p:FlowStep.Next>
          </p:FlowStep>
        </p:FlowDecision.False>
      </p:FlowStep>
    </p:FlowDecision>
  </p:FlowStep>
</p:FlowStep.Next>
  <p:FlowStep x:Name="__ReferenceID5">
    <sap:WorkflowViewStateService.ViewState>
      <scg:Dictionary x:TypeArguments="x:String, x:Object">
        <av:Point x:Key="ShapeLocation">370,369</av:Point>
        <av:Size x:Key="ShapeSize">200,22</av:Size>
      </scg:Dictionary>
    </sap:WorkflowViewStateService.ViewState>
  </p:FlowStep>
</p:FlowStep>

```

```

        </sap:WorkflowViewStateService.ViewState>
        <p:TerminateWorkflow sap:VirtualizedContainerService.HintSize="200,22" Reason="fin
del proceso" />
        </p:FlowStep>
        </p:FlowStep.Next>
        </p:FlowStep>
        </p:FlowDecision.False>
        </p:FlowDecision>
        </p:FlowStep.Next>
        </p:FlowStep>
        </p:FlowStep.Next>
        </p:FlowStep>
        </p:FlowStep.Next>
        </p:FlowStep>
        <x:Reference>__ReferenceID1</x:Reference>
        <x:Reference>__ReferenceID0</x:Reference>
        <x:Reference>__ReferenceID2</x:Reference>
        <x:Reference>__ReferenceID3</x:Reference>
        <x:Reference>__ReferenceID4</x:Reference>
        <x:Reference>__ReferenceID5</x:Reference>
    </p:Flowchart>
</WorkflowService>

```



---

---

## APÉNDICE II

### SERVICIO WEB ASOCIADO A UN MODELO DE PROCESO

---

---



## SERVICIO WEB ASOCIADO A UN MODELO DE PROCESO

Tal y como se explicó en el apartado correspondiente a los ejemplos de uso de los prototipos para la generación de aplicaciones, incluido en el [capítulo 8](#) de este documento, en caso de que el usuario elija crear una aplicación web a partir del modelo de proceso construido con BPLLevel Modeler, el proceso de generación transforma el modelo de proceso definido en un servicio web personalizado.

Para ello se utilizan las características de la plataforma Windows Workflow Foundation, que posibilitan la creación de un fichero WSDL, un tipo de ficheros que describen los servicios web en formato XML, de forma automática a partir de los ficheros XAMLX generados por el prototipo de herramienta de modelado presentado en este trabajo.

A continuación se incluye el fichero WSDL resultante del proceso de generación de una aplicación web a partir del modelo de proceso que representa la inserción de incidencias, cuyo fichero XAMLX ha sido presentado en el apartado anterior. Tal y como se puede observar este fichero descriptor del servicio web está configurado para la recepción de las peticiones de la aplicación web y el envío de las respuestas correspondientes a la realización de las acciones que componen el proceso de inserción de incidencias. Además, se puede observar como existe una operación declarada en este fichero para cada una de las actividades incluidas en el modelo de proceso correspondiente.

```
<wsdl:definitions name="servicio" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://tempuri.org/Imports">
      <xsd:import schemaLocation="http://localhost:50975/servicio.xamlx?xsd=xsd1"
        namespace="http://schemas.datacontract.org/2004/07/ComplementosWorkflow.Contratos" />
      <xsd:import schemaLocation="http://localhost:50975/servicio.xamlx?xsd=xsd0"
        namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xsd:import schemaLocation="http://localhost:50975/servicio.xamlx?xsd=xsd2"
        namespace="http://schemas.datacontract.org/2004/07/DesignerRehosting.BaseDatos" />
    </xsd:schema>
  </wsdl:types>

  <wsdl:message name="servicio_recibeLogin_InputMessage">
    <wsdl:part name="LoginRequest" element="q1:LoginRequest"/>
  </wsdl:message>

  <wsdl:message name="servicio_recibeLogin_OutputMessage">
    <wsdl:part name="LoginResponse" element="q2:LoginResponse"/>
  </wsdl:message>

  <wsdl:message name="servicio_recibeListar_InputMessage">
    <wsdl:part name="ListarRequest" element="q3:ListarRequest"/>
  </wsdl:message>

  <wsdl:message name="servicio_recibeListar_OutputMessage">
    <wsdl:part name="ArrayOfIncidencia" element="q4:ArrayOfIncidencia"/>
  </wsdl:message>

  <wsdl:message name="servicio_recibeNueva_InputMessage">
    <wsdl:part name="NuevaIncidenciaRequest" element="q5:NuevaIncidenciaRequest"/>
  </wsdl:message>

  <wsdl:message name="servicio_recibeNueva_OutputMessage">
    <wsdl:part name="NuevaIncidenciaResponse" element="q6:NuevaIncidenciaResponse"/>
  </wsdl:message>
```

```

<wsdl:message name="servicio_recibeCalificar_InputMessage">
<wsdl:part name="CalificarRequest" element="q7:CalificarRequest"/>
</wsdl:message>

<wsdl:message name="servicio_recibeCalificar_OutputMessage">
<wsdl:part name="CalificarResponse" element="q8:CalificarResponse"/>
</wsdl:message>

<wsdl:portType name="servicio">
<wsdl:operation name="recibeLogin">
<wsdl:input wsaw:Action="http://tempuri.org/servicio/recibeLogin"
message="tns:servicio_recibeLogin_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/servicio/recibeLoginResponse"
message="tns:servicio_recibeLogin_OutputMessage"/>
</wsdl:operation>

<wsdl:operation name="recibeListar">
<wsdl:input wsaw:Action="http://tempuri.org/servicio/recibeListar"
message="tns:servicio_recibeListar_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/servicio/recibeListarResponse"
message="tns:servicio_recibeListar_OutputMessage"/>
</wsdl:operation>

<wsdl:operation name="recibeNueva">
<wsdl:input wsaw:Action="http://tempuri.org/servicio/recibeNueva"
message="tns:servicio_recibeNueva_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/servicio/recibeNuevaResponse"
message="tns:servicio_recibeNueva_OutputMessage"/>
</wsdl:operation>

<wsdl:operation name="recibeCalificar">
<wsdl:input wsaw:Action="http://tempuri.org/servicio/recibeCalificar"
message="tns:servicio_recibeCalificar_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/servicio/recibeCalificarResponse"
message="tns:servicio_recibeCalificar_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>

<wsdl:binding name="BasicHttpBinding_servicio" type="tns:servicio">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="recibeLogin">
<soap:operation soapAction="http://tempuri.org/servicio/recibeLogin" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="recibeListar">
<soap:operation soapAction="http://tempuri.org/servicio/recibeListar" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="recibeNueva">
<soap:operation soapAction="http://tempuri.org/servicio/recibeNueva" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
</wsdl:output>

```



```
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="recibeCalificar">
<soap:operation soapAction="http://tempuri.org/servicio/recibeCalificar" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="servicio">
  <wsdl:port name="BasicHttpBinding_servicio" binding="tns:BasicHttpBinding_servicio">
    <soap:address location="http://localhost:50975/servicio.xamlx"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```



---

---

## GLOSARIO Y DICCIONARIO DE DATOS

---

---



**BPLOM: Business Process Level Oriented Methodology.** Metodología de modelado de procesos de negocio por niveles, propuesta en este trabajo de investigación como una alternativa al modelado de procesos de negocio por parte de expertos del dominio.

**BPM: Business Process Modeling.** Conjunto de técnicas que permiten realizar el modelado de los procesos para cualquier área de negocio, estableciendo mecanismos para el modelado, la gestión, la optimización y la ejecución de estos procesos

**BPMI: Business Process Modeling Initiative.** Grupo de trabajo existente dentro del OMG que propone la creación de la notación estándar de modelado de procesos de negocio.

**BPMN MUSIM: BPMN MUy SIMplificada.** Nivel inicial de la metodología de modelado de procesos de negocio por niveles. Se considera el conjunto mínimo de símbolos necesarios para el modelado de cualquier proceso de negocio.

**BPMN: Business Process Modeling Notation.** Notación estándar de modelado de procesos de negocio propuesta por el grupo BPMI, perteneciente a la estructura del OMG.

**jBPM: Java Business Process Modeling.** Suite de BPM implementada sobre la plataforma Java, propuesta por la JBoss Community.

**jPDL: Java Process Definition Language.** Lenguaje de definición de procesos de negocio utilizado por la suite jBPM. Fue concebido como una simplificación de la notación estándar de modelado de procesos de negocio a través de la disminución del número de entidades que ofrecer para la construcción de los modelos y de la inclusión de un código de colores.

**MDE: Model Driven Engineering.** Aproximación al desarrollo de software basada en la utilización de modelos y la realización de transformaciones entre estos propuesta en el año 2000. Intenta solucionar los problemas tradicionales del desarrollo de software a través de la automatización de algunas de sus fases.

**OMG: Object Management Group.** Organización dedicada a la creación y promoción de estándares relacionados con las tecnologías orientadas a objetos. Algunos de los estándares creados por este consorcio son: UML, BPMN y XMI, entre otros.

**SBPMN: Simple BPMN.** Simplificación de la notación estándar de modelado de procesos de negocio propuesta por el grupo de investigación MDE-RG de la Universidad de Oviedo.

**UML AD: UML Activity Diagrams.** Tipo de diagramas incluidos en el lenguaje unificado de modelado. Son los encargados de ofrecer soporte para el modelado de procesos de negocio.

**UML: Unified Modeling Language.** Lenguaje unificado de modelado de sistemas software, es el más utilizado en la actualidad. Los tipos de diagramas que es posible crear con este lenguaje están divididos en las siguientes categorías: estructura, comportamiento e interacción.

**WWF: Windows Workflow Foundation.** Componente de la plataforma .NET de Microsoft que incluye los mecanismos necesarios para ofrecer soporte a BPM.

**XML: Extensible Markup Language.** Lenguaje de marcas desarrollado por el W3C que es utilizado para el almacenamiento de información de forma legible e independiente de cualquier plataforma. Este lenguaje es utilizado de forma común para el almacenamiento de modelos.

---

---

# ÍNDICE ALFABÉTICO

---

---





<b>A</b>	
Appian BPM Suite .....	100
ARIS Platform.....	98
<b>B</b>	
Bonita Open Solution .....	100
BPLLevel Generator .....	234
BPLLevel Modeler.....	227
BPLM .....	136
BPM .....	49
BPM Software Suite.....	99
BPMI .....	32
BPMN MUSIM.....	139
Business Process Visual Architect.....	95
<b>E</b>	
Enterprise Architect.....	93
entorno de trabajo de jBPM .....	101
entorno de trabajo de Windows Workflow Foundation .....	105
esquema de funcionamiento de BPM .....	50
estructura de aplicación de la metodología BPLM .....	223
extensión de actividades .....	145
extensión de datos .....	147
extensión de decisiones .....	143
extensión de eventos .....	144
<b>F</b>	
Facilis BPMN .....	96
<b>H</b>	
herramientas de modelado .....	92
<b>I</b>	
Intalio BPMS .....	99
<b>J</b>	
jBPM .....	101
jPDL .....	72
<b>M</b>	
Magic Draw .....	93
MDE.....	111
Microsoft Visio .....	92
<b>O</b>	
OMG.....	52
<b>P</b>	
plataformas de modelado y ejecución .....	98
<b>R</b>	
redes de Petri .....	79
redes de Petri de alto nivel .....	82
<b>S</b>	
SBPMN .....	75
<b>U</b>	
UML Activity Diagrams.....	66
UModel .....	94
<b>W</b>	
Windows Workflow Foundation .....	104, 227
Workflow Petri Net .....	83
<b>Y</b>	
Yaoqiang BPMN Editor .....	95