



Universidad
de Alcalá

UNIVERSIDAD DE ALCALÁ
ESCUELA POLITÉCNICA SUPERIOR
TRABAJO FIN DE GRADO

Autor:

Pedro Palazón Martínez de Alegría

Tutor:

Salvador Otón Tortosa

**Implementación de una herramienta de apoyo al
despliegue de Software bajo el prisma de
ITIL v.3 2011 con Spring MVC y un BPM**



Universidad
de Alcalá

UNIVERSIDAD DE ALCALÁ
ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE GRADO

Autor:

Pedro Palazón Martínez de Alegría

Tutor:

Salvador Otón Tortosa

Tribunal:

Presidente:

Vocal 1º:

Vocal 2º:

Calificación:

Fecha: ... de de

*A mis padres por su amor incondicional.
A mi compañera de viaje, por traer la luz a mis días.
Gracias por todo.*

Pregúntate si lo que estás haciendo hoy te acerca al lugar en el que quieres estar mañana.

Walt Disney

CAPÍTULO 1

RESUMEN



ÍNDICE RESUMINDO

1. RESUMEN	17
2. PALABRAS CLAVE	21
3. RESUMEN EJECUTIVO	25
4. GLOSARIO DE TÉRMINOS Y ABREVIATURAS	33
5. MEMORIA	39
6. PRESUPUESTO Y MEDIOS EMPLEADOS.....	123
7. CONCLUSIONES Y FUTURAS LÍNEAS DE ACTUACIÓN.....	129
8. BIBLIOGRAFÍA.....	135
9. ANEXO A – Código Fuente.....	141

ÍNDICE GENERAL

1.	RESUMEN	17
1.1.	Resumen	17
1.2.	Abstract	18
2.	PALABRAS CLAVE	21
2.1.	Palabras Clave	21
2.2.	Key Words	21
3.	RESUMEN EJECUTIVO	25
3.1.	Introducción al trabajo a realizar	25
3.2.	Objetivos	26
3.3.	Resultados	27
3.4.	Metodología	27
3.5.	Recursos	28
3.6.	Planificación	28
4.	GLOSARIO DE TÉRMINOS Y ABREVIATURAS	33
4.1	Glosario de Abreviaturas	33
4.2	Glosario de Términos	34
5.	MEMORIA	39
5.1.	Introducción	39
5.1.1.	Escenarios e hipótesis de partida	39
5.1.2.	Objetivos para el proyecto y propuesta de solución	42
5.2.	Problemática planteada	42
5.3.	Estado del arte.....	44
5.3.1.	Las Factorías de Software.....	44
5.3.2.	Marco de referencia ITIL	47
5.3.2.1.	Mapa conceptual ITIL	48
5.3.2.2.	Proceso ITIL. Mejora Continua del Servicio.....	48
5.3.3.	Herramientas BPM y su soporte a la prestación de servicios	49
5.3.3.1.	Diagrama de Procesos de una Factoría	49
5.3.3.2.	Herramientas BPM	50
5.3.3.3.	Notación BPMN	51
5.3.3.4.	Principales BPM de mercado.....	52
5.4.	Diseño de la Solución	55
5.4.1.	Diseño Conceptual de la Solución	55
5.4.1.1.	Juntándolo todo. Desafío de Diseño	55

5.4.1.2. Elección de herramienta BPM.....	57
5.4.1.3. Diseño de un BPM con Activiti	58
5.4.2. Diseño Funcional de la Solución	75
5.4.2.1. Marco de Trabajo	75
5.4.2.2. Pila de Producto. Historias de Usuario.....	77
5.4.2.3. Diseñando un BPM para el despliegue del Software	79
5.4.2.4. Presentando la solución CofaBPM	84
5.4.3. Diseño Técnico de la Solución	85
5.4.3.1. Solución Arquitectónica	85
5.4.3.2. Hosting87	
5.4.3.3. Seguridad.....	89
5.4.3.4. BackEnd del Sistema.....	95
5.4.3.4.1. Configuración de la Aplicación	96
5.4.3.5. FrontEnd del Sistema	103
5.5. Normativa.....	118
6. PRESUPUESTO Y MEDIOS EMPLEADOS.....	123
6.1. Programas Utilizados.....	123
6.2. Tiempos Incurridos.....	123
6.3. Costes	124
6.4. Beneficios Esperados.....	125
7. CONCLUSIONES Y FUTURAS LÍNEAS DE ACTUACIÓN.....	129
7.1. Conclusiones.....	129
7.2. Acciones Futuras	131
8. BIBLIOGRAFÍA.....	135
8.1 Relación de Referencias Bibliográficas.....	135
9. ANEXO A – Código Fuente	141
9.1 BPM generado	141
9.2 BackEnd del Sistema.....	154
9.3 FrontEnd	169

ÍNDICE DE FIGURAS

Figura 3-1 Diagrama de Gantt Inicial.....	29
Figura 5-1 Tipologías de Mantenimientos Software.....	40
Figura 5-2 Ejemplo de Hilo de conversación sobre unas pruebas	43
Figura 5-3 Problemáticas de las factorías de sw que tienen muchos stakeholders	43
Figura 5-4 Problemáticas de las factorías de sw que tienen muchos stakeholders	44
Figura 5-5. Evolución de las factorías de Sw [14].....	46
Figura 5-6 Modelo Conceptual ITIL	47
Figura 5-7 . Modelo de prestación de servicios.....	49
Figura 5-8 BPM integrado con otras herramientas de gestión empresarial.....	51
Figura 5-9 Tabla comparativa herramientas finalistas BPM	57
Figura 5-10 Suite Activiti	58
Figura 5-11 Tabla Comparativa SGBD	61
Figura 5-12 Visibilidad de los diferentes componentes de Activiti.....	62
Figura 5-13 Pantalla de Descarga.....	65
Figura 5-14 Paquete Descargado	66
Figura 5-15 Entrada al Activiti Modeler	66
Figura 5-16 Menú del Activiti Modeler	67
Figura 5-17 Detalle botón crear proceso	67
Figura 5-18 Formulario de Creación	68
Figura 5-19 Pantalla Principal del Modelador.....	68
Figura 5-20 BPM de Ejemplo.....	69
Figura 5-21 Detalle creación de formulario	69
Figura 5-22 Biblioteca de procesos modelados	70
Figura 5-23 BPM Evolutivo Parte I	81
Figura 5-24 BPM Evolutivo Parte II	82
Figura 5-25 BPM Evolutivo Parte III	83
Figura 5-26 Cofa de un barco.....	84
Figura 5-27 Arquitectura Modelo Vista Controlador.....	85
Figura 5-28 Pila Arquitectónica de CofaBPM.....	86
Figura 5-29 Nominalia Panel de Control	87
Figura 5-30 Droplet CofabPM Digital Ocean.....	88
Figura 5-31 Conexión Segura	89
Figura 5-32 de la CA Certificado Let's Encrypt	90
Figura 5-33 Nota de Qualys SSL Labs	91
Figura 5-34 Grupos de Participantes	93
Figura 5-35 Estructura de Carpetas del Proyecto	95
Figura 5-36 Diagrama de Clases.....	98
Figura 5-37 Pantalla de Login.....	103
Figura 5-38 Uso de SpringMVC	104
Figura 5-39 Cabecera de la Aplicación	105
Figura 5-40 Cuerpo de la Aplicación	105
Figura 5-41 Pie de la Aplicación	106
Figura 5-42 Diagrama de la Aplicación Front.....	107

Figura 5-43 Menu.....	108
Figura 5-44 Consulta de Buzones.....	109
Figura 5-45 Formulario tareas de usuario: Entrega Diseño Físico	110
Figura 5-46 Consulta de Evolutivos.....	111
Figura 5-47 Detalle del Evolutivo	112
Figura 5-48 Sala de Máquinas.....	113
Figura 5-49 Finalización de la Sesión	114
Figura 5-50 Mensaje de error	115
Figura 5-51 Mensaje de éxito	115
Figura 5-52 Mensaje de estado de conexión.....	116
Figura 5-53 Mensaje de control de acceso	116
Figura 5-54 Plantilla de email enviada.....	117
Figura 6-1 Diagrama de Gantt Inicial	123

1. RESUMEN

1.1. Resumen

En este proyecto se presenta y expone la implementación con Activiti, de una herramienta de soporte al proceso de desarrollo y despliegue de sw de carácter evolutivo.

La automatización de este proceso a través de un workflow mejora la comunicación de los stakeholders implicados, y garantiza la normalización del mismo. Además, se su repetibilidad y estabilidad.

El proceso de desarrollo de sw evolutivo no es un proceso aislado, ya que, en el escenario presentado, se encuentra dentro de una factoría de sw cuya operativa se realiza desde la perspectiva de un modelo de gestión por procesos. Además, y a modo de contexto, se especifica que la necesidad de mejorar la comunicación dentro de este proceso, se ha detectado a partir de los resultados del proceso de mejora continua que se implanta cuando dentro de una compañía se trabaja con respecto al modelo de buenas prácticas denominado ITIL, como es el caso.

Como resultado y concluyendo, es cierto que la implantación de una herramienta de tipo workflow, que normaliza la forma de trabajar, implica a todos los roles participantes, ordena la forma y tipo de comunicación y permite la trazabilidad y no repudio de los mensajes, redonda en unos resultados mejores en cuanto a:

- Registro de las necesidades de los clientes
- Claridad en los requisitos
- Existencia de aceptación y certificaciones formales
- Disponibilidad de un histórico de control de cambios.
- Disponibilidad de una trazabilidad de los acuerdos tomados.
- Mejora de comunicación síncrona y asíncrona

1.2. Abstract

This project presents and exposes the implementation of a tool with Activiti. Which support the process of development and deployment of sw of an evolutionary character.

The automation of this process through a workflow improves the communication of the stakeholders involved, and guarantees the standardization of the same. Besides its repeatability and stability.

The process of development of evolutionary sw is not an isolated process, since in the presented scenario, it is located within a factory of sw whose operation is carried out from the perspective of a management model by processes.

In addition, as a context, it is specified that the need to improve communication within this process has been detected from the results of the process of continuous improvement that is implemented when a company is working with respect to the model of good practices called ITIL, as is the case.

As a result and concluding, it is true that the implementation of a workflow-type tool, which normalizes the way of working, involves all the participating roles, sorts the form and type of communication and allows for traceability and non-repudiation of messages, in better results in terms of:

- Record customer needs.
- Clarity in the requirements.
- Existence of acceptance and formal certifications.
- Availability of a change control history.
- Availability of a traceability of the agreements taken.
- Improvement of synchronous and asynchronous communication.

CAPÍTULO 2

PALABRAS CLAVE



2. PALABRAS CLAVE

2.1. Palabras Clave

Modelo de Gestión por Procesos, Flujo de Trabajo, Proceso, Proceso automatizado,
Herramientas Colaborativas; MGP; mejora continua; ITIL

2.2. Key Words

Process Management Model; workflow; process; automated process; collaborative tools;
BPM; improvement; continuous improvement; TIL

CAPÍTULO 3

RESUMEN EJECUTIVO



3. RESUMEN EJECUTIVO

3.1. Introducción al trabajo a realizar

Disponer de aplicaciones con un código limpio y verificable como elemento productivo para la reutilización y mantenimiento software, así como la optimización y mejora de procesos, es un reto para todas las compañías de desarrollo Software.

Actualmente, la revolución del uso de la tecnología hace que la hegemonía de las compañías y sus valores diferenciales cambien drásticamente, esto es, “competitividad por intangibles”^[1] por lo que el uso de la tecnología, la adaptabilidad y la mejora en la gestión es el objetivo de todas ellas.

En este sentido, la implantación de la gestión de procesos^[2] se ha revelado como una de las herramientas de mejora de gestión más efectivas³. Cualquier actividad, que utiliza recursos para transformar elementos de entrada (especificaciones, información, servicios,...) en resultados, puede considerarse como un proceso. Tradicionalmente, el soporte principal de un proceso era un sistema de información. En este momento, estos sistemas son los denominados BPMS (Business Process Management Suite) herramientas que permiten una mayor flexibilidad, automatización y potencia en la gestión de operaciones, puesto que las ventajas que ofrecen los BPMS provienen del uso de MDA (Model Driven Architecture)⁴ y de SOA (Service Oriented Architecture)⁵, reduciendo la brecha entre dominios, en este caso, entre el modelado de procesos de negocio, y las características de los lenguajes de programación actuales. Por su parte, SOA permite (re)utilizar componentes de software distribuidos en múltiples aplicaciones para crear y modificar estos procesos, de manera que la integración de aplicaciones se vuelve casi transparente desde un BPMS para los diferentes procesos modelados.

Pero en la actualidad, no sólo aporta valor diferencial el sistema de información elegido, sino la propia **tecnología** que lo implementa, convertida así en un componente organizacional en la medida en que influencia poderosamente a la organización. La **tecnología** está aportando otra vigorosa influencia como es la especialización, integración y cooperación.

Este proyecto tiene por objeto implementar un servicio software armonizando todos estos dominios: BPM/MGP (Modelo de Gestión por Procesos)⁶, MGS (Modelo de Gestión de Servicios)⁵, Workflow definido con una herramienta BPMS y la elección de una tecnología y entorno de desarrollo como es Spring

¹ La gestión de la tecnología como factor estratégico de la competitividad industrial. **AHN**. Escuela Técnica Superior de Ingenieros Industriales. *Universidad Politécnica de Madrid*

² Sistemas de Gestión de la Calidad. Requisitos ISO 9001:2000, COPANT/ISO 9001:200 NMX-CC-9001-IMNC-2000.

³ Nave, Dave. How To Compare Six Sigma, Lean and the Theory of Constraints. *Quality Progress*, Marzo 002, pp. 73-78.

⁴ Object Management Group, OMG Model Driven Architecture. <http://www.omg.org/mda/>

⁵ Hansen, M., SOA using Java Web Services, Prentice Hall, 2007.

⁶ E. Porter. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*. 1980

MVC con el objetivo de disponer de un apoyo infraestructural a nivel de aplicación evitando el encorsetamiento a entornos de implementación específicos.

3.2. Objetivos

La finalidad del proyecto es desarrollar una aplicación que ofrezca cobertura al proceso de gestión de entregas y despliegue software del modelo de gestión por procesos correspondiente a una Factoría de Software, todo ello según el estándar ITIL v.3 20117.

Para ello, el propósito principal del trabajo es armonizar todos los modelos y metodologías de los dominios implicados, obteniendo una solución segura, estable y óptima, siendo necesaria la definición del proceso (BPM), su automatización a través de un BPMS, el desarrollo de una herramienta web siguiendo el patrón MVC bajo Spring, por último, realizar un estudio de mejora y posible integración con aquellas APIs de mercado que ofrezcan un valor añadido a la solución y la doten de mayor potencia en cuanto a colaboración, cooperación ó seguridad.

Para conseguirlo, se plantean los siguientes objetivos específicos:

1. Seleccionar el Modelo de Gestión de Servicios de Tecnologías de la Información y verificar que se sigue la referencia de esta mejor práctica (ITIL v.3 2011).
2. Evaluar y seleccionar la metodología de Modelo de Gestión por Procesos más apropiada para la definición del mencionado servicio.
3. Identificar las tendencias en cuanto a las mejores suites de BMPS que permitan formalizar el proceso de gestión de entregas y despliegue de software.
4. Desarrollar el entorno de servicios web a través de Spring MVC.
5. Estudiar las APIs de mercado que puedan aportar valor desde el punto de vista de la seguridad y de la colaboración para su posible integración en la solución implementada.
6. Ejecutar la operativa del proceso de gestión de entregas y distribución de software con el servicio software desarrollado.
7. Redactar un informe final valorando las tecnologías utilizadas y la mejora en la gestión con la automatización implementada.

⁷ Desarrollada a finales de 1980, la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL®) se ha convertido en el estándar mundial de facto en la Gestión de Servicios Informáticos.

3.3. Resultados

El resultado principal del trabajo será una memoria detallada con los siguientes contenidos:

- Un escenario de la problemática a resolver en los Proyectos de evolución del Software
 - Incorrecta gestión de requisitos
 - Incorrecto control de cambios
 - Poca implicación del cliente en el desarrollo de la nueva versión ó localización
 - Incorrecta gestión del cambio
 - Inexistencia de certificación de la implantación
- Definición de un proceso BPM en el intervengan todos los roles de una software Factory. Este proceso impedirá desarrollos no normalizados y exige la automatización y control de tareas tales como la gestión de requisitos técnicos, funcionales y de seguridad, control de cambios, alcance invariable de la nueva solución, así como la integración del cliente en fases del desarrollo como pueden ser las pruebas de integración o de pre-explotación. Por último, este proceso propone una certificación final de la solución localizada para el cliente.
- Definición del proceso de desarrollo evolutivo según ITIL v3.2011^{8 9} y bajo la notación BPMN 2.0.
- Diagramación del proceso de desarrollo de la evolución de una versión de un determinado software.
- Arquitectura técnica y requisitos de despliegue
- Propuesta de prestación de servicios software como servicio
- Manual de uso de la herramienta y manual de despliegue de la misma
- Propuestas de mejora para futuros trabajos
- Conclusiones y valoraciones sobre el trabajo realizado

3.4. Metodología

En primer lugar para el estudio del estado del arte acerca de las metodologías, modelos y buenas prácticas en materia de BPM^{10 11}, BPMS y su integración con Spring se ha utilizado una metodología de investigación exploratoria. La herramienta de trabajo fundamental ha sido internet.

⁸ ITIL. The Official Introduction to the ITIL Service Lifecycle - by Office of Government Commerce (OGC) TSO © 2007.

⁹ ITIL Service Design, 2011 Edition by Office of Government Commerce (OGC) -The Stationery Office Copyright Crown Copyright © 2011

¹⁰ IBM Software. Informe sobre liderazgo experto. Once hábitos para el éxito de los programas de BPM. Pasos probados para comenzar con la mejora del proceso.WSW14117-ESE.

¹¹Juan José Camargo Vega, Jorge Enrique Orálora Luna y Aura Beatriz Alvarado Gaona. Revista Ingeniero N 9. Todo alrededor del BPM. Editorial, Desde su origen la tradición intelectual y la científica. ISSN Virtual 2322-8415

Posteriormente para definir los dominios del proyecto se ha recurrido a la metodología métrica versión 3, para enfocar el desarrollo de forma tradicional mediante un enfoque en cascada.

Para la valoración de los riesgos y requisitos de seguridad se ha tenido como referencia la metodología merise.

En el momento de trabajar el estado del arte, se ha recurrido a una metodología explicativa, ya que en esta parte del trabajo se pretenden determinar los orígenes y causas de la situación de las tecnologías actuales.

3.5. Recursos

Para la realización del proyecto se han necesitado los siguientes recursos:

- Recursos Hardware: PC con maquinas virtuales para la simulación del servidor.
- Recursos Software: herramientas de software libre.
 - Framework Desarrollo: Spring MVC
 - BPMS: Activiti, BonitaBPM (en estudio)
 - Herramienta de Modelado UML: OmniGraffle
 - Repositorio de Versionado: BitBucket
 - Herramienta de virtualización: VMware
 - Servidor: Apache
 - Navegador Web: Chrome, Firefox, IExplorer.
 - Apis de extensión: a estudio
 - Suite Ofimática: Microsoft Office
- Recursos personales: un ingeniero de software dedicado 100%.

3.6. Planificación

El proyecto tiene como fecha de inicio el 12 de Enero de 2015 y fecha de finalización el 1 de Junio de 2015.

El coste total del proyecto es de 100 dd/hh. En el siguiente diagrama de Gantt pueden verse las diferentes fases en los que se divide el proyecto y los entregables de cada fase.

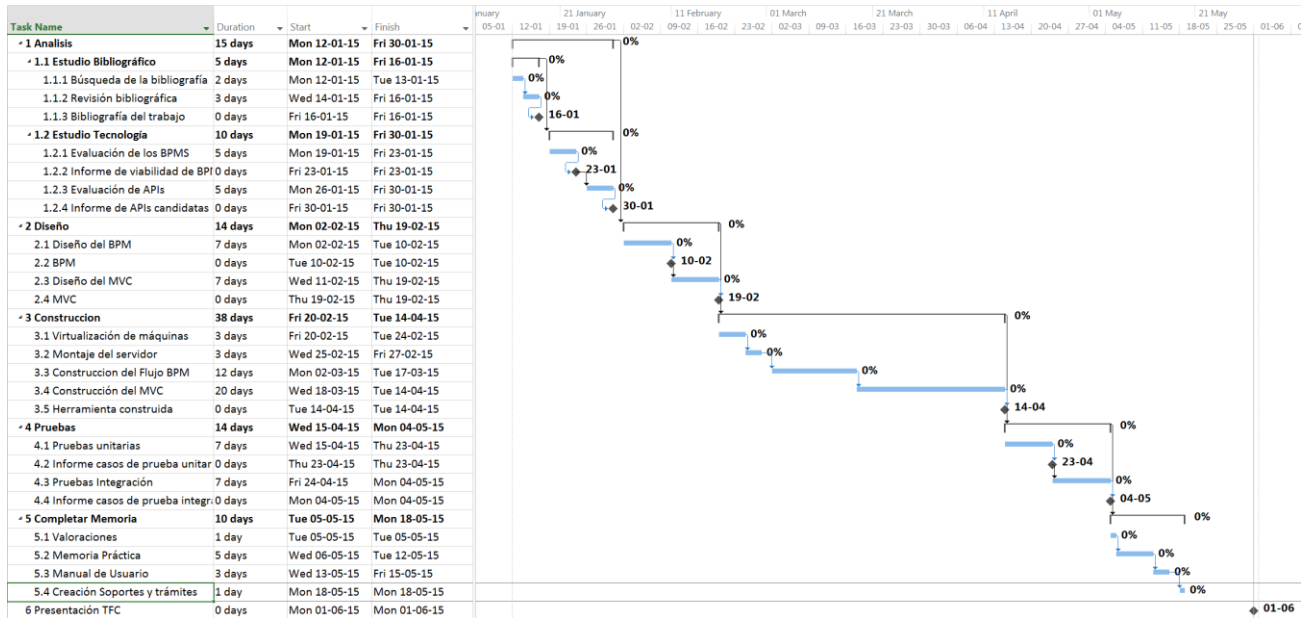


Figura 3-1 Diagrama de Gannt Inicial

CAPÍTULO 4

GLOSARIO DE TÉRMINOS



4. GLOSARIO DE TÉRMINOS Y ABREVIATURAS

4.1 Glosario de Abreviaturas

BBDD:	Bases de Datos
BPM:	Business Process Management
BPMN:	Business Process Model and Notation
BPMI:	Business Process Management Initiative
BPMS:	Business Process Management Software
BPS:	Business Process System
CMMI:	Capability maturity Model Integration
ISO:	International Organization for Standardization
ITIL:	Information Technology Infrastructure Library
NTI:	Norma Técnica de Interoperabilidad
NTS:	Norma Técnica de Seguridad
OMG:	Object Management Group
OPM3:	Organization project Management Maturity Model
SDLC:	System Development Life Cycle
SGC:	Sistema de Gestión de Contenidos
SGD:	Sistema de Gestión de Documentos
SI:	Sistemas de Información
Sw:	Software
TI:	Tecnología de la Información
TIC:	Tecnologías de la Información y la Comunicación
IT mark:	Modelo de calidad internacional para pequeñas y medianas empresas (ESI)
Cobit:	Control Objectives for Information and Related Technology
Spice:	Software process improvement and capability determination (ISo 15504)
ITIL	Information Technology Infrastructure Library
ITSM:	Information Technology Service Management

4.2 Glosario de Términos

BPM / Gestión por Procesos: La Gestión de Procesos (en inglés: Business Process Management o BPM) es una disciplina de gestión, que para mejorar la eficiencia y eficacia, diseña, modela, organiza, documenta y optimiza de forma continua sus procesos de negocio. Este modelo de administración por procesos, apuesta por un cambio operacional en la organización, en la que migra de operaciones funcionales a operaciones administrada por procesos.

BPMN: Business Process Model and Notation (BPMN), en español Modelo y Notación de Procesos de Negocio, es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow).

BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el Object Management Group (OMG), después de la fusión de las dos organizaciones en el año 2005.

Entorno colaborativo: entorno de trabajo o sistema de soporte que permite acceder a un espacio virtual dónde se puede acceder, compartir y gestionar todo tipo de datos, informaciones y documentos, de manera que los miembros o personas implicadas en el workflow al que da soporte puedan, pueden gestionar correctamente las modificaciones y cambios, ya que es posible controlar las versiones de trabajo. Estos entornos, normalmente cuentan con herramientas integradas que permiten una comunicación síncrona y asíncrona, así como la gestión de eventos, alarmas, indicadores de gestión y métricas de soporte, evitando así incurrir en tiempos de coordinación, que son directamente tratados por estas herramientas o tecnologías integrables. Este fenómeno se denomina colaborativo ya que favorece en gran medida la participación cooperativa de diferentes roles, equipos y hasta compañías en el desarrollo de una tarea, trabajo, proyecto o prestación de servicio.

Factoría de Software: Es un centro de trabajo dedicado al desarrollo de Sw que abordaría dicha actividad como un proceso industrial, definible, repetible, y medible en un entorno de alta productividad. Para ello, apuestan por la reutilización de componentes, la creación de líneas de producto, la mejora continua de procesos y los métodos colaborativos y equipos de trabajo distribuidos.

Las líneas de producto tienen por objeto dar un paso más en la búsqueda por aumentar la productividad equilibrando el costo y la calidad del Sw, para ello crean los denominados Core Assets [14].

A lo largo de la historia de la Ingeniería del Software ha aparecido repetidamente el concepto de fábrica software con diferentes matices que se han ido adoptando de acuerdo a la propia evolución

de la tecnología y los procesos software. En la actualidad el término ha vuelto a tomar relevancia en el sector de la industria del software, debido a las especiales condiciones socio-económicas, tecnológicas y de madurez de la ingeniería del software [15].

Desarrollo Evolutivo: modelo de desarrollo Sw conceptualizado en base a desarrollos sucesivos sobre un prototipo. En este modelo los requerimientos son cuidadosamente examinados, y sólo esos son bien comprendidos y seleccionados para primer incremento. Los desarrolladores construyen una versión parcial del producto en base sólo a esos requerimientos

Ingeniería del Software: La aplicación de un enfoque sistemático, disciplinado, y cuantificable al desarrollo, operación, y mantenimiento del software

Mejora continua: concepto del siglo XX que pretende mejorar los productos, servicios y procesos. Actitud general que debe ser la base para asegurar la estabilización de un proceso y la posibilidad de su mejora. Cuando hay crecimiento en una organización es preciso identificar los procesos de su modelo de gestión y producción, realizar un análisis pormenorizado de cada uno de ellos y después de su valoración o auditoria en fase a los modelos de mejora o estándares de calidad, identificar acciones de mejora , de manera que se obtenga una mayor calidad y calidad en los productos y servicios prestados, así como una mayor eficiencia en las organizaciones.

Modelo de Gestión por Procesos [ISO 9001]: el enfoque basado en procesos que propone ISO-9001 reside en la gestión y control de cada una de las interacciones ente los procesos y las jerarquías funcionales de la organización.

Los procesos deben servir para aportar valor a una organización, además deben estar alineados con los objetivos, alcance y grado de complejidad de la organización [12]

OMG: The c® (OMG®) is internacional y el número de miembros que lo componen es abierto. Fue fundada en 1989, y los OMG estándares son dirigidos por vendedores, usuarios, entidades académicas, gobiernos, etc. Las tareas y esfuerzos de OMG van encaminadas hacia la integración de estándares para una amplia gama de tecnologías y una gama aún más amplia de industrias [17].

Procedimiento [ISO 9001]: Según la norma ISO 9000, un procedimiento es una forma específica para llevar a cabo una actividad o un proceso. Cuando se tiene un proceso que tiene que ocurrir en una forma específica, y se especifica cómo sucede, se tiene un procedimiento. Un ejemplo de procedimiento puede ser un proceso de revisión de un contrato que puede que haya establecido con uno de sus clientes para dar la orden de compra, existe un conjunto definido de medias para revisar, aprobar y aceptar dicha orden, además la orden se registra y se distribuye de cierta forma en sus trabajadores.

Es muy importante tener en cuenta que no todos los procesos necesitan contar con un procedimiento. Si cuenta con un proceso con que el que sólo se compra producto de un proveedor autoriza, pero no tiene una forma definida para agregar un proveedor a dicha lista, entonces tiene un proceso pero no un procedimiento.

(*) En el desarrollo de software que no se tenga una forma correcta de insertar un código, y que los desarrolladores de software escriban el código de forma diferente cada vez, significa que no están siguiendo un procedimiento.

Proceso [ISO 9001]: un proceso es un conjunto de actividades que están interrelacionadas y que pueden interactuar entre sí. Estas actividades transforman los elementos de entrada en resultados, para ello es esencial la asignación de recursos. [12]

Workflow: diagrama que refleja la operativa de un trabajo, la organización de las tareas, su orden correlativo, sincronización, flujo de información que da soporte a dichas tareas y la manera en que se realiza el seguimiento y certificación de realización de las mismas. Normalmente los flujos de trabajo se modelan con redes de Petri.

A pesar de que un workflow se relaciona con un modelo de gestión por procesos, un proceso o un diagrama de actividades, el concepto de workflow se encuentra íntimamente relacionado con las tecnologías de la información, concretam

ente con los sistemas de información. Esta parte es especialmente importante para software de trabajo colaborativo, un paso hacia delante de los tradicionales sistemas de información ó de gestión, cuyo objetivo era soportar la automatización de tareas de estos procesos o workflow de trabajo [16].

CAPÍTULO 5

MEMORIA



5. MEMORIA

5.1. Introducción

5.1.1. Escenarios e hipótesis de partida

En el presente trabajo se van a mostrar las problemáticas a las que se enfrenta un proceso de Desarrollo Sw dentro de una Factoría especializada en el desarrollo de soluciones informáticas, y cómo con la implementación de un workflow de trabajo automatizado, y a través del uso intensivo de herramientas y de tecnologías de colaboración, aumenta la productividad y la calidad del producto o servicio software.

El escenario planteado es una Factoría de software ya que los procesos de desarrollo implementados en la misma, son procesos necesariamente “core”, y regularmente están basados en modelos de madurez y buenas prácticas como Cobit o ITIL, por lo que el objetivo de mejorar la productividad y la calidad del producto, es un objetivo primordial en estas compañías. Por otra parte, las Factorías de Software apuestan por el cumplimiento de los estándares y modelos de buenas prácticas más innovadores, dado que la calidad del servicio obtenido en términos de seguridad, integrabilidad, cumplimiento normativo, y escalabilidad es fundamental para su hegemonía comercial.

Como punto de partida y a modo de introducción, se presenta un breve recorrido de las hipótesis de trabajo con las que se parte para proponer una solución que las resuelva o mitigue. Esta solución que se va a exponer consta de varias vertientes:

1. Vertiente normativa. Se apuesta por el cumplimiento y seguimiento de las pautas de las mejores y más actuales “best practices” de referencia.
2. Vertiente organizativa. El escenario se plantea dentro de una Factoría de Sw, ya que es dónde aparecen las figuras profesionales más especializadas en materia de desarrollo Sw.
3. Vertiente de proceso: se apuesta porque la Factoría de Sw realice su desarrollo en base a un modelo de gestión por procesos, ya que este modelo permite que el desarrollo en este caso evolutivo del sea repetible, mejorable y evolucionable.
4. Vertiente tecnológica: la solución vendrá soportada por tecnologías de integración y herramientas colaborativas que mejoran la productividad y calidad.
5. Vertiente de calidad: mejora continua como base para ir hacia la excelencia en todos los procesos de la Factoría.

Cabe preguntarse si la eficiencia de una Factoría de Software se relaciona directamente con la evolución de sus prácticas en Ingeniería del Software, y si realmente la mejora continua de sus procesos la está haciendo más eficiente. Por otra parte, otra de las hipótesis de trabajo es la de preguntarse y valorar si la tecnología que se va a emplear como soporte al proceso de desarrollo evolutivo, tendrá mucho que ver con la efectividad resultante.

Tradicionalmente, eran los sistemas de información de soporte lo que realmente diferenciaba a las compañías, y en muchos casos, eran fuente indudable de éxito, pero en este momento, la hipótesis de trabajo es la de evaluar si la propia tecnología elegida, en sí misma, puede llegar a ser fuente de éxito.

La mayoría de las empresas que ofrecen servicios de tecnología más y mejor introducidos en el mercado, tienen sus factorías deslocalizadas y los equipos de trabajo distribuidos geográficamente. Esto, junto al hecho de que una Factoría de sw es un centro de trabajo especializado en desarrollo sw y que lo aborda como un proceso industrial, definible, repetible y medible en un entorno de alta productividad, hace que parezca el mejor de los casos para probar la automatización de un proceso a través de tecnologías colaborativas y de integración.

El desarrollo evolutivo se puede definir como aquel que pretende actualizar y modificar funcionalidades de un sw que eran correctas, con el objeto de aumentar o cambiar las funcionalidades del sistema, bien sea por necesidades del usuario, para el aumento de rendimiento, por necesidades normativas, o por mejoras del proceso al que ofrecen cobertura. Normalmente, una percepción común del proceso de mantenimiento sw, es que éste se centra sólo en la corrección de errores. Sin embargo, un estudio indicó que el 80% del tiempo se dedicaba a la implementación de mejoras [16]. Por este motivo, este trabajo va a trabajar sobre el subproceso de desarrollo evolutivo, ya que se trata de la parte del mantenimiento más crítica para la mejora de la productividad.



Figura 5-1 Tipologías de Mantenimientos Software

El proceso al que tiene que ofrecer cobertura la solución que se proponga, es el proceso de desarrollo evolutivo y en la actualidad, este proceso es depositario de problemáticas bien conocidas en el mundo de la ingeniería. Estas problemáticas son las siguientes:

- Existencia de requerimientos iguales
- Existencia de requerimientos incompletos o incorrectos
- Perdurabilidad y no cambio de los requerimientos para que el alcance y la estimación no cambie
- Saber qué código pertenece a cada requerimiento.
- Cuáles son los casos de prueba asociados
- Si el código de partida se encuentra alineado a los estándares de la Factoría
- Difícil gestión de configuración
- Problemática en la gestión de cambios
- Dificultad en la certificación del despliegue
- Difícil certificación de la implantación de la nueva versión localizada
- Falta de gestión de riesgos
- Inexistencia de políticas o procesos estables software.
- Coste que requiere la organización de la localización
- El coste de la organización de la versión / localización / evolutivo
 - o Actividades que se derivan de la organización
 - o Identificar y agrupar las funciones, actividades y tareas del proyecto.
 - o Seleccionar estructuras organizacionales
 - o Crear posiciones organizacionales
 - o Definir responsabilidades y autoridades.
 - o Tareas de gestión del Proyecto (documentación de decisiones, gestión de costes, informes de avance, informes de gestión de riesgos, etc.)

Por todo lo expuesto, dentro de las factorías de sw, la definición de los procesos que implican el desarrollo sw incluido el proceso de desarrollo evolutivo, se fundamenta en algún modelo o buena práctica de referencia como puede ser CMMI, OPM3, IT mark, Cobit, Spice o ITIL, de manera que la gestión por procesos permite el diseño, modelado, organización, documentación y optimización continua. Esta evolución, cambia la manera de trabajar de una organización, pasando de trabajar con tareas funcionales a trabajar con operaciones administradas por procesos. Este hecho, nos coloca en un excelente escenario para el desarrollo de una solución de soporte, que es el objetivo de este trabajo.

5.1.2. Objetivos para el proyecto y propuesta de solución

Dado que el desarrollo sw que se realiza en la Factoría se desarrolla en base a un modelo de gestión por procesos bajo el estándar de referencia ITIL, se tiene como marco de trabajo la estrategia de mejora continua, dentro la cual se enmarcaría el proyecto de implantación de esta herramienta que mejora la gestión de la entrega y el despliegue de soluciones sw entre sus clientes. El objetivo es flujogramar e implantar un workflow (BPM) de soporte al desarrollo evolutivo mediante el uso de la tecnología Acitiviti.

Puesto que es indudable la actual necesidad de disponer de entornos colaborativos y de la capacidad de la tecnología de impulsar este factor, se elige la tecnología de Activiti, puesto que es una tecnología que facilita la colaboración en torno al proceso de negocio que se implementa.

Activiti cuenta con elementos que facilitan la interacción entre los diferentes roles de trabajo implicados (analistas, directivos y desarrolladores), y su motor permite importar ficheros XML BPNM2.0 con una baja carga en cuanto al uso de memoria y almacenamiento en BBDD. Por otra parte, Activiti incorpora una API de Java que permite definir tareas que se pueden modelar gráficamente y ejecutarlas mediante este motor, de manera que la actividad personalizada, puede ser reutilizada y reconfigurada a través de un sencillo interfaz.

5.2. Problemática planteada

Dentro del presente escenario y del modelo de procesos con los que cuenta la Factoría de Sw (Estrategia, Diseño, Transición, Operación y Mejora continua), centrándonos en el proceso de mejora continua y tras el análisis, métrica, y evaluación del proceso, se ha detectado un área de mejora que es el perfeccionamiento de la comunicación entre los diferentes participantes implicados, pensando tanto en el personal de la factoría como en el propio cliente. La hipótesis nos lleva a pensar que actualmente existen comunicaciones no formalizadas a través de medios tales como el telefónico, o mediante correo electrónico, o en base a reuniones dónde están parcialmente representados todos los stakeholders.

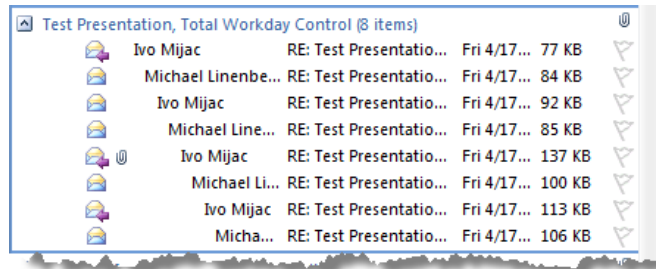


Figura 5-2 Ejemplo de Hilo de conversación sobre unas pruebas

La falta de claridad en la comunicación, la confusión en los acuerdos tomados al no contar con un canal oficial, la falta de información en los equipos de trabajo, el acaparamiento de información por parte de algunos perfiles, o la falta de actualización y control de versiones de los acuerdos tomados, son entre otros, los muchos inconvenientes que surgen. Por lo que uno de los objetivos a conseguir con el BPM es contar con información ordenada, trazable y reglada, para que se disponga de procesos horizontales ágiles y efectivos que integren la información dispersa en diferentes SIs, y provoque a la participación intensiva de todos los implicados.

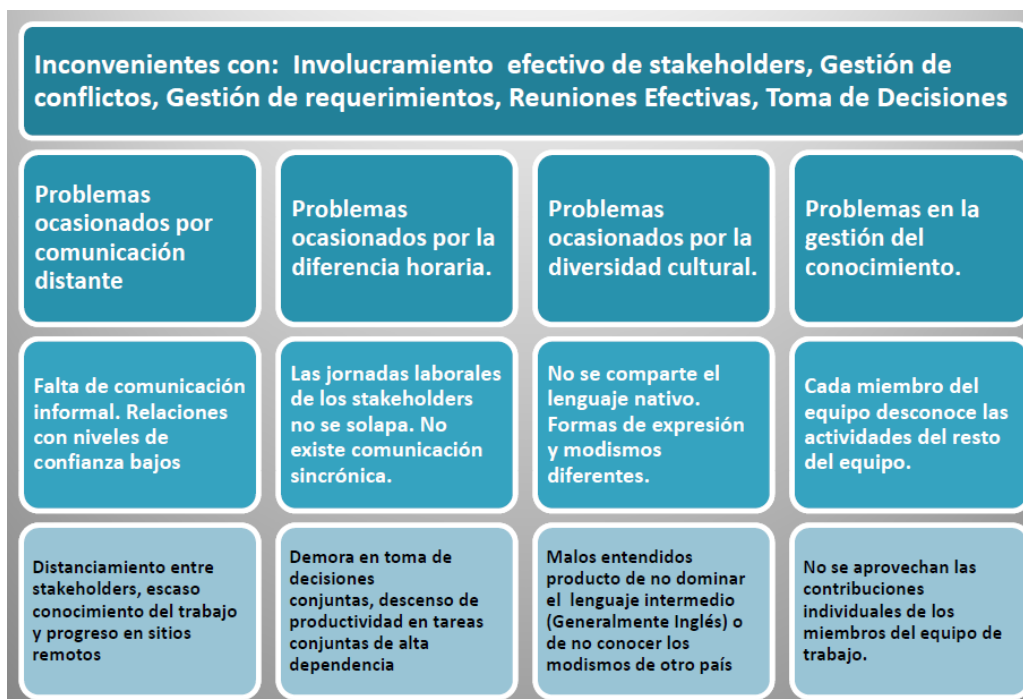


Figura 5-3 Problemáticas de las factorías de sw que tienen muchos stakeholders

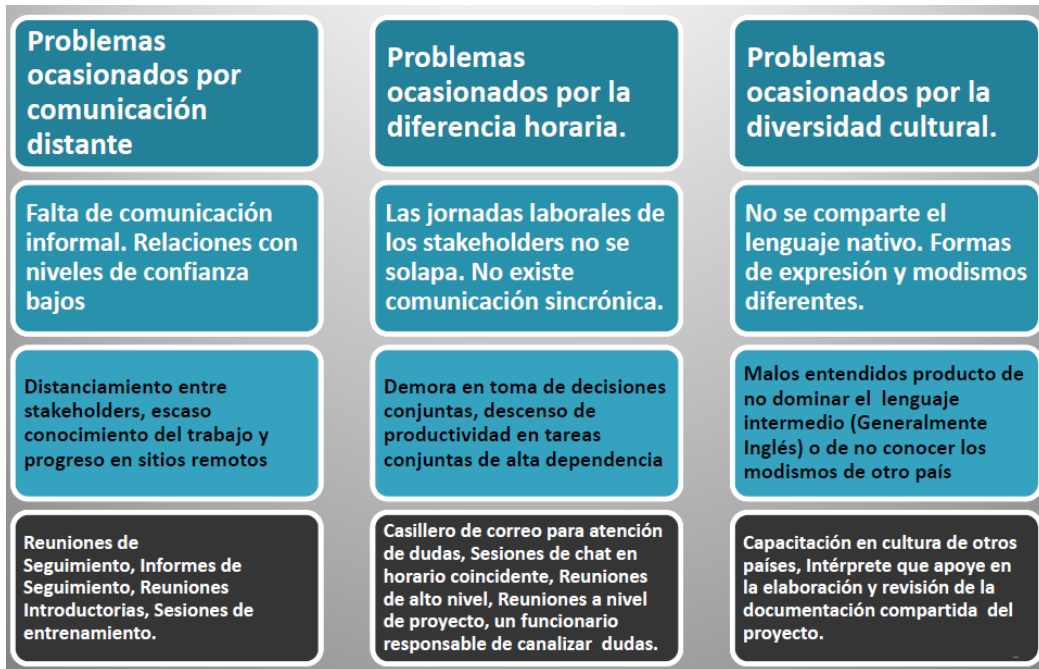


Figura 5-4 Problemáticas de las factorías de sw que tienen muchos stakeholders

5.3. Estado del arte

En este punto se va a realizar una presentación de porqué surgen las factorías de sw, y cómo van evolucionando desde su creación hasta la actualidad que se han convertido en auténticos proveedores de servicios sw. A continuación se realiza una breve introducción al modelo de buenas prácticas ITIL, para posteriormente enmarcar bajo su prisma, la acción de mejora propuesta mediante la herramienta BPM.

A continuación, se va explicar qué es un BPM, la notación BPMN empleada para la diagramación de procesos, así como de la unión de las tres líneas ITIL-FS-BPM surge el “Desafío de Diseño” de la herramienta que se va a construir.

5.3.1. Las Factorías de Software.

La fabricación de software es una de las principales actividades económicas. El concepto de fábrica de software fue mencionado por primera vez en 1968 por Bemer mencionando entornos específicos para la producción económica de programas...”. Un año después, se lanzaba la primera fábrica de software por Hitachi Software Works. Desde entonces, el término fábrica de software denota industrializar el desarrollo software para reducir costes y mejorar su calidad [18].

Actualmente el término vuelve a tener relevancia dada la actual importancia de la tecnología en todos los ámbitos (educativo, turístico, sanitario, cultural, etc.) y de madurez de la ingeniería del software [19, 20].

Evolución de las Factorías de Sw

1.- Años 80: origen de las fábricas de software. Pensando que es imposible que los programadores bajo la supervisión humana, se forja el término ya que en una fábrica se mide y controla la productividad y calidad, se mantienen registros financieros y hay planificación.

2. Años 90: CASE, reutilización y procesos. Durante los noventa surgen diferentes aproximaciones a las fábricas de software.

- a) **basadas en Entornos de Desarrollo Integrados.** Se implantaron las primeras generaciones de herramientas CASE, y los denominados Entornos Integrados de Desarrollo de Software (conocidos por sus siglas inglesas ISDE, *Integrated Software Development Environments*), y los Entornos de Ingeniería del Software orientados al Proceso (*PSEE, Process-centered Software Engineering Environment*). Lo crean grandes empresas europeas, fabricantes de ordenadores, desarrolladoras de software y universidades.
- b) **basadas en experiencia.** Surgen con tres objetivo: mejorar la eficacia del proceso, reducir la cantidad de re-proceso y reutilizar los productos de ciclo de vida.
- c) **basada en la madurez de procesos.** El objetivo es crear un marco para la mejora de procesos sw que permitan conseguir un proceso predecible, fiable y mejorable [21].
- d) **fábrica de software basada en la reutilización.** se apuesta por una reutilización efectiva creando bibliotecas para el código, y se combina la idea de sistemas de fabricación flexible con el ensamblado de componentes y productos.
- e) **fábricas de renovación de software.** se trata de una efecto estela del “año 2000” y de la llegada del euro. En estas fábricas trabajan en líneas de ensamblado (transformadores de código, re-generadores, re-estructuradores, migradores, etc.)
- f) **fábricas enfocadas a la calidad:** destacan por la aplicación de TQM (Gestión de Calidad Total) y reutilización, así como generadores de código, buscando la flexibilidad. [22].

3. Años 2000: COMPONENTES, MODELOS y LÍNEAS DE PRODUCTOS

A partir del año 2000 las fábricas desarrollan en base a la ingeniería basada en modelos, el desarrollo basado en componentes, las líneas de productos y servicios y los modelos de madurez de procesos. Se combina, la gestión de la calidad orientada a procesos, con las líneas de producto basadas en tecnologías de componentes.



Figura 5-5. Evolución de las factorías de Sw [14].

De este diagrama se deduce que en la actualidad la apuesta es la de la integración global, la cooperación y el uso intensivo de tecnologías de comunicación síncrona y asíncrona. La entrada de las TI en nuestras vidas ha ocasionado una auténtica revolución, y ha cambiado la relación de causalidad, espacio y tiempo, y a pesar de que el arranque ha sido desde el espacio de ocio, se ha producido una auténtica revolución dentro de las compañías con el uso de herramientas de comunicaciones unificadas, sistemas de información geográfica, etc., lo cual ha llegado también a los procesos de fabricación y a las compañías más tradicionales.

5.3.2. Marco de referencia ITIL

Dentro de las compañías dedicadas al desarrollo sw, los modelos de gestión por procesos se encuentran definidos en base a modelos de madurez y referencias de buenas prácticas. Uno de los modelos más extendidos es ITIL, que se constituye como una librería o conjunto de buenas prácticas que guían a la organización a la consecución de objetivos de forma ordenada.

La Biblioteca de Infraestructura de Tecnologías de Información (ITIL) son un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las gestión de infraestructura, desarrollo y operaciones de TI [8].

La definición oficial de ITIL es: “Un conjunto de publicaciones de mejores prácticas para Gestión de servicios de TI. ITIL proporciona asesoramiento sobre la provisión de servicios de TI de calidad y de los procesos, funciones y demás capacidades necesarias para darles apoyo. El marco de ITIL está basado en un ciclo de vida del servicio y consiste de cinco etapas (estrategia del servicio, diseño del servicio, transición del servicio, operación del servicio y mejora continua del servicio) que cuentan con su propia publicación de apoyo. También hay un conjunto de publicaciones complementarias de ITIL que brindan asesoramiento específico para distintos sectores económicos, tipos de organizaciones, modelos de operación y arquitectura de tecnología”.



Figura 5-6 Modelo Conceptual ITIL.

A priori puede parecer que está reñida con otros enfoques como puede ser Agile, lo cierto es que ambos pueden ser utilizadas como parte de un todo mayor. Lo importante es conocer los enfoques y aportar lo mejor de los dos mundos.

En el escenario que nos atañe es importante ver la creación de software para un cliente como un servicio, así puede entenderse la entrega y despliegue de ese software como un proceso. En este sentido, todo proceso debe incluir actividades de control y medición.

5.3.2.1. Mapa conceptual ITIL

ITIL estructura la gestión de los servicios TI sobre el concepto de Ciclo de Vida de los Servicios. Este enfoque tiene como objetivo ofrecer una visión global de la vida de un servicio.

ITIL v3 reestructura el manejo de los temas para consolidar el modelo de "Ciclo de Vida del Servicio" separando y ampliando algunos subprocesos hasta convertirlos en procesos especializados. El Ciclo de Vida del Servicio consta de cinco fases ó disciplinas, correspondientes a los nuevos libros de ITIL:

1. Estrategia del Servicio
2. Diseño del Servicio
3. Transición del Servicio
4. Operación del Servicio
5. Mejora Continua del Servicio

5.3.2.2. Proceso ITIL. Mejora Continua del Servicio

En este proceso se utilizan herramientas de medición y feedback para documentar la información referente al funcionamiento del servicio, los resultados, problemas ocasionados, soluciones tomadas, etc. Es en este proceso de donde se obtiene la problemática expuesta en referencia a la falta de comunicación y coordinación de los equipos intervinientes en el proceso de desarrollo sw evolutivo [ver punto 5.1.2. Objetivos para el proyecto y propuesta de solución].

En este momento es cuando se reflexiona acerca de que "Toda mejora es un cambio, pero no todo cambio implica una mejora" por lo que este proceso de mejora continua monitorizará todos los cambios propuestos y soportados por la herramienta.

5.3.3. Herramientas BPM y su soporte a la prestación de servicios

5.3.3.1. Diagrama de Procesos de una Factoría.

A continuación se presenta el modelo de prestación de servicios entre los que se encuentra el desarrollo de sw evolutivo

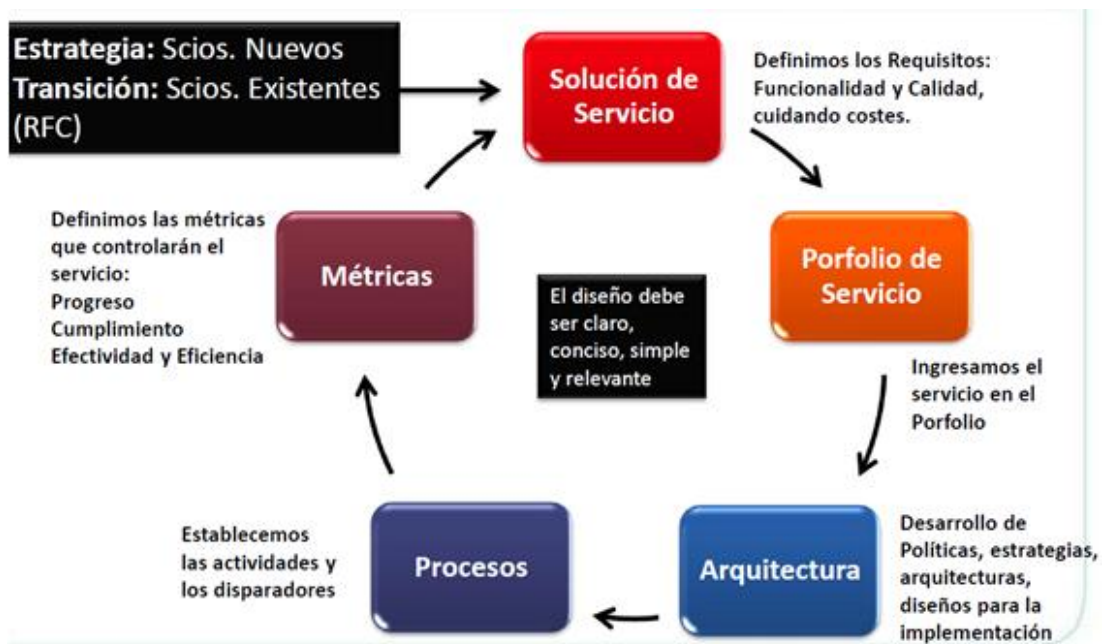


Figura 5-7 . Modelo de prestación de servicios.

El uso de esta herramienta creará una red de valor entre la Factoría de Software y el cliente. La garantía de un buen servicio en la realización de la entrega y despliegue, va a estar avalada por la buena comunicación y cooperación de todos los participantes involucrados, la normalización del proceso y su automatización, que exige que el mismo sea siempre realizado de la misma forma. En este punto, cabe destacar que una gestión efectiva de la transición puede mejorar significativamente la capacidad de un proveedor para gestionar un alto número de cambios y entregas

Los subprocesos a abordar en la transición de un servicio son los siguientes:

1. Gestión de Cambios

Asegurar que los cambios se despliegan de forma controlada.

2. Gestión de Activos y Configuración del Servicio

Gestiona los activos para dar apoyo a los demás procesos.

3. Gestión del Conocimiento

Mejora la calidad de las decisiones mejorando la disponibilidad de información confiable.

4. Planificación y Apoyo a la Transición

Planificación y coordinación de los recursos que cumplan con las especificaciones del Diseño del Servicio.

5. Gestión de Versiones y Despliegues

Construye, prueba y despliega los Servicios especificados en el Diseño del Servicio.

6. Validación y Prueba del Servicio

Asegurarse de que el nuevo Servicio o el Cambio cumple con el objetivo y con las necesidades de uso.

7. Evaluación del Cambio

Proceso genérico que asegura que el rendimiento es aceptable.

En la actualidad, existen en el mercado una considerable oferta de suites BPM. Algunas de ellas como servicios online, otras integradas dentro de herramientas, algunas gratuitas, otras de pago, etc. De todas ellas destacan por encima del resto por diferentes consideraciones, las siguientes: BonitaBPM, JBoss BPM y Activiti. En cualquier caso, en primer lugar se analiza qué es, para qué sirve y cuál es su notación antes de la presentación de la herramienta elegida.

5.3.3.2. Herramientas BPM

Podría definirse como una nueva categoría de sw que permite a las empresas modelizar, implementar y ejecutar tareas y actividades dependientes e interrelacionadas de cualquier naturaleza. Estas herramientas poseen extensiones para incluir a clientes, proveedores y otros agentes en dichas tareas.

Aunque los procesos diseñados dentro del BPM pueden trabajar integradamente con sw de gestión tal como un ERP ó un CRM, deben ser previamente modelizados y unificados.

Las características que suelen cumplir son la siguientes: 1. Diagramador de procesos BMP: permite crear mapas de flujo; 2. Generador de formularios: personalizados para los procesos de la organización. 3. Constructor de Documentos: Es posible crear documentos imprimibles con ésta herramienta. 4. Motor de Reglas: permite definir de manera sencilla la lógica del negocio detrás de cada proceso. 5. Debugger: permite seguir paso por paso el avance de un proceso formulado.

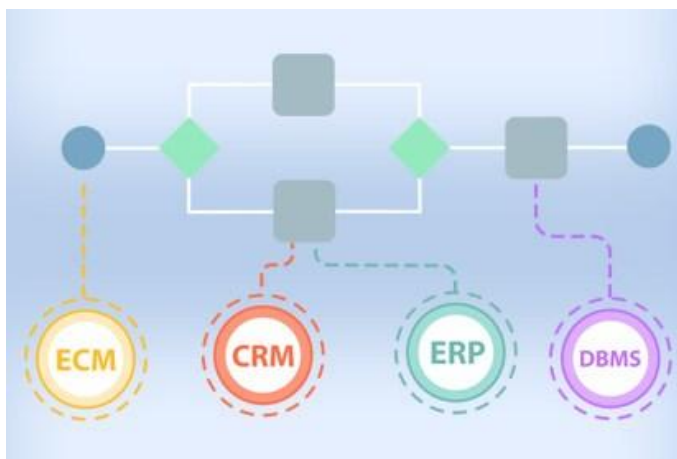


Figura 5-8 BPM integrado con otras herramientas de gestión empresarial

5.3.3.3. Notación BPMN

La notación Business Process Model and Notation (BPMN), en castellano Modelo y Notación de Procesos de Negocio, es una notación gráfica estandarizada que permite modelar procesos de negocio en un formato de flujo de trabajo (workflow). Su versión actual, es la v2.0.2 (2013)

El principal objetivo de este modelo, es ofrecer una notación gráfica estándar fácilmente legible y entendible por parte de todos los stakeholders. En síntesis, el modelo de notación BPMN pretende eliminar la brecha de comunicación existente entre los diseñadores de procesos de negocio y los responsables de su implementación técnica, quienes deberán ofrecer soporte a todas aquellas tareas susceptibles de ser automatizadas.

Actualmente hay una amplia variedad de lenguajes, herramientas y metodologías para el modelado de procesos de negocio, pero sin lugar a dudas, la adopción cada vez mayor de la notación BPMN como estándar, ha unificado la expresión de conceptos básicos de procesos de negocio. La consolidación de este modelo también viene condicionado al hecho de que la gestión por procesos se confirma como uno de los mejores sistemas de organización empresarial para la mejora de la calidad, la productividad y la excelencia. Además, actualmente, en un contexto empresarial, mundial y económico tan globalizado y competitivo, la gestión por procesos se ha convertido en una auténtica necesidad, ya que en muchos casos depende de ello la propia subsistencia de la compañía.

El modelado en BPMN se formula mediante diagramas sencillos de elementos gráficos básicos. Las cuatro categorías básicas de elementos son [24]:

- Objetos de Flujo: Eventos, Actividades, Rombos de control de flujo (gateways);
- Objetos de Conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación;

- Carriles de nado (swimlanes): Piscina, Carril;
- Artefactos: Objetos de Datos, Grupo, Anotación.

5.3.3.4. Principales BPM de mercado

De todas ellas hay tres que destacan por encima del resto por diferentes consideraciones. Por eso, en este apartado se presentan las tres herramientas más representativas en el panorama actual y después de realizar la descripción de cada una de ellas se presentará la elegida para el presente trabajo.

BonitaBPM

El desarrollo de Bonita comenzó en Francia, en la National Institute for Research in Computer Science (INRIA). Más tarde, su proceso de incubación fue en una compañía francesa como es Bull (Bull es una sociedad francesa especializada en la informática profesional). En la actualidad el desarrollo pertenece a la empresa BonitaSoft (empresa multinacional con sede en Francia y con oficinas en EEUU (New York y San Francisco)).

Entre sus casos de éxito destacan empresas soluciones para empresas tan importantes como Carrefour, BBVA, Sony, etc.

Actualmente es la herramienta más extendida en el mercado, dado que tiene una licencia de código abierto, y proporciona a sus usuarios una experiencia de inicio a fin a la hora de crear aplicaciones basadas en procesos, sin la necesidad de tener que escribir ni una sola línea de código. Actualmente, soporta el lenguaje de procesos BPMN 2.0.

BonitaBPM, es con diferencia la solución BPM con más cantidad de desarrollos de terceros. Tiene un ecosistema tal, que es prácticamente imposible no encontrar un módulo que conecte con una determinada herramienta de su suite, lo que la convierte en una solución muy flexible para desarrollos rápidos.

En la actualidad BonitaBPM se encuentra en la versión 7.5.4 lanzada a principios de 2017. La plataforma tiene dos versiones:

Community Edition:

- Versión gratuita soportada por la comunidad.

Subscription Edition:

- Contiene soporte profesional global.
- Herramientas de personalización de interfaces
- Interfaces específicas para móviles y tablets.
- Cuadro de mando y monitorización.
- Posibilidad de clusterización y balanceo de carga

Por el contrario, a todas estas ventajas, puesto que esta herramienta crea el código de forma muy automática, no se tiene un gran control sobre él, y es difícil modificarlo fuera de su entorno y herramientas. Uno de los problemas que presenta, es que la integración con lenguajes como Java está mucho menos desarrollada que otros BPM.

Red Hat JBoss BPM (jBPM)

Esta herramienta, es considerada por muchos, el padre de los BPMs de código abierto. El desarrollo de este BPM fue liderado por Tom Baeyens en la empresa JBoss.

Aunque nace como un proyecto con entidad propia. Su ciclo de vida ha estado completamente ligada al gestor documental Alfresco (motor de tareas que ejecutaba esta herramienta para realizar tareas de todo tipo: versionado, interacciones, etc.).

Entre sus clientes como casos de éxito se encuentran: Amadeus, Casio, Cisco, Adobe etc. En la actualidad la empresa que mantiene la herramienta es Red Hat. Esta se encuentra en la versión 6.4 lanzada en Noviembre de 2016 y al igual que BonitaBPM tiene dos versiones pero con diferentes matices:

Gratuita: permite realizar trabajos de desarrollo y no se puede utilizar para usos en producción.

De pago: ofrece soporte de RedHat y autorización para desplegar la herramienta en producción.

Se distribuye bajo licencia Apache. Se trata de un motor escrito en Java

Actualmente jBPM se basa en la Máquina Virtual de Procesos (PVM), que es el fundamento de la comunidad JBoss para soportar múltiples lenguajes de proceso de forma nativa. La comunidad JBoss actualmente se centra en el uso de la especificación BPMN 2.0 para la definición de los procesos de negocio.

Activiti

En mayo de 2010, desde la empresa Alfresco se funda Activiti. El objetivo principal de esta empresa era crear un nuevo motor BPM escrito en Java que utilizase la nueva notación estándar BPMN2. Esta tarea fue encomendada a las dos personas más importantes de jBPM, Tom Baeyens and Joram Barrez, quienes empezaron a escribir el código de la primera versión en mayo de 2010 y en diciembre de ese mismo año se liberó.

Dentro de la comunidad de Activiti se incluyen compañías como SpringSource, FuseSource, Mulesoft, lo que han llevado a este motor a un nivel superior de integración en el ecosistema Java, siendo una de sus principales valías la integración de forma nativa con la suite de Spring.

En la actualidad este es el motor oficial que utiliza Alfresco. Como sucedía hace algunos años con jBPM, si bien es un proyecto independiente del propio Alfresco, su ciclo de vida y lanzamientos están muy ligados.

Frente a competidores como Bonita, tiene la debilidad de no existir una gran variedad de módulos de integración con software de terceros. Por el contrario, al ser escrito y consumido principalmente en Java, se puede implementar cualquier funcionalidad que se ocurra y tener el control absoluto sobre ella.

El líder actual de este proyecto es Mauricio Salatino, que ha llevado a esta herramienta hasta su versión 6.12 y entre sus casos de éxito (que son los de Alfresco) tiene clientes como Cisco, Nasa, Oxford University, etc.

Un nuevo BPM entra en la escena. FlowAble

En octubre de 2016, se produce una noticia inquietante en el seno de Activiti. Las divergencias de opinión en relación hacia dónde debe dirigirse el proyecto, hace que uno de sus líderes actuales Tijs Rademakers abandone el proyecto original, desligando su carrera profesional de Alfresco y realizando un fork al proyecto bajo el nombre de “FlowAble”. Uno meses después, la otra alma del motor de procesos de Activiti, Joram Barrez también se une al de FlowAble. Al producirse este acontecimiento la versión de Activiti, se encontraba en la versión 5.xx, y unos meses después aparece la versión 6.0 con nuevas funcionalidades.

Una de las principales divergencias que produjeron la ruptura vino producida por el sistema de soporte de Activiti y la liberación de versiones. Mientras que los creadores originales querían seguir manteniendo el alma de software libre y colaborativo que definieron bajo la licencia Apache. Desde Alfresco, aun siguiendo este tipo de licencia, quieren que el soporte a Activiti se monetice. Aunque esta información fue publicada en varios foros de internet, al escribir estas líneas no se ha producido ningún movimiento por parte de Activiti para crear un soporte de pago.

En la actualidad ambos softwares son casi gemelos. Incluso los ejemplos de inicio de la herramienta son los mismos. Pero sin duda habrá que estar atentos a FlowAble porque seguro que a futuro contendrá nuevas y útiles funcionalidades.

La elección: Activiti

Actualmente Activiti es un proyecto con un volumen importante de impulso y desarrollo. A medida que avanza el desarrollo, cada vez es más fácil implementar procesos y trabajar con las diferentes herramientas y componentes.

Activiti es un motor ligero, fácil de usar (para desarrolladores de Java) e intenta facilitar la colaboración al rededor del proceso de negocio.

Tradicionalmente, el desarrollo de los procesos de negocio es una tarea de “analistas”. Esta función consiste en dibujar diagramas de procesos, definir el flujo de las tareas, definir a qué rol se asigna una tarea, y qué pasos se pueden automatizar. Hecho lo anterior, se desarrolla “código ejecutable” como soporte a las actividades susceptibles de automatización. Activiti tiene una serie de elementos que facilitan la colaboración entre los diferentes roles involucrados (Activiti Engine; Activiti Explorer; Activiti Modeler; Activiti Designer).

En primer lugar, Activiti utiliza BPMN 2.0, estándar abierto ampliamente aceptado para modelado de procesos. Lo bonito de esta versión 2.0 del estándar es que, más allá del lenguaje, el código XML subyacente también define un estándar y lo puede ejecutar directamente (este XML). Activiti también proporciona un módulo adicional donde se facilita la colaboración de los roles involucrados al estilo ‘2 .0’, mejora el proceso de desarrollo de los procesos de negocio, y mejorar las definiciones de proceso actuales. Daré más detalles sobre esto más adelante en este trabajo.

5.4. Diseño de la Solución

5.4.1. Diseño Conceptual de la Solución

5.4.1.1. Juntándolo todo. Desafío de Diseño

Una vez presentada la problemática en el capítulo anterior, se pretende crear una solución que palie las deficiencias encontradas y así optimizar el servicio de la Factoría de Software.

El desafío de diseño queda definido en la siguiente frase:

“¿Cómo podemos homogenizar las comunicaciones entre todos los participantes del evolutivo y automatizarlas lo mayormente posible?”

La respuesta es la creación de una herramienta que de soporte a estas comunicaciones. Creación de punto de encuentro donde cualquier participante del sistema pueda comprobar qué tareas tiene asignadas, cuáles ha completado, en qué estado se encuentra el evolutivo en los que tiene competencias, etc.

Sería deseable que este sistema también enviase de forma automática mensajes cuando se produjeran incidentes en algún punto del proceso.

Esta herramienta vendría a compaginar (no a sustituir) las posibles herramientas para la gestión del ciclo de vida que pueda estar usando tanto la factoría como el cliente. Con la creación de esta herramienta se quiere eliminar el “gap” existente en estas comunicaciones. Se tratará de una aplicación web basada en procesos, por tanto utilizará una herramienta BPM para el modelado y ejecución de esos procesos.

Esta aplicación deberá tener una interfaz sencilla y clara donde la auto descripción de la información y de los componentes haga de ella una herramienta intuitiva y cómoda.

Los campos a informar en cada una de las fases deben ser el menor número posible.

La interface con los usuarios será de tipo Web para que sea más fácil su uso des localizado. Deberá ser lo suficientemente flexible para adaptarse a nuevos flujos de trabajo, con nuevas tareas o participantes.

























Con todo esto se pretende minimizar el impacto del cambio durante el servicio.


El uso de esta herramienta creará una red de valor entre la Factoría de Software y sus clientes. Siendo un valor añadido al servicio prestado.


Con ello se pretende tener la garantía fehaciente de que antes de la realización de la entrega y despliegue, todos los participantes involucrados han dado su conformidad.


5.4.1.2. Elección de herramienta BPM

Tras la presentación de los BPMs más relevantes realizados en el capítulo 5.2, el siguiente cuadro muestra una comparativa de los tres: BonitaBPM, jBPM y Activiti. Se descarta Flowable debido a que en la actualidad las diferencias con Activiti son insignificantes.

	 Activiti	 BonitaBPM	 jBPM
Motor BPM propio			
Editor de procesos			
Integración con tercero			
Integración nativa con Java básico			
Integración nativa con Spring			
Versión Gratuita			
Uso estándar BPMN2			




Buena


Regular



Mala

Figura 5-9 Tabla comparativa herramientas finalistas BPM

Los tres tienen su propio motor de procesos. En el caso de BonitaBPM el motor está apoyado con las otras herramientas de la suite para construir aplicaciones personalizadas. En el caso jBPM y Activiti tienen la posibilidad de funcionar como sistemas aislados o como parte de un sistema mayor; debido a su naturaleza inicial de integración con Alfresco.

Las tres soluciones tienen su propio editor visual de procesos. Si bien las tres basan su funcionamiento en la notación BPMN2, cada una de ellas desarrolla partes extendidas completando y enriqueciendo el lenguaje con ciertas notaciones o funcionalidades. Por ejemplo, a la hora de definir una ServiceTask encargada de enviar un email, el estándar no reconoce una tarea propia para este fin. Sin embargo, todas tienen en su editor la EmailTask.

En la integración con terceros es BonitaBPM la herramienta con mejor puntuación. Se trata de uno de sus principales puntos fuertes. Sin embargo, Activiti solo tiene de forma nativa integración con Apache Camel y MuleTask. Por su parte, jBPM está muy centrada en integración con terceras herramientas, pero todas dentro del paraguas de RedHat.

A la hora de hablar de soporte para desarrollos, Activiti es por mucho la mejor posicionada. Tanto Activiti como jBPM tienen soporte nativo para Java y su motor puede correr dentro de un desarrollo de software realizado en esta tecnología. Esto no ocurre de forma tan simple con BonitaBPM. Aunque también pueden desarrollarse a través de la su API Bonita BPM Engine.

Lo que es exclusivo de Activiti es la integración con el ecosistema Spring. Activiti se integra perfectamente con Spring Boot, con Spring MVC y el motor soporta JPA.

Además, cuenta con su propia API Rest que facilita enormemente la creación de soluciones web que interactúen con el motor.

Esto unido a que a fecha de hoy es la única de las tres herramientas completamente gratuita y sin ninguna restricción hace, que Activiti salga como clara vencedora para el propósito de este trabajo.

5.4.1.3. Diseño de un BPM con Activiti

La herramienta Activiti, se divide en una serie de componentes que se engloban en tres áreas:

- Herramientas de Diseño
- Motor de Procesos
- Herramientas de Soporte

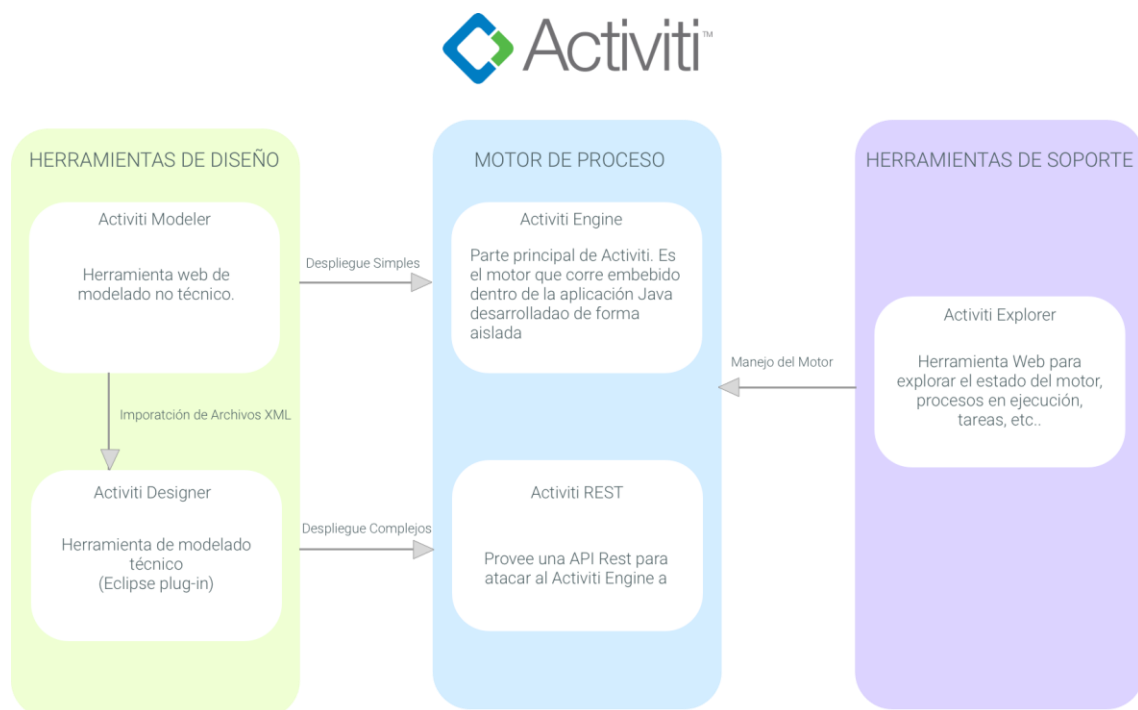


Figura 5-10 Suite Activiti

Herramientas de Diseño

Una de las principales ventajas que aportan los BPM al mundo del software es la creación de un lenguaje común basado en la notación BPMN20. Eso hace que participantes de diferentes departamentos puedan comunicarse bajo un mismo lenguaje formal, pero a la vez accesible.

Dentro de esta funcionalidad están los editores visuales de diagramas BPMN. Estos editores permiten la creación de flujos de procesos de negocio de forma visual que más tarde son interpretados por los motores de BPM.

Activiti, tiene dos herramientas que pueden utilizarse para diagramar flujos: un editor Web y un plugin de Eclipse. El Editor Web está pensado para realizar diagramas más sencillos o de conceptualización, mientras que el plugin de Eclipse permite un mayor control técnico sobre el modelado.

En cualquiera de los dos casos, las dos herramientas producen la misma salida, un archivo en formato XML capaz de ser interpretado por el motor cuando este lo interpreta.

Activiti Modeler

En la última versión de Activiti, el editor ha sido remodelado considerablemente y se pueden realizar más acciones con él, aunque el editor de Eclipse todavía permite mayor control sobre el flujo diagramado. Esta herramienta ha sido donada al producto por la empresa Signavio (empresa donde trabaja actualmente Tom Baeyens).

Activiti Designer

Se trata de un Plugin de Eclipse que contiene más opciones de edición que el modelador Web. Al estar integrado en Eclipse, permite incorporar información del propio proyecto Java, como variables, entidades y cualquier otro elemento del proyecto necesario.

Motor de Procesos

Activiti Engine

Corazón del motor de procesos y por lo tanto de Activiti. Es el encargo de recibir los modelos BPM, interpretarlos, desplegarlos, gestionar los procesos, manejar sus tareas, etc.

Con una visión simplificada se puede decir que un BPM es un conjunto de tareas de usuario o automáticas. Estas tareas se conectan entre sí a través de conectores y puertas de enlaces (gateways) que van conformando el flujo de ejecución del mismo. Estos gateways puede ser excluyente, paralelos, temporizados.



El motor puede tener ejecutándose múltiples BPM y cada uno de ellos levantar múltiples procesos. Es decir, el motor puede tener un BPM-A que se esté ejecutando en el Proceso1 y en el Proceso2. Cada proceso es completamente independiente.

Internamente el motor es una máquina de estados que va controlando la lógica y las transiciones de los mismos. Esta máquina de estados está completamente escrita en Java.

Activiti Rest

Punto de entrada al motor de Activiti. Expone las operaciones del motor a través de servicios REST. Se puede consultar procesos, completar tareas, listar tareas terminadas, consultar usuarios y prácticamente cualquiera de las operaciones que pueden realizarse con el motor.



Herramientas de Soporte

Activiti Explorer es una herramienta web que sirve como cuadro de mando para el motor. Se puede consultar información como el número de procesos desplegados, instancias que están corriendo, realizar la gestión de los mismos, etc.

Sistema Gestor de Base de Datos en Activiti.

Una de las principales ventajas de Activiti es su compatibilidad con el estándar jdbc pudiendo realizar conexiones con las principales bases de datos actuales del mercado. Por defecto la herramienta viene preparada para utilizar H2 Database Engine, una base de datos con modo memoria puramente Java, de un tamaño muy reducido, pero bastante versátil y potente.

Tras realizar una comparativa de las diferentes opciones, se opta por utilizar la base de datos por defecto H2 en el desarrollo de este trabajo. Por ser la que más se acercaba al propósito del mismo.

					
Java Nativo					
Modo Memoria					
Base de datos Cifrada					
Driver ODBC					
Búsqueda por Fulltext					
Multi version concurrente					
	~1Mb	~2Mb	~1Mb	~4Mb	~6Mb


 Si
  No

Figura 5-11 Tabla Comparativa SGBD

La distribución de Activiti viene con los scripts de creación de los esquemas para diferentes distribuciones: Db2, H2, Hsql, Oracle y Postges

Independientemente de la base de datos elegida, los esquemas creados son iguales siguiendo los siguientes patrones:

- ACT_RE_*: RE de repository. Las tablas con este prefijo contienen la información estática como definiciones de proceso, recursos (imágenes, reglas, etc.).
- ACT_RU_*: RU de runtime. Estas tablas contienen los datos de las instancias a los procesos, tareas de usuarios, variables, etc. Activiti solo usa estas tablas durante la ejecución de instancias de los procesos y borra las filas creadas cuando esa instancia de proceso ha terminado. Esto hace que estas tablas estén optimizadas al ser más pequeñas y rápidas.
- ACT_ID_*: ID de identity. Estas tablas contiene la información de identidad como usuarios, grupos, etc.
- ACT_HI_*: HI de history. Estas tablas contienen los datos histórico, como instancias de procesos, variables, tareas, etc..
- ACT_GE_*: GE de general. Estas tablas tienen varios casos de uso para el funcionamiento de la herramienta.

El conocimiento del esquema es deseable para conocer el funcionamiento interno de Activiti, pero el desarrollador en ningún momento tiene por que atacar directamente estos modelos. El motor en sí, contiene los métodos necesarios para realizar cualquier tipo de consulta. En último caso, si fuera necesario alguna consulta muy compleja que no tuviese un método creado para ello, el motor contiene métodos especiales denominadas QueryNative que permite ejecutar consultas más elaboradas.



Figura 5-12 Visibilidad de los diferentes componentes de Activiti

A la hora de configurar la base de datos, dependiendo de la solución tecnológica que se vaya a realizar es necesario configurarla de una forma u otra, pero siempre dentro de los estándares de jdbc.

Por ejemplo:

- Realizar la configuración programáticamente en tiempo de ejecución del programa:

```
ProcessEngineConfiguration cfg = new StandaloneProcessEngineConfiguration()
    .setJdbcUrl("jdbc:h2:mem:activiti;DB_CLOSE_DELAY=1000").setJdbcUsername("sa")
    .setJdbcPassword("")
    .setJdbcDriver("org.h2.Driver")
    .setDatabaseSchemaUpdate(ProcessEngineConfiguration.DB_SCHEMA_UPDATE_TRUE);
```

- Configuración de la base de datos en un proyecto Java simple

Se añade la información el archivo activiti.cfg.xml que irá en la carpeta Resources del proyecto.

```
<property name="jdbcUrl" value="jdbc:h2:mem:activiti;DB_CLOSE_DELAY=1000"/>
<property name="jdbcDriver" value="org.h2.Driver" />
<property name="jdbcUsername" value="sa" />
<property name="jdbcPassword" value="">
```

- Configuración de la base de datos en un proyecto con Spring Boot

Cuando el proyecto se realiza con SpringBoot se crea una instancia H2 de forma automática. Si se desea modificar esta instancia por otro motor de base de datos, basta con incluir un Bean similar a este para su reconfiguración.

```
@Bean public DataSource database() {
    return DataSourceBuilder.create()
        .url("jdbc:mysql://127.0.0.1:3306/activiti-spring-boot?characterEncoding=UTF-8")
        .username("alfresco")
        .password("alfresco")
        .driverClassName("com.mysql.jdbc.Driver")
        .build();
}
```

Persistencia en Activiti

Además del uso de su propio gestor de base de datos, Activiti tiene una gran ventaja respecto a sus competidores a la hora de manejar información en sus tareas; es capaz de manejar objetos POJO para usar la interfaz de JPA.

Por defecto si se usa SpringBoot, está preparado para trabajar con Hibernate.

Se debe añadir la siguiente dependencia en el POM

```
<dependency>
  <groupId>org.activiti</groupId>
  <artifactId>activiti-spring-boot-starter-jpa</artifactId>
  <version>${activiti.version}</version>
</dependency>
```

Con esto se incluyen en la configuración de Spring y los beans necesarios para usar JPA. El proveedor que viene por defecto configurado es Hibernate.

Esto se puede utilizar para guardar información fuera del modelo del motor o para alimentar con información de un sistema externo al BPM.

Más adelante, en la explicación técnica de la solución creada se explicará con más detalle su uso.

Envío de Emails en Activiti

Uno de los tipos de tareas de servicios que permite Activiti es la de envío de emails. Para poder completar este tipo de tareas es necesario configurar el proyecto con un servidor de correo que soporte SMTP. Las tareas de tipo Email son capaces de enviar emails a uno o más destinatarios, incluir texto normal, incluir código HTML, enviar CC, CCO, etc.

Para configurarlo es necesario incluir las propiedades de configuración en el archivo

activiti.cfg.xml

Las posibles propiedades de configuración son:

Propiedad	Requerido	Descripción
mailServerHost	No	El nombre del host de tu servidor. Por defecto es localhost.
mailServerPort	Si, si no es el puerto por defecto	El puerto para el tráfico SMTP. Por defecto es el 25
mailServerDefaultFrom	No	La dirección por defecto del que envía el mensaje.
mailServerUsername	Si el servidor requiere autenticación	Usuario del correo
mailServerPassword	Si el servidor requiere autenticación	Password del correo
mailServerUseSSL	Si es aplicable al servidor	Algunos servidores requieren por defecto usar SSL. Por defecto está a falso
mailServerUseTLS	Si es aplicable al servidor	Algunos servidores requieren por defecto usar TLS. Por defecto está a falso

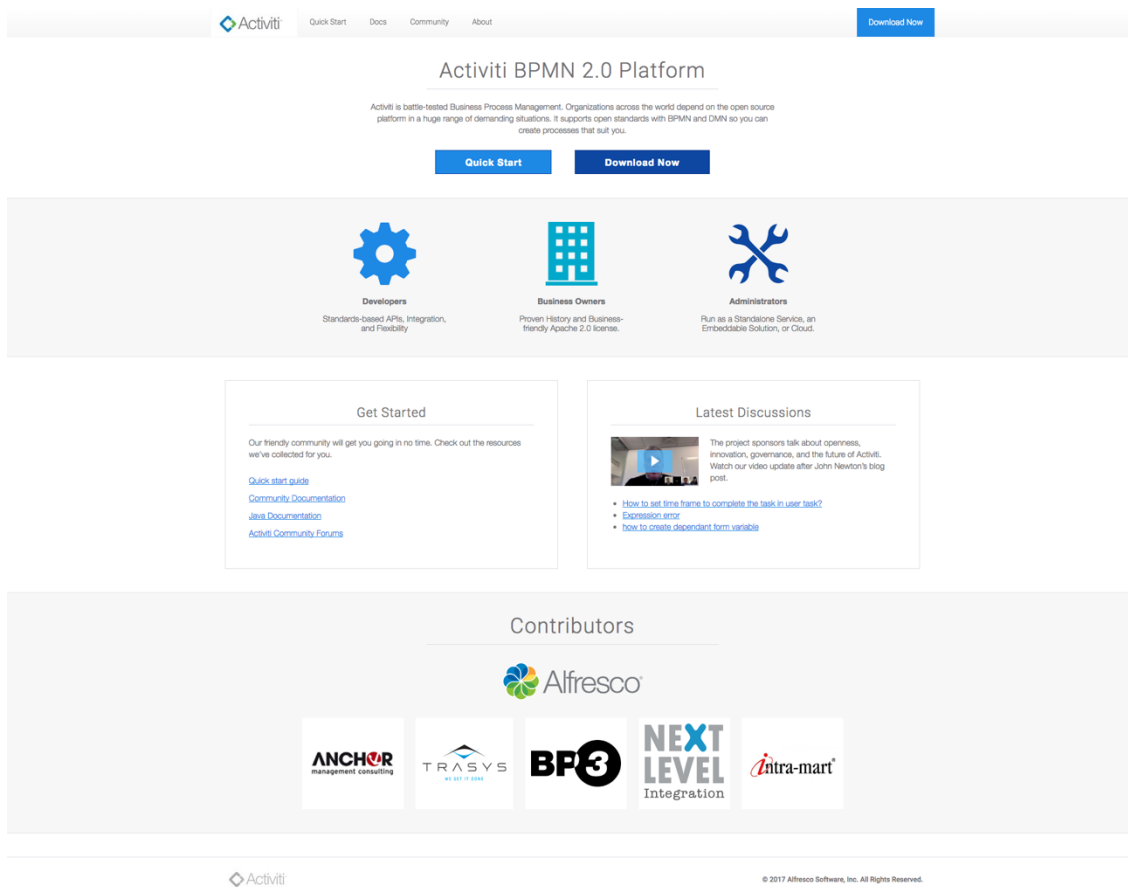
En el BPM, las tareas de tipo email se definen así:

```
<serviceTask id="sendMail" activiti:type="mail">
```

Instalación y Primer Ejemplo

Para entender mejor el funcionamiento de Activiti, vamos realizar una primera aproximación con un ejemplo sencillo cubriendo todo el desarrollo. Vamos a crear una BPM con dos pasos desde el modelador y después crear un proyecto que lo consuma.

Desde la página web <https://www.activiti.org/> realizar la descarga



The screenshot shows the Activiti BPMN 2.0 Platform website. At the top, there is a navigation bar with links for Quick Start, Docs, Community, and About, and a Download Now button. The main heading is "Activiti BPMN 2.0 Platform". Below this, a brief description states: "Activiti is battle-tested Business Process Management. Organizations across the world depend on the open source platform in a huge range of demanding situations. It supports open standards with BPMN and DMN so you can create processes that suit you." There are two buttons: "Quick Start" and "Download Now".

Below the buttons, there are three columns representing different user roles:

- Developers**: Standards-based APIs, Integration, and Flexibility.
- Business Owners**: Proven History and Business-friendly Apache 2.0 license.
- Administrators**: Run as a Standalone Service, an Embeddable Solution, or Cloud.

There are two main content boxes:

- Get Started**: "Our friendly community will get you going in no time. Check out the resources we've collected for you." Links include: Quick start guide, Community Documentation, Java Documentation, and Activiti Community Forums.
- Latest Discussions**: "The project sponsors talk about openness, innovation, governance, and the future of Activiti. Watch our video update after John Newton's blog post." Links include: How to set time frame to complete the task in user task?, Expression error, and how to create dependant form variable.

At the bottom, there is a "Contributors" section with logos for Alfresco, ANCHOR management consulting, TRASY, BP3, NEXT LEVEL Integration, and intra-mart. The footer contains the Activiti logo and the copyright notice: "© 2017 Alfresco Software, Inc. All Rights Reserved."

Figura 5-13 Pantalla de Descarga

Se descargará el archivo activiti-6.0.0.zip. La distribución contiene lo siguiente:

activiti-6.0.0		
Nombre	Tamaño	Clase
▶ database	--	Carpeta
▶ libs	--	Carpeta
▶ wars	--	Carpeta
license.txt	11 KB	Texto
notice.txt	15 KB	Texto
readme.html	14 KB	HTM...me
activiti_readme.png	14 KB	Imag...PN

Figura 5-14 Paquete Descargado

- Carpeta database: contiene los scripts de creación, actualización y borrado de los esquemas del modelo para los diferentes sistemas gestores de base de datos.
- Carpeta libs: contiene las librerías *.jar del motor y las diferentes utilidades
- Carpeta wars contiene las aplicaciones para desplegar
 - activiti-admin: herramienta administrativa
 - activiti-app: modelador
 - actitiviti-rest: api-rest

Estas aplicaciones se despliegan en un servidor Tomcat previamente instalado. Para el ejemplo que vamos a realizar usaremos activiti-app que contiene el modelador.

Una vez desplegada entramos a la ruta localhost:8080\activiti-app y nos logamos con admin/test

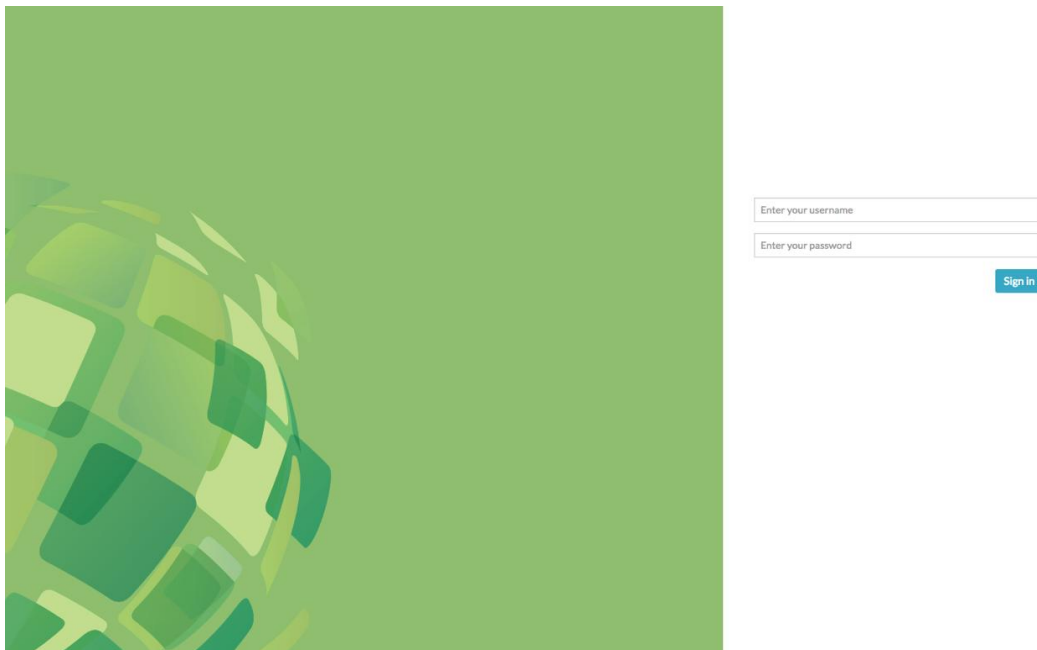


Figura 5-15 Entrada al Activiti Modeler

La siguiente pantalla es un menú donde elegiremos la opción Kickstart App para ir a la ventana de procesos que contiene la instancia levantada ahora mismo.

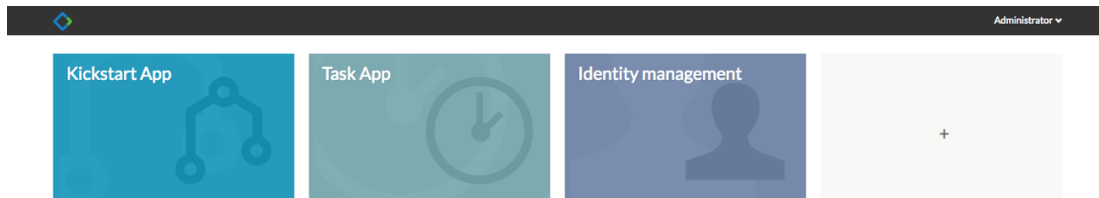


Figura 5-16 Menú del Activiti Modeler

En esta pestaña pinchamos sobre Create Process

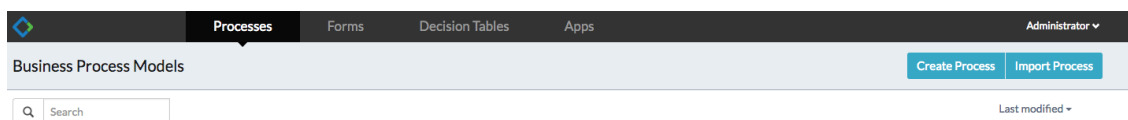


Figura 5-17 Detalle botón crear proceso

Nos pedirá la información para la creación del modelo: nombre, clave y descripción.

Create a new business process model

You need to give a name for the new model and you may want to add a description at the same time.

Model name

PruebaBPM

Model key

PruebaBPM

Description

Mi primer BPM

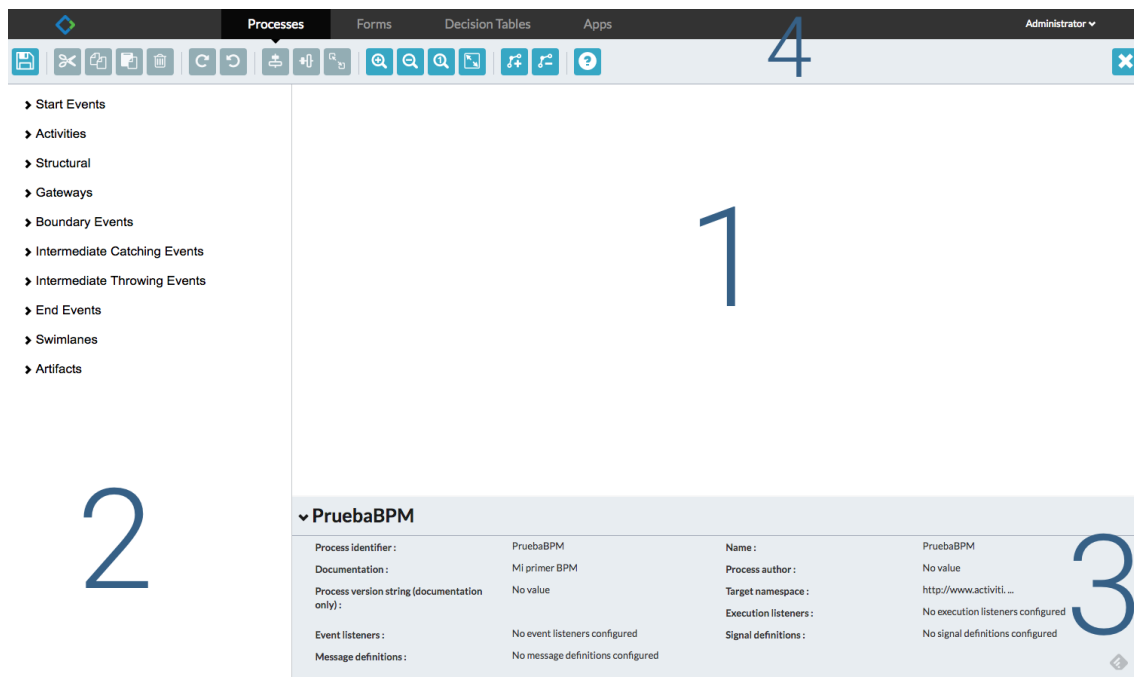
Cancel

Create new model

Figura 5-18 Formulario de Creación

Con esto se abre la ventana del modelador: este consta de las siguientes áreas:

1. Área de dibujo donde iremos modelando el BPM.
2. Área de componentes a seleccionar.
3. Información de selección. Se pueden editar campos de los componentes directamente aquí.
4. Barra de menú



PruebaBPM			
Process identifier:	PruebaBPM	Name:	PruebaBPM
Documentation:	Mi primer BPM	Process author:	No value
Process version string (documentation only):	No value	Target namespace:	http://www.activiti...
Event listeners:	No event listeners configured	Execution listeners:	No execution listeners configured
Message definitions:	No message definitions configured	Signal definitions:	No signal definitions configured

Figura 5-19 Pantalla Principal del Modelador

El BPM que vamos a pintar consta de cuatro componentes conectados: el evento de inicio, tarea 1, tarea 2 y evento final.



Figura 5-20 BPM de Ejemplo

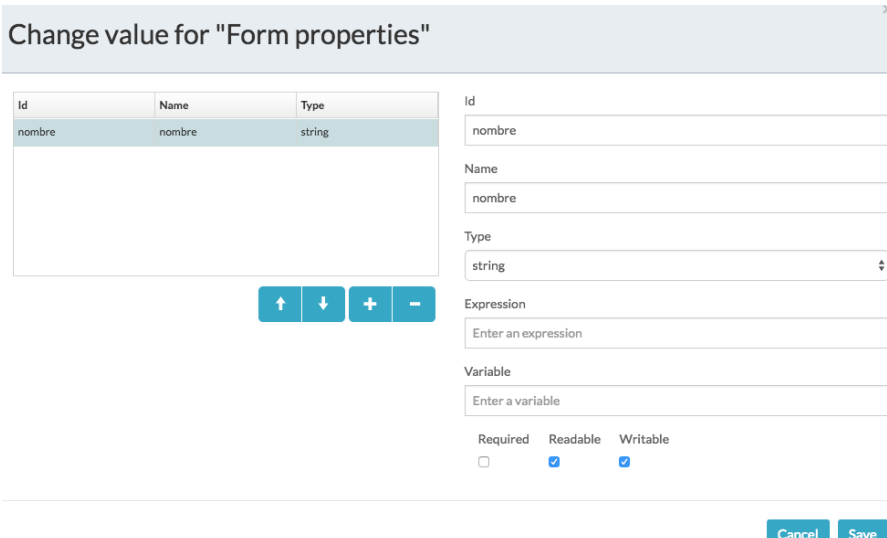
Tanto la tarea1 como la tarea 2 van a tratarse de tareas de usuario. La primera será la tarea de solicitar nombre y la segunda la de solicitar apellido.

Para ello definiremos una variable de formulario en cada una de las tareas de tipo String.

En Activiti los campos que se definen en el formulario son asociados a la tarea como variables. Como se ha explicado previamente, estas pueden ser variables de tipo simple o utilizar datos más complejos como entities de JPA.

Cada tarea, tiene un nombre y un id. El id es el campo clave. Si no se informa, Activiti marca uno por defecto. En nuestro caso vamos a rellenar los dos con el mismo valor porque resultará mucho más útil a la hora de hacer uso de él durante el desarrollo.

Para cada una de las tareas se informan los campos de formulario que se quieren añadir en una pantalla como la siguiente:



Change value for "Form properties"

Id	Name	Type
nombre	nombre	string

Id:

Name:

Type:

Expression:

Variable:

Required: Readable: Writable:

Cancel Save

Figura 5-21 Detalle creación de formulario

Una vez terminado salvamos el diagrama y cerramos el editor. En la pantalla posterior veremos lo siguiente.

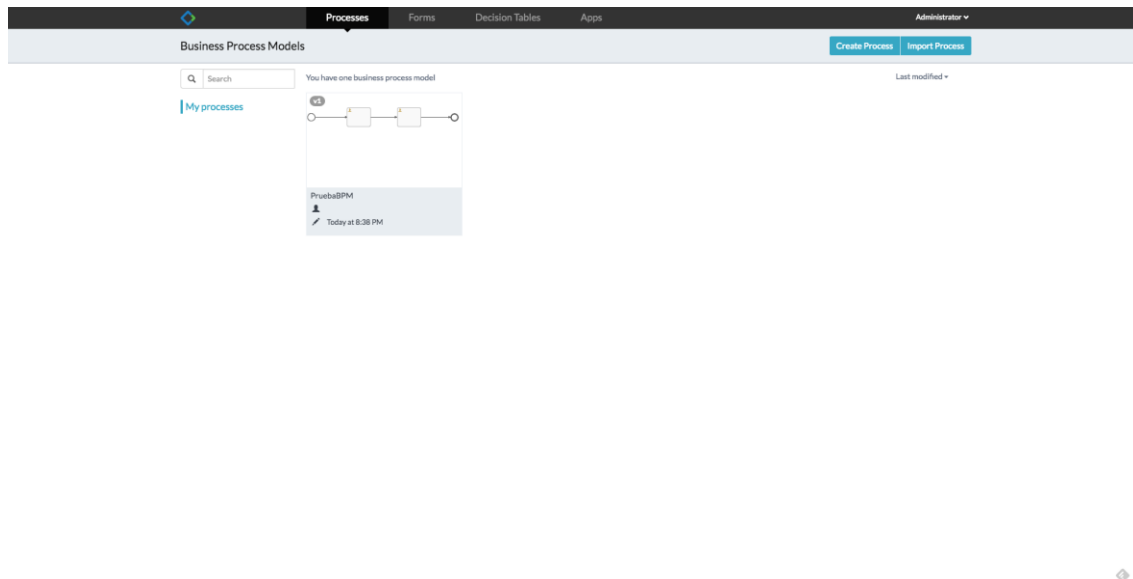


Figura 5-22 Biblioteca de procesos modelados

Si lo seleccionamos y le damos a importar procesos, descargaremos el fichero XML con el BPM generado que tiene el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:activiti="http://activiti.org/bpmn"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI" typeLanguage="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.w3.org/1999/XPath" targetNamespace="http://www.activiti.org/processdef">
  <process id="PruebaBPM" name="PruebaBPM" isExecutable="true">
    <documentation>Mi primer BPM</documentation>
    <startEvent id="startEvent1" name="Inicio Proceso"></startEvent>
    <userTask id="Paso1" name="Solicitar Nombre">
      <extensionElements>
        <activiti:formProperty id="nombre" name="nombre" type="string"></activiti:formProperty>
      </extensionElements>
    </userTask>
    <userTask id="Paso2" name="Solicitar Apellido">
      <extensionElements>
        <activiti:formProperty id="apellido" name="apellido" type="string"></activiti:formProperty>
      </extensionElements>
    </userTask>
    <sequenceFlow id="sid-7A31CF5D-FBCB-4F66-94C8-1FE4330EAD88" sourceRef="Paso1"
targetRef="Paso2"></sequenceFlow>
    <sequenceFlow id="sid-FBD6B2A4-DB61-4D4C-AD8D-16E017573428" sourceRef="startEvent1"
targetRef="Paso1"></sequenceFlow>
    <endEvent id="sid-733D9836-4D9A-4639-A2D3-26BBD8BDD3AF" name="Final del Proceso"></endEvent>
    <sequenceFlow id="sid-FC697087-2088-4776-9671-0BBECE38C558" sourceRef="Paso2" targetRef="sid-733D9836-
4D9A-4639-A2D3-26BBD8BDD3AF"></sequenceFlow>
  </process>
  <bpmndi:BPMNDiagram id="BPMNDiagram_PruebaBPM">
    <bpmndi:BPMNPlane bpmnElement="PruebaBPM" id="BPMNPlane_PruebaBPM">
      <bpmndi:BPMNShape bpmnElement="startEvent1" id="BPMNShape_startEvent1">
        <omgdc:Bounds height="30.0" width="30.0" x="90.0" y="165.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
      <bpmndi:BPMNShape bpmnElement="Paso1" id="BPMNShape_Paso1">
        <omgdc:Bounds height="80.0" width="100.0" x="255.0" y="140.0"></omgdc:Bounds>
      </bpmndi:BPMNShape>
    </bpmndi:BPMNPlane>
  </bpmndi:BPMNDiagram>

```

```

    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="Paso2" id="BPMNShape_Paso2">
      <omgdc:Bounds height="80.0" width="100.0" x="465.0" y="140.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="sid-733D9836-4D9A-4639-A2D3-26BBD8BDD3AF" id="BPMNShape_sid-733D9836-4D9A-4639-A2D3-26BBD8BDD3AF">
      <omgdc:Bounds height="28.0" width="28.0" x="690.0" y="166.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNEdge bpmnElement="sid-7A31CF5D-FBCB-4F66-94C8-1FE4330EAD88" id="BPMNEdge_sid-7A31CF5D-FBCB-4F66-94C8-1FE4330EAD88">
      <omgdi:waypoint x="355.0" y="180.0"></omgdi:waypoint>
      <omgdi:waypoint x="465.0" y="180.0"></omgdi:waypoint>
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge bpmnElement="sid-FBD6B2A4-DB61-4D4C-AD8D-16E017573428" id="BPMNEdge_sid-FBD6B2A4-DB61-4D4C-AD8D-16E017573428">
      <omgdi:waypoint x="120.0" y="180.0"></omgdi:waypoint>
      <omgdi:waypoint x="255.0" y="180.0"></omgdi:waypoint>
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge bpmnElement="sid-FC697087-2088-4776-9671-0BBECE38C558" id="BPMNEdge_sid-FC697087-2088-4776-9671-0BBECE38C558">
      <omgdi:waypoint x="565.0" y="180.0"></omgdi:waypoint>
      <omgdi:waypoint x="690.0" y="180.0"></omgdi:waypoint>
    </bpmndi:BPMNEdge>
  </bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</definitions>

```

Ahora, utilizando el IDE Eclipse vamos a crear un proyecto Maven. Antes en la descarga realizada hemos obtenido todas las librerías necesarias para trabajar con Activiti, pero por comodidad vamos a realizar la gestión de dependencias usando Maven.

Las dependencias necesarias para añadir al pom son las siguientes:

```

<dependencies>
  <dependency>
    <groupId>org.activiti</groupId>
    <artifactId>activiti-engine</artifactId>
    <version>6.0.0</version>
  </dependency>
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>1.4.193</version>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.7.12</version>
  </dependency>
</dependencies>

```

Añadimos el motor de Activiti, la base de datos h2 y slf4j que es usado internamente por el motor para los logs.

Ahora, copiamos el archivo xml con el BPM a la carpeta resources desde donde lo cargaremos. Creamos el paquete com.uah.miprimerbpm con la clase principal APP.java.

Dividiendo la clase en diferentes bloques explicaremos cada uno de estos:

En este primer bloque se crea la configuración del process engine. Se crea una configuración standalone que se ejecuta de forma aislada y se configura los valores de la base de datos a utilizar.

```
System.out.println("Mi Primer BPM");
System.out.println("=====");
ProcessEngineConfiguration cfg = new StandaloneProcessEngineConfiguration()
    .setJdbcUrl("jdbc:h2:mem:activiti;DB_CLOSE_DELAY=1000").setJdbcUsername("sa")
    .setJdbcPassword("")
    .setJdbcDriver("org.h2.Driver")
    .setDatabaseSchemaUpdate(ProcessEngineConfiguration.DB_SCHEMA_UPDATE_TRUE);
ProcessEngine processEngine = cfg.buildProcessEngine();
String pName = processEngine.getName();
String ver = ProcessEngine.VERSION;
System.out.println("ProcessEngine [" + pName + "] Version: ["
```

Una vez que el motor está ejecutándose, vamos a desplegar el BPM generado y mostrar por consola el nombre y el id del BPM desplegado.

```
RepositoryService repositoryService = processEngine.getRepositoryService();
Deployment deployment = repositoryService.createDeployment()
    .addClasspathResource("PruebaBPM.bpmn20.xml").deploy();
ProcessDefinition processDefinition = repositoryService.createProcessDefinitionQuery()
    .deploymentId(deployment.getId()).singleResult();
System.out.println("Encontrado process definition [" + processDefinition.getName() + "] con id ["+
processDefinition.getId() + "]);
```

Se crea una instancia de proceso con el BPM desplegado. Arranca el BPM.

```
RuntimeService runtimeService = processEngine.getRuntimeService();
ProcessInstance processInstance = runtimeService.startProcessInstanceByKey("PruebaBPM");
System.out.println("Comienza la instancia del proceso con id [" + processInstance.getProcessInstanceId()
    + "] clave [" + processInstance.getProcessDefinitionKey() + "]);
```

Se obtiene del motor el el servicio de Tareas, el de Formularios y el Histórico.

Con el primer servicio se gestionan las tareas del proceso, con el segundo se gestiona la información de los formularios de cada tareas y el tercero sirve para consultar la información de las tareas y procesos que ya están completados.


```
TaskService taskService = processEngine.getTaskService();  
FormService formService = processEngine.getFormService();  
HistoryService historyService = processEngine.getHistoryService();
```

En el siguiente párrafo se usa un bucle par ir leyendo last areas pendientes hasta que no queda ninguna. Por cada tarea pendiente se busca los campos que lleva su formulario y se solicitan a través de consola. Con la información recogida se completa la tarea lo que hace que el motor se coloque en la siguiente.

```
Scanner scanner = new Scanner(System.in);  
List<Task> tasks = taskService.createTaskQuery().list();  
while (tasks.size() > 0) {  
    System.out.println("Tareas Pendientes: [" + tasks.size() + "]");  
    for (int i = 0; i < tasks.size(); i++) {  
        Task task = tasks.get(i);  
        System.out.println("Procesando la tarea [" + task.getName() + "]");  
        Map<String, Object> variables = new HashMap<String, Object>();  
        FormData formData = formService.getTaskFormData(task.getId());  
        for (FormProperty formProperty : formData.getFormProperties()) {  
            if (StringFormType.class.isInstance(formProperty.getType())) {  
                System.out.println(formProperty.getName() + "?");  
                String value = scanner.nextLine();  
                variables.put(formProperty.getId(), value);  
            } else {  
                System.out.println("<formato no soportado");  
            }  
        }  
        taskService.complete(task.getId(), variables);  
    }  
    tasks = taskService.createTaskQuery().list();  
}
```

Por último se utiliza el servicio de histórico para recorrer todas las tareas finalizadas y mostrar su información.

```
HistoricActivityInstance endActivity = null;  
List<HistoricActivityInstance> activities =  
historyService.createHistoricActivityInstanceQuery().processInstanceId(processInstance.getId())  
    .finished().orderByHistoricActivityInstanceEndTime().asc().list();  
  
for (HistoricActivityInstance activity : activities) {  
    if (activity.getActivityType() == "startEvent") {  
        System.out.println("COMIENZO " + processDefinition.getName() + " [" +  
            + processInstance.getProcessDefinitionKey() + "] " +  
            activity.getStartTime());  
    }  
    if (activity.getActivityType() == "endEvent") {  
        endActivity = activity;  
    } else {  
        System.out.println("-- " + activity.getActivityName() + " [" +  
            activity.getActivityId() + "] " + activity.getDurationInMillis() + " ms");  
    }  
}  
  
if (endActivity != null) {  
    System.out.println("-- " + endActivity.getActivityName() + " [" +  
        endActivity.getActivityId() + "] " + endActivity.getDurationInMillis() + " ms");  
    System.out.println("FINAL " + processDefinition.getName() + " [" +  
        processInstance.getProcessDefinitionKey()+ "] " +
```

```
endActivity.getEndTime());  
}
```

La ejecución queda de la siguiente manera.

=====
Mi Primer BPM

```
[main] INFO org.activiti.engine.compatibility.DefaultActiviti5CompatibilityHandlerFactory - Activiti 5 compatibility handler implementation not  
[main] INFO org.activiti.engine.impl.db.DbSqlSession - performing create on engine with resource org/activiti/db/create/activiti.h2.create.engi  
[main] INFO org.activiti.engine.impl.db.DbSqlSession - performing create on history with resource org/activiti/db/create/activiti.h2.create.his  
[main] INFO org.activiti.engine.impl.db.DbSqlSession - performing create on identity with resource org/activiti/db/create/activiti.h2.create.id  
[main] INFO org.activiti.engine.impl.ProcessEngineImpl - ProcessEngine default created  
ProcessEngine [default] Version: [6.0.0.4]  
Encontrado process definition [PruebaBPM] con id [PruebaBPM:1:4]  
Comienza la instancia del proceso con id [5] clave [PruebaBPM]  
Tareas Pendientes: [1]  
Procesando la tarea [Solicitar Nombre]  
nombre?  
Pedro  
Tareas Pendientes: [1]  
Procesando la tarea [Solicitar Apellido]  
apellido?  
Palazón  
-- Inicio Proceso [startEvent1] 1 ms  
-- Solicitar Nombre [Paso1] 33656 ms  
-- Solicitar Apellido [Paso2] 2161 ms  
-- Final del Proceso [sid-733D9836-4D9A-4639-A2D3-268BD88DD3AF] 0 ms
```

El paso Final de Proceso tiene una clave sid-* porque la ha creado automáticamente el modelador al no informar una concreta.

Con este ejemplo se ha cubierto un desarrollo de principio a fin de un BPM. En el capítulo siguiente, se entrará más en detalle en la integración real realizada. Orquestando el BPM con tecnologías Spring.

5.4.2. Diseño Funcional de la Solución

5.4.2.1. Marco de Trabajo

Para llevar a cabo el diseño de la solución y demostrar las hipótesis planteadas vamos a definir un caso de estudio.

Nos encontramos con una Factoría de Software que da servicios a varios clientes. Esta factoría se encuentra dividida en grupos disciplinares. Dentro de cada grupo existe un responsable. Hay más de un grupo por cada disciplina. Los evolutivos son llevados por un único jefe de proyecto, aunque puede recibir apoyo de otros jefes de proyecto en el tiempo. Así mismo, los equipos disciplinares pueden estar trabajando en varios proyectos a la vez de forma simultánea o intercalarse en diferentes fases del evolutivo.

Generalmente un evolutivo que es comenzado por un grupo se mantiene en el tiempo. Si bien es cierto que un grupo disciplinar puede cambiar e intercalarse en diferentes fases del proyecto.

La Factoría recibe en sus peticiones los diseños técnicos detallados. Estos diseños van pasando por diferentes procesos hasta llegar a la solución final que es enviada al cliente para su despliegue en producción.

De esta forma quedan definidos los siguientes participantes del sistema:

ProductOwner

Es el interlocutor designado por el cliente para hablar en su nombre durante el evolutivo. Puede ser alguien de su plantilla relacionada con negocio, con gestión de proyectos, a veces un usuario, etc. Es externo a la Factoría de Software.

Entre sus tareas están:

- Entrega del diseño físico.
- Pruebas de integración.
- Aprobación al despliegue.

Jefe de Proyecto

Personal de la Factoría. A veces externo a la Factoría, pero de la misma organización. Es el encargado de llevar el evolutivo de principio a fin; velando porque se cumplan en cada una de las fases con los términos acordados. El evolutivo generalmente será dado de alta por él en el sistema para que comience el BPM. Visará cada una de las fases indicando su conformidad o disconformidad con lo realizado. Los equipos disciplinares son autónomos y funcionan de forma independiente, pero en última instancia en lo relativo a un evolutivo, los responsables dependen del Jefe de Proyecto.

Entre sus tareas están:

- Alta del Proyecto.
- Conformidad a los activos codificados.
- Conformidad a las pruebas unitarias realizadas
- Conformidad a las Pruebas Funcionales

Responsable Analistas

Especialista encargada de realizar los análisis funcionales y técnicos de los sistemas. Su labor dentro de la Factoría del Software es velar que los diseños entregados están en línea con lo esperado funcionalmente del evolutivo y que están completos y correctos para que el equipo de desarrollo pueda realizar su trabajo. Durante el ciclo de vida del evolutivo también interviene durante las pruebas funcionales del sistema.

La figura de responsable analista tiene a su cargo a diferentes tipos de analistas para realizar las diferentes tareas: analistas expertos en datos, analistas expertos en funcionalidades, analistas expertos en una determinada tecnología, etc.

Entre sus tareas están:

- Validación del diseño entregado.
- Pruebas funcionales.

Responsable Desarrollo

Tiene a su cargo un equipo de programadores encargados de codificar las soluciones de los diseños entregados.

Entre sus tareas están:

- Codificación de activos.

Responsable de Testing QA

Es la responsable del equipo de tester encargados escribir y ejecutar las pruebas unitarias de los activos codificados. Se encarga de velar por las buenas prácticas de la codificación y los estándares de calidad en el mismo.

Entre sus tareas están:

- Pruebas unitarias.

5.4.2.2. Pila de Producto. Historias de Usuario

Con este marco de trabajo, pasamos a confeccionar las historias de usuario que conformará la pila de producto.

A la hora de realizar un análisis funcional, la técnica de la pila de producto a través de historias de usuario, es una técnica utilizada en la gestión de proyectos con metodologías ágiles como SCRUM.

Se ha optado por el uso de esta técnica en vez una más clásica como puede ser la ASI de Métrica 3 por varios motivos:

1. Las definiciones son más rápidas y sencillas de realizar.
2. No se pierde completitud al hacer la especificación.
3. Se pueden mezclar aspectos técnicos y funcionales con facilidad.
4. Explica mucho mejor el producto creado.
5. Actualmente están cogiendo mucha fuerza las metodologías ágiles en equipos de alto rendimiento.

Estas historias de usuario son creadas como una estructura de oración concreta:

Como *participante* quiero *necesidad* para *objetivo*.

1. Como usuario quiero identificarme en el sistema para poder trabajar con mis tareas asignadas.
2. Como usuario quiero poder salir del sistema de forma segura para evitar cualquier usurpación de identidad.
3. Como usuario quiero poder consultar las tareas que tengo asignadas en ese momento para poder ver mi carga de trabajo de forma rápida.
4. Como usuario quiero poder consultar las tareas que he realizado en cualquier evolutivo para poder tener una traza real de mi trabajo.
5. Como usuario quiero poder ver todos los evolutivos que hay en el sistema para estar informado de la carga de trabajo de la factoría.
6. Como usuario quiero ver esta información en una lista paginable para que me resulte más cómodo estudiar la información.
7. Como usuario quiero poder filtrar por búsqueda los evolutivos para que me resulte más cómodo estudiar la información.
8. Como usuario quiero poder ver los servicios modelados por procesos que tiene la factoría para conocer en qué proceso puedo trabajar.
9. Como usuario quiero poder ver el estado de la herramienta y el global de información de las tareas y evolutivos en curso. Para tener una fotografía global del volumen de trabajo de la factoría.
10. Como usuario quiero poder dar de alta un evolutivo para comenzar a trabajar en él.
11. Como jefe de proyecto quiero poder dar de alta un evolutivo informando: su código, el nombre, su descripción, una fecha de inicio y a veces la fecha de finalización.

12. Como ProductOwner quiero poder realizar la entrega del diseño físico informando: una breve descripción del diseño, el número de entregables, la fecha de entrega y la ruta del repositorio. Para que la entrega quede lo más descriptiva posible.
13. Como Responsable Analista quiero poder dar mi conformidad o disconformidad a la entrega de un diseño incluyendo el motivo, para cumplir con el cometido de supervisar los diseños que recibe la factoría.
14. Como Responsable de Desarrollo quiero entregar los activos codificados indicando: el número de activos codificados, la ruta del repositorio donde se encuentran y las observaciones que considera. Para realizar de forma oficial la entrega del código.
15. Como Jefe de Proyecto quiero poder dar mi conformidad o disconformidad a los activos codificados para cumplir el cometido de validar el trabajo realizado por los desarrolladores.
16. Como Responsable de Testing quiero poder informar: el número de test que se han realizado, el número de test que han fallado, la cobertura de código que se ha cubierto, la ruta donde se encuentren las evidencias de las pruebas, las observaciones que puedan ser útiles. Para dejar constancia del trabajo hecho en la fase de pruebas lo más minuciosamente posible.
17. Como Jefe de Proyecto quiero poder dar mi conformidad o disconformidad a las pruebas unitarias para cumplir el cometido de validar el trabajo realizado testadores.
18. Como Responsable Analista quiero poder informar las pruebas funcionales con: los casos de uso realizados, los casos de uso erróneos, la ruta donde se dejan las evidencias de las pruebas y mis observaciones. Para dejar constancia de los trabajos realizados en probar el evolutivo.
19. Como Jefe de Proyecto quiero poder dar mi conformidad o disconformidad a las pruebas funcionales para cumplir el cometido de validar el trabajo realizado por los analistas.
20. Como ProductOwner quiero poder dar mi conformidad o disconformidad a las pruebas hechas en el entorno de integración. Para cumplir el cometido de dejar pasar el evolutivo al estado de pre-liberación.
21. Como ProductOwner quiero poder dar mi conformidad o disconformidad al despliegue del software indicando la fecha que quiero que se despliegue. Para tener control sobre los evolutivos que se liberan.
22. Como usuario quiero recibir un email informativo cuando comience el evolutivo, para estar informado del comienzo.¹
23. Como ProductOwner quiero recibir un correo informativo cuando un diseño entregado sea devuelto. Para corregirlo lo más rápido posible.
24. Como Responsable de desarrollo, quiero recibir un correo cuando algún test unitario sea erróneo. Para corregirlo lo más rápido posible.

25. Como Responsable Analista quiero recibir un email cuando se produzcan errores en las pruebas funcionales. Para actuar lo más rápido posible.
26. Como usuario quiero recibir un email cuando se paralice el evolutivo durante por errores en las pruebas de integración. Para estar informado de la situación.
27. Como usuario quiero recibir un email cuando se paralice el evolutivo por decisión de negocio. Para estar informado de la situación.
28. Como usuario quiero recibir un email cuando el evolutivo sea implantado. Para estar informado de la situación.
29. Como Jefe de Proyecto, quiero recibir todos los emails que tengan que ver con mi evolutivo. Para estar informado de la situación.

5.4.2.3. Diseñando un BPM para el despliegue del Software

Con toda la información recabada en el análisis funcional. Estamos en disposición de crear el BPM que dirigirá el evolutivo. El modelo queda definido por:

- Once tareas de Usuario
 1. Alta proyecto.
 2. Entrega Diseño Físico.
 3. Validación Diseño.
 4. Codificación Activos.
 5. Validación Activos.
 6. Pruebas Unitarias de Activos.
 7. Validación Pruebas Unitarias.
 8. Pruebas Funcionales.
 9. Validación Pruebas Funcionales.
 10. Pruebas de Integración.
 11. Aprobación y Planificación Despliegue.
- Siete Tareas de Servicio
 1. Envío mensaje Comienzo Evolutivo.
 2. Envío mensaje Errores Validación Diseño.
 3. Envío mensaje Errores en Pruebas Unitarias.
 4. Envío Informa Validación Pruebas Funcionales
 5. Envío informe Paralización Despliegue en pruebas de integración.
 6. Envío informe Paralización Despliegue decisión de negocio.



7. Envió informe final despliege.
- Ocho gateways exclusivos:
 1. ¿Es correcto el diseño? Si/No
 2. ¿Conformidad a la validación de activos? Si/No
 3. ¿Hay algún test erróneo? $\text{testErroneos} = 0 / \text{testErroneos} > 0$
 4. ¿Conformidad a las pruebas unitarias? Si/No
 5. ¿Hay algún error funcional? $\text{casosErroneos} = 0 / \text{casosErroneos} > 0$
 6. ¿Conformidad a las pruebas funcionales? Si / No
 7. ¿Conformidad a las pruebas integración? Si / No
 8. ¿Conformidad al despliegue? Si / No

Una vez completada toda la información del BPM a través del modelado, se descarga el archivo XML generado que es lo que realmente interpreta el motor.

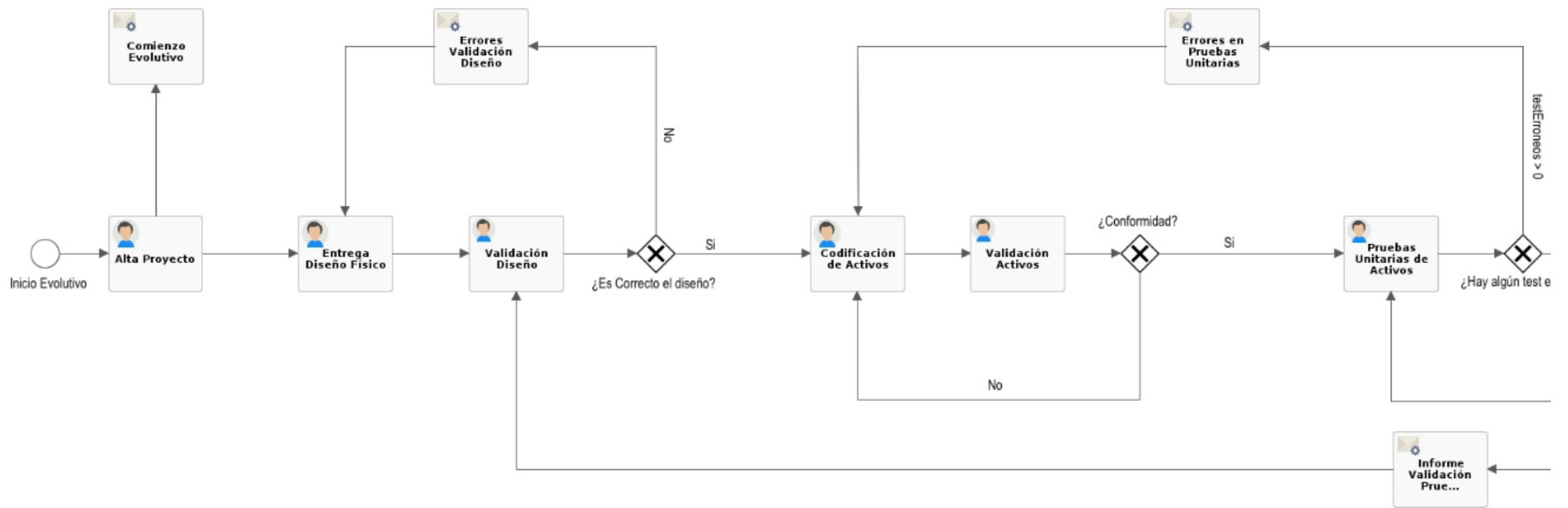


Figura 5-23 BPM Evolutivo Parte I

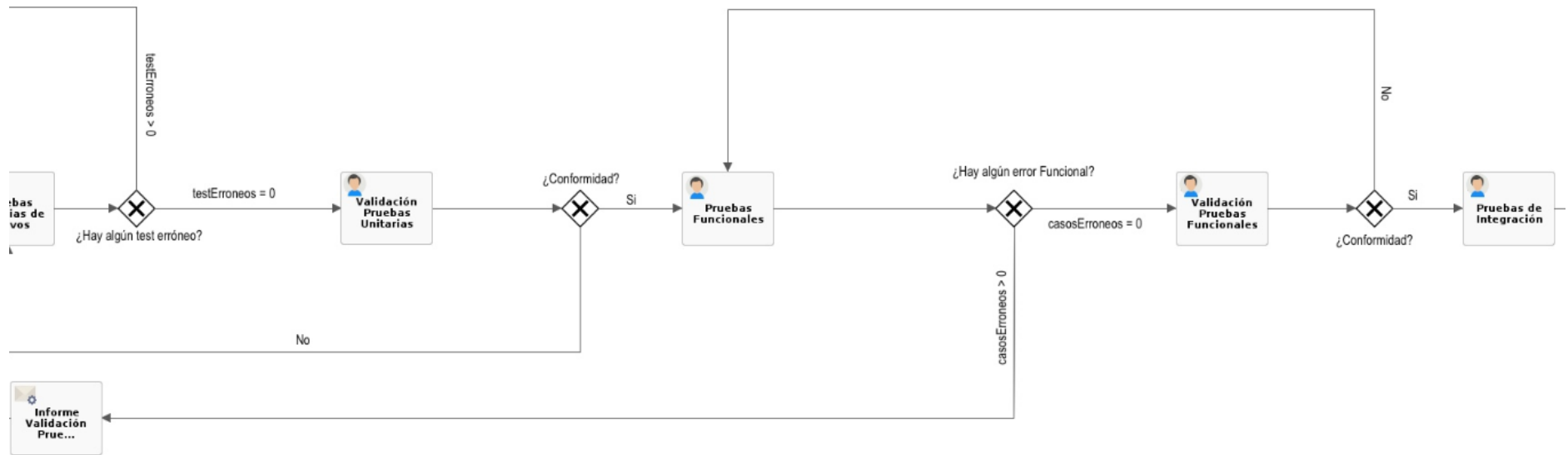


Figura 5-24 BPM Evolutivo Parte II

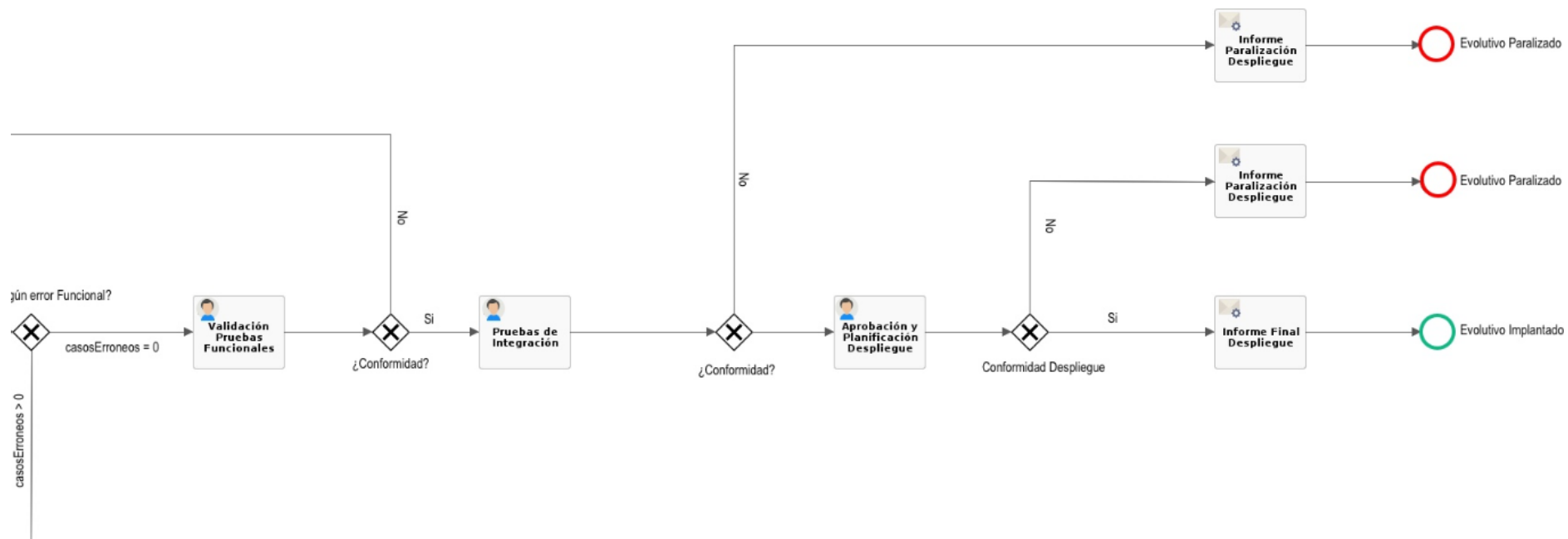


Figura 5-25 BPM Evolutivo Parte III

5.4.2.4. Presentando la solución CofaBPM

CofaBPM es el nombre de la herramienta creada como solución a la comunicación de la Factoría de Software y su cliente durante el ciclo de vida del proyecto. Con especial atención a los hitos más importantes de los mismos y su comunicación entre participantes.

Con un simple vistazo en la herramienta, se puede ver el estado en el que se encuentra el proyecto. Las acciones ya concluidas, las futuras y las acciones requeridas en ese momento.

El nombre de COFABPM es una alegoría náutica. En el mundo náutico, la cofa es la plataforma situada en lo alto de los palos. Su finalidad es facilitar las maniobras de las velas y servir como lugar de observación.

Esta herramienta pretende ser un lugar de observación para ver la salud del proyecto y que sirva de ayuda para iniciar las maniobras de gobierno, al igual que se gobierna un barco.



Figura 5-26 Cofa de un barco

5.4.3. Diseño Técnico de la Solución

5.4.3.1. Solución Arquitectónica

La solución realizada se basa en el patrón arquitectónico Modelo Vista Controlador (MVC). Este patrón se caracteriza por separar los datos y la lógica de negocio de la interfaz de usuario. La finalidad que busca este es tener un gran desacoplamiento y una fuerte cohesión. Con esto se consigue la separación de conceptos y una mayor reutilización de código.

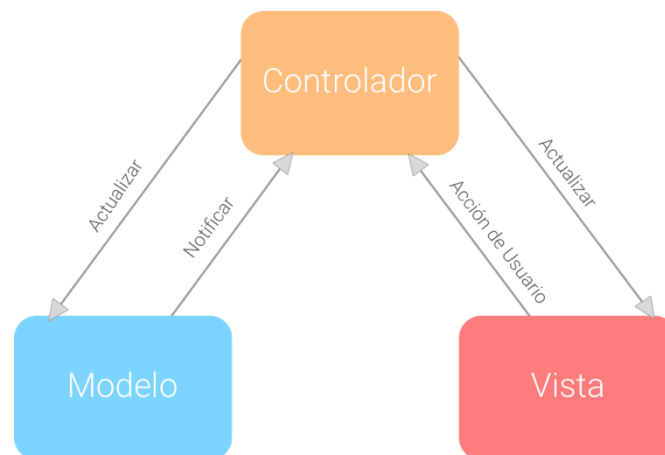


Figura 5-27 Arquitectura Modelo Vista Controlador

En este caso la parte del modelo y el controlador se ejecutan en el lado del servidor y la parte de la vista se ejecuta en lado del cliente, dentro del navegador web.

Se trata de una aplicación atípica en la que no existe un modelo propio diseñado. Por tanto, se considera como Modelo el BPM y la persistencia que lleva intrínseca el motor con su propia base de datos (H2).

El controlador es una aplicación de Spring que expone su lógica de negocio a la vista a través de servicios REST.

La Vista es una Single Page Application (SPA) realizada en AngularJs cuya interfaz de usuario ha sido diseñada usando componentes de la librería de UI Materialize. Esta librería se basa en los estándares Web de Google denominados Material.

La aplicación ha sido desplegada usando Spring Boot. Spring Boot, es una de las últimas herramientas en incorporarse a la suite de Spring. Su funcionalidad es la de crear aplicaciones de instancia única (stand-alone) con el concepto de 'solo ejecuta' (just run). Esto se consigue porque el target construido es una clase java autoejecutable que lleva embebido un servidor Apache Tomcat.

La decisión de utilizar este tipo de tecnología ha venido motivada por su sencillez a la hora de desplegar el proyecto. Además de la integración existente con toda la pila tecnológica de Spring de forma muy sencilla y con el propio motor de Activiti.

A la hora de crear la aplicación usando este modelo se analizaron dos alternativas:

Alternativa A

Utilizar por completo la API REST de Activiti expuesta y trasladar toda la lógica de negocio al cliente dentro de la SPA de Angular.

Alternativa B

Crear una aplicación Spring que contuviese la lógica de negocio y la expusiese a través de servicios REST. De este modo la SPA de Angular solo se encargaría de la parte de visualización. Puntualmente utilizar la API REST de Activiti.

Tras evaluar las dos opciones se opta por la Alternativa B. De esta forma se consigue exponer menos la lógica de negocio al manejarla casi por completo en el cliente. Además, al crear nuestros propios servicios REST tenemos más control sobre la información expuesta y se consigue optimizarla. En algunas vistas se analizó que eran necesarias varias llamadas a servicios de la API REST de Activiti para montar la información requerida, con la sobrecarga que ello conllevaba.

La pila de tecnologías utilizadas viene dibujada en el siguiente diagrama.

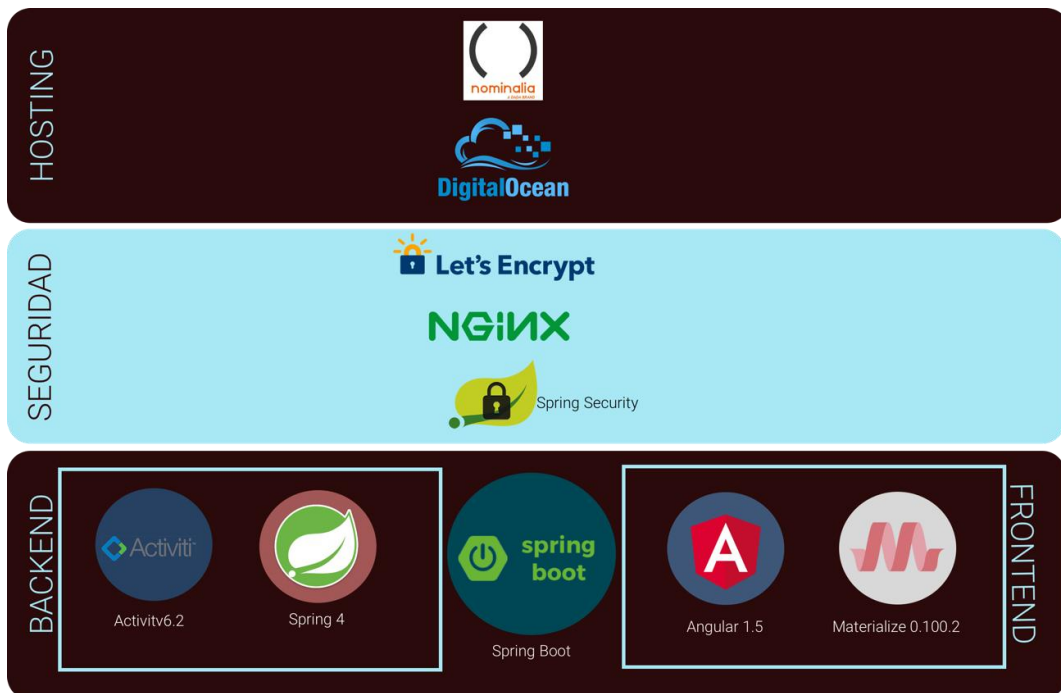
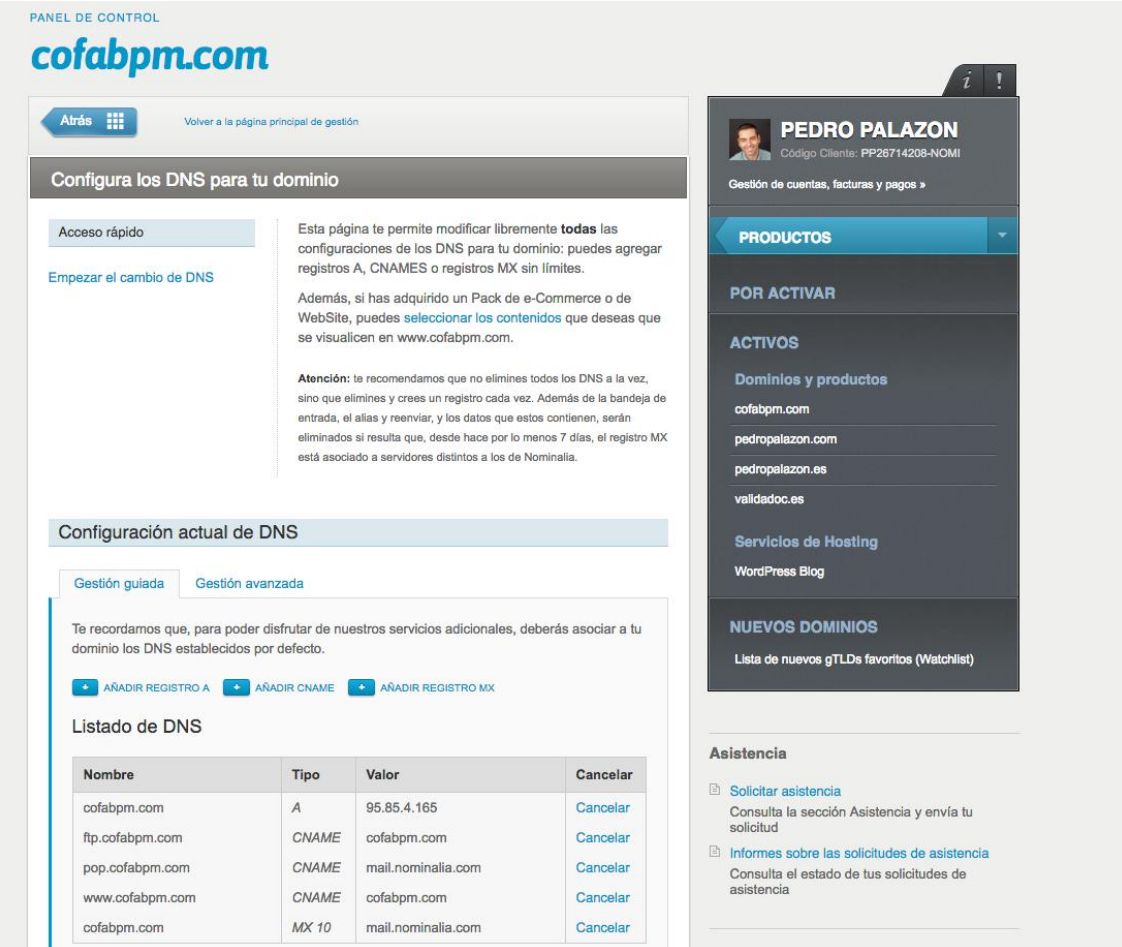


Figura 5-28 Pila Arquitectónica de CofaBPM

5.4.3.2. Hosting

El dominio www.cofabpm.com se ha comprado a través del proveedor de servicio de registros de dominio Nominalia. Con este dominio viene incluido un servidor mail con 5 cuentas de correo que pueden utilizarse a través de SMTP. Una de estas cuentas de correo se ha configurado como info@cofabpm.com y es la utilizada para enviar los mensajes en las Email Task del BPM. Tal y como puede verse en la configuración del DNS del dominio esté ase asocia a la ip del cloud donde está ejecutándose la herramienta.



PANEL DE CONTROL
cofabpm.com

Atrás ⌵ [Volver a la página principal de gestión](#)

Configura los DNS para tu dominio

Acceso rápido
Empezar el cambio de DNS

Esta página te permite modificar libremente **todas** las configuraciones de los DNS para tu dominio: puedes agregar registros A, CNAMEs o registros MX sin límites.

Además, si has adquirido un Pack de e-Commerce o de WebSite, puedes **seleccionar los contenidos** que deseas que se visualicen en www.cofabpm.com.

Atención: te recomendamos que no elimines todos los DNS a la vez, sino que elimines y crees un registro cada vez. Además de la bandeja de entrada, el alias y reenviar, y los datos que estos contienen, serán eliminados si resulta que, desde hace por lo menos 7 días, el registro MX está asociado a servidores distintos a los de Nominalia.

Configuración actual de DNS

Gestión guiada + Gestión avanzada

Te recordamos que, para poder disfrutar de nuestros servicios adicionales, deberás asociar a tu dominio los DNS establecidos por defecto.

[+ AÑADIR REGISTRO A](#) [+ AÑADIR CNAME](#) [+ AÑADIR REGISTRO MX](#)

Listado de DNS

Nombre	Tipo	Valor	Cancelar
cofabpm.com	A	95.85.4.165	Cancelar
ftp.cofabpm.com	CNAME	cofabpm.com	Cancelar
pop.cofabpm.com	CNAME	mail.nominalia.com	Cancelar
www.cofabpm.com	CNAME	cofabpm.com	Cancelar
cofabpm.com	MX 10	mail.nominalia.com	Cancelar

PRODUCTOS

POR ACTIVAR

ACTIVOS

Dominios y productos

- cofabpm.com
- pedropalazon.com
- pedropalazon.es
- validadoc.es

Servicios de Hosting

- WordPress Blog

NUEVOS DOMINIOS

Lista de nuevos gTLDs favoritos (Watchlist)

Asistencia

- [Solicitar asistencia](#)
Consulta la sección Asistencia y envía tu solicitud
- [Informes sobre las solicitudes de asistencia](#)
Consulta el estado de tus solicitudes de asistencia

Figura 5-29 Nominalia Panel de Control

El Cloud elegido para desplegar la aplicación ha sido DigitalOcean. La elección de este servicio frente a otros es la relación calidad/precio. Se trata de un cloud que te permite tener maquinas virtuales con prestaciones medias por precios mucho más inferiores que otros servicios como AWS.

Para el despliegue se ha utilizado un droplet (maquina en Digital Ocean) con las siguientes prestaciones:

- 1GB de Memoria RAM
- 1 CPU,
- 30 GB de SSD DIKS
- 2TB Transferencias

Las primeras pruebas de despliegue se hicieron en una máquina con la mitad de memoria RAM 512Mb. Los periodos de latencia al realizar red despliegues del sistema se disparaban, por lo que hubo que aumentar a 1Gb la memoria RAM.

El tiempo de despliegue actual está en torno a los 10 segundos.

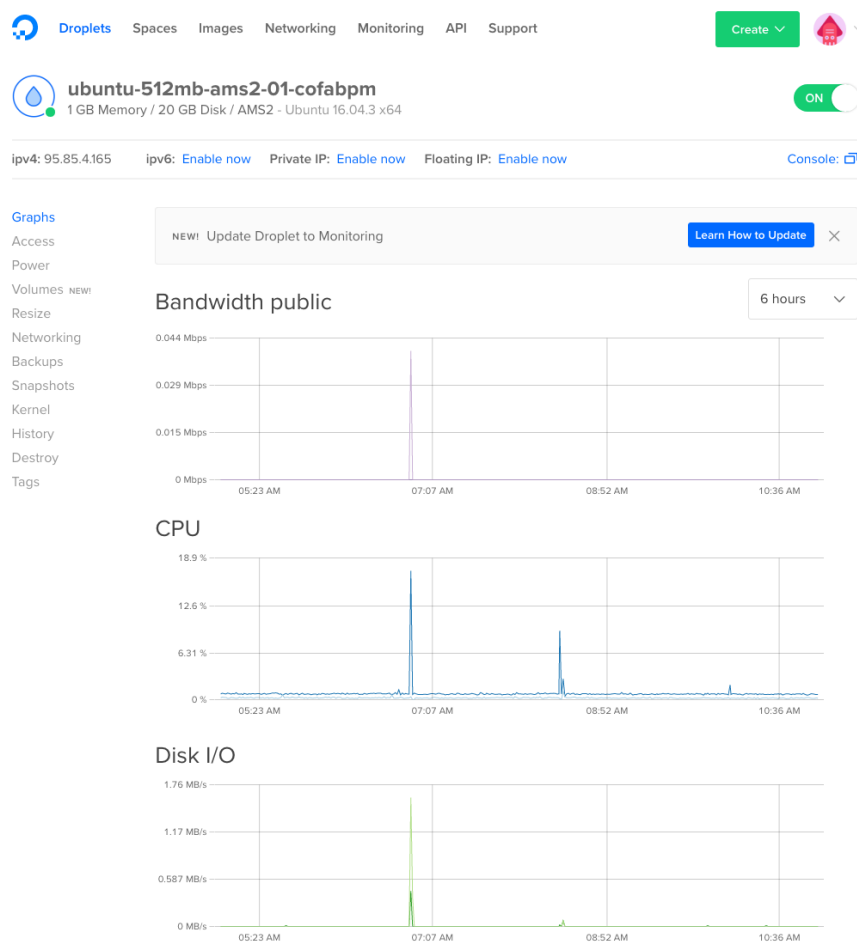


Figura 5-30 Droplet CofabPM Digital Ocean

La máquina tiene instalado como sistema operativo la distribución de Linux Ubuntu 16.04.03 x64.

5.4.3.3. Seguridad

Una de las principales obsesiones que debe tener un desarrollador hoy en día es la seguridad. El conjunto de buenas prácticas a la hora de programar soluciones y de desplegarlas debe ser un mantra para cualquier profesional que quiera desenvolverse en este mundo de forma exitosa.

Un reciente estudio de EMC afirma que cada año se acumula 1,7 billones de dólares de pérdidas por fallos informáticos en todo el mundo.

A la hora de realizar esta solución, no podía ser menos. Por eso se ha tratado este tema con sumo cuidado realizando las siguientes acciones:

Creación de certificado para dominio cofabpm.com

La finalidad es que la página web viaje cifrada a través del protocolo HTTPS. Para crear este certificado se ha usado Let's Encrypt. Se trata de la autoridad de certificación gratuita que emite certificados de tipo X.509 desde principios de 2016. Detrás de esta autoridad de certificación sin ánimo de lucro se encuentran empresas tan importantes como la Fundación Mozilla o Cisco.

Antes había que pagar cantidades muy altas por conseguir tener un certificado para tu dominio. Let's Encrypt ha conseguido democratizar el mundo de la seguridad web convirtiéndose en menos de 2 años en un referente.



Figura 5-31 Conexión Segura



Figura 5-32 de la CA Certificado Let's Encrypt

Uso de Proxy Inverso Nginx

El uso de un proxy inverso en la creación de la solución, provee de una seguridad extra: conseguir anonimato para el despliegue de la aplicación; filtrar todo el tráfico web entrante al servidor para dirigirlo hacia donde se quiera.

Además, es en este punto es donde se usa el certificado de seguridad generado. En el caso de CofaBPM la aplicación se encuentra levantada en el puerto 5000 del sistema. Nginx redirige cualquier tipo de tráfico entrante entrante HTTP (puerto 80) a tráfico HTTPS (puerto 443) para hacer que todo vaya cifrado.

Por último, al tratarse de un proxy inverso y poder dirigir tráfico a varias máquinas, se ha configurado para que todo el tráfico <https://www.cofabpm.com/> sea dirigido internamente al localhost:5000 donde está la instancia de Spring Boot.

Hasta aquí pueden verse las medidas a nivel operacional realizadas para securizar el sistema.

Si a esta configuración le pasamos el prestigioso test de seguridad de Qualys. SSL Labs <https://www.ssllabs.com> obtenemos una de las certificaciones más altas posibles: A+

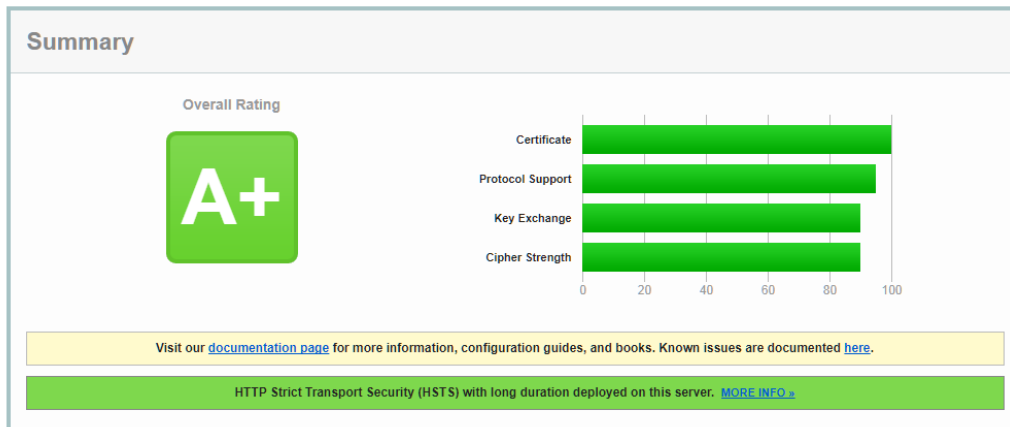
You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > www.cofabpm.com**SSL Report: www.cofabpm.com (95.85.4.165)**Assessed on: Mon, 11 Sep 2017 12:55:38 UTC | [Hide](#) | [Clear cache](#)[Scan Another »](#)

Figura 5-33 Nota de Qualys SSL Labs

La otra medida de seguridad aportada viene de la parte del desarrollo:

Incluyendo Spring Security en la Aplicación

En el próximo apartado se tratará con más detalle la parte BackEnd de la aplicación, pero en este punto vamos a definir la parte de seguridad.

Para el uso de Spring Security con Spring Boot hay que añadir la siguiente dependencia.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

La autenticación en la aplicación va a venir ligada a los diferentes roles que pueden actuar en ella. Por tanto, en la configuración global se va a dar de alta cada usuario y su rol. Realizamos esta configuración directamente en memoria.

Se crea la clase `WebSecurityConfig.java`. Usamos las anotaciones:

`@Configuration` para indicar a Spring que es una clase contenedor a de definiciones Beans.

@EnableWebSecurity esta anotación activa SpringSecurity.

La configuración realizada es para cualquier acceso que se realiza tenga que estar autorizado. A excepción de los recursos públicos necesarios como imágenes, css y librerías js de terceros como Angular.

El resto de direcciones deben pasar por el formulario de Login. Además, está activo el control CSRF a través de token.

WebSecurityConfig.java

```
package com.uah.cofabpm;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.authentication
    .builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration
    .WebSecurityConfigurerAdapter;
import org.springframework.security.web.csrf.CookieCsrfTokenRepository;
import org.springframework.security.config.annotation
    .web.configuration.EnableWebSecurity;

@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
            .antMatchers("/library/**", "/css/**", "/img/**").permitAll()
            .anyRequest().authenticated()
            .and()
            .csrf()
            .csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse())
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll()
            .and()
            .logout()
            .permitAll();
    }
}
```

```
@Autowired
public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
    auth.inMemoryAuthentication()
        .withUser("Pedro").password("Pedro2017").roles("JefeProyecto");
    auth.inMemoryAuthentication()
        .withUser("Salvador").password("Salvador2017").roles("ProductOwner");
    auth.inMemoryAuthentication()
        .withUser("Ana").password("Ana2017").roles("ResponsableAnalista");
    auth.inMemoryAuthentication()
        .withUser("Jorge").password("Jorge2017").roles("ResponsableDesarrollo");
    auth.inMemoryAuthentication()
        .withUser("Maria").password("Maria2017").roles("ResponsableTesting");
}
}
```

En la configuración se crean 5 usuarios con 5 roles diferentes. Estos roles son relevantes en el uso de CofaBPM puesto que servirán para mostrar las tareas pendientes de su grupo en los buzones.

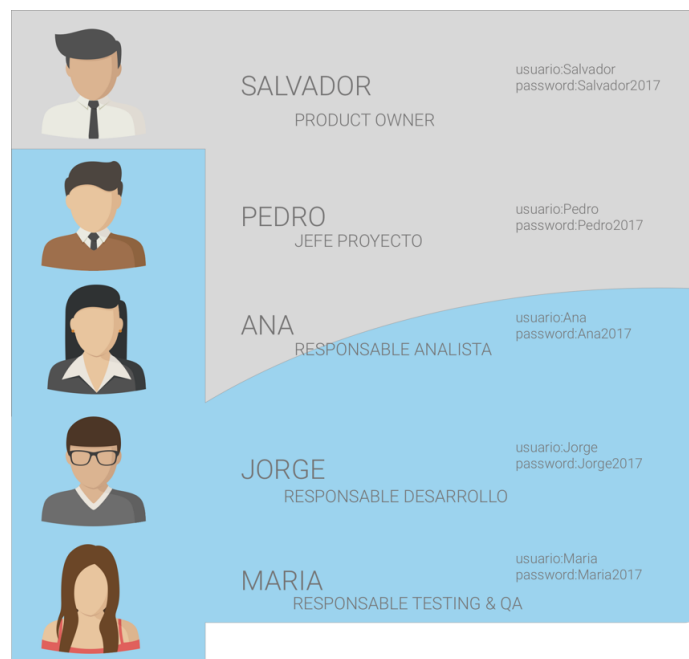


Figura 5-34 Grupos de Participantes

Problemática Enfrentada al combinar Nginx con SpringSecurity

El primer despliegue de CofaBPM se hizo con la securización HTTPS y Nginx sin incluir SpringSecurity. El funcionamiento del proxy era correcto y se alcanzaba la aplicación sin problemas. Posteriormente se introdujo en el desarrollo SpringSecurity. Las pruebas realizadas en local sobre SpringSecurity fueron correctas. Sin embargo, al subir al entorno de producción la

aplicación, empezó a dar dos errores: a veces marcaba el error 502 Bad Gateway y otras veces el error 504 Gateway timeout.

Tras investigar el motivo de estos fallos y preguntar en varios foros, se descubrió que los problemas tenían que ver con la cookie de seguridad CSRF para evitar los ataques de sitios cruzados. Nginx no bajaba hasta SpringSecurity dicha cookie.

Para que funcione correctamente, es necesario incluir dos parámetros más en la configuración de Nginx: uno para indicar que baje la cookie hasta la aplicación origen y otro cuál es la aplicación origen.

```
# Pass the csrf token (see https://de.wikipedia.org/wiki/Cross-Site-Request-Forgery)  
  
# Default in Spring Boot and required. Without it nginx suppresses the value  
proxy_pass_header X-XSRF-TOKEN;  
  
# Set origin to the real instance, otherwise a of Spring security check will fail  
  
# Same value as defined in proxy_pass  
proxy_set_header Origin "http://localhost:5000";
```

5.4.3.4. BackEnd del Sistema.

Utilizando un proyecto de Maven se ha creado el artefacto com.uah.cofabpm. El pom completo y el resto de de código puede consultarse en el [Anexo A](#).

Las dependencias manejadas en el pom son:

Dependencia	Descripción
activiti-spring-boot-starter-basic	SpringBoot con Activiti integrado
spring-boot-starter-thymeleaf	Integración de Thymeleaf para SpringBoot
activiti-spring-boot-starter-actuator	API REST de Activiti
activiti-spring-boot-starter-jpa	Soporte JPA para Activiti
h2	Base de datos H2
spring-boot-starter-security	Spring Security
commons-io	Utilidades Comunes

La estructura de carpetas del Proyecto queda definida así:



Figura 5-35 Estructura de Carpetas del Proyecto

src/main/java/com/uah/cofabpm

Contiene las clases de la aplicación del paquete com.uah.cofabpm

src/resources/processes

Los BPMS a desplegar. Cualquier BPM que se encuentre en esta carpeta es desplegado automáticamente.

src/resources/static

Frontend

src/templates

Templates de thymeleaf que son dos: login y el index que carga la aplicación Angular.

5.4.3.4.1. Configuración de la Aplicación

La configuración de Activiti y Spring se simplifica aún más a la hora de usar Spring Boot.

En esta conjunción el fichero de propiedades de Activiti desaparece y las configuraciones pueden escribirse directamente en application.properties que Spring Boot carga por defecto con todas las propiedades de la aplicación.

application.properties

```
server.port=5000
spring.activiti.mailServerHost=authsmtp.cofabpm.com
spring.activiti.mailServerPort=587
spring.activiti.mailServerDefaultFrom=info@cofabpm.com
spring.activiti.mail-server-user-name=info@cofabpm.com
spring.activiti.mailServerPassword=*****
endpoints.shutdown.enabled=true
management.security.enabled=false
```

En el fichero de propiedades de la aplicación marca que la aplicación debe correr en el puerto 5000, se dan los parámetros para la configuración del servidor email utilizado por las EmailTask. También se aplican dos configuraciones relacionadas con la API rest donde se indica que el motor puede ser apagado desde un endpoint y no activada la seguridad para tareas de management.

El paquete de la aplicación es com.uah.cofabpm

Tal y como puede verse en el diagrama UML la clase principal es Application.java. Esta es la encargada de ejecutar SpringBoot.

En este punto es donde se ve el potencial de la integración de SpringBoot con Activiti.

Como se ha visto en el ejemplo del capítulo anterior donde se creaba un BPM con Activiti desde 0. Había que realizar una serie de pasos que eran:

- Configurar la Base de Datos y arrancarla.
- Crear un Motor de Procesos.
- Desplegar el BPM seleccionado en el Motor de Procesos.

Al usar el componente de integración de Activiti con Spring surge “la magia” y todo eso queda reducido a la anotación `@EnableAutoConfiguration`.

Esta anotación realiza automáticamente lo siguiente:

- Configuración de la Base de Datos y su arranque (por defecto H2).
- Crea el motor de proceso.
- Despliega cualquier BPM que encuentre en la carpeta `resources/processes`

Como puede observarse esto simplifica de forma bastante significativa el trabajo a realizar.

Además, si fuera necesario realizar alguna configuración extra o algún tipo de tratamiento al inicio del proceso se cuenta con el objeto `CommandLineRunner`. En el caso de CofaBPM se utiliza este Bean para mostrar el nombre y la versión del motor en ejecución.

Por otro lado, la anotación `@ComponentScan` escanea todos los componentes para ver las directivas de tipo `@Configuration` para su uso

.

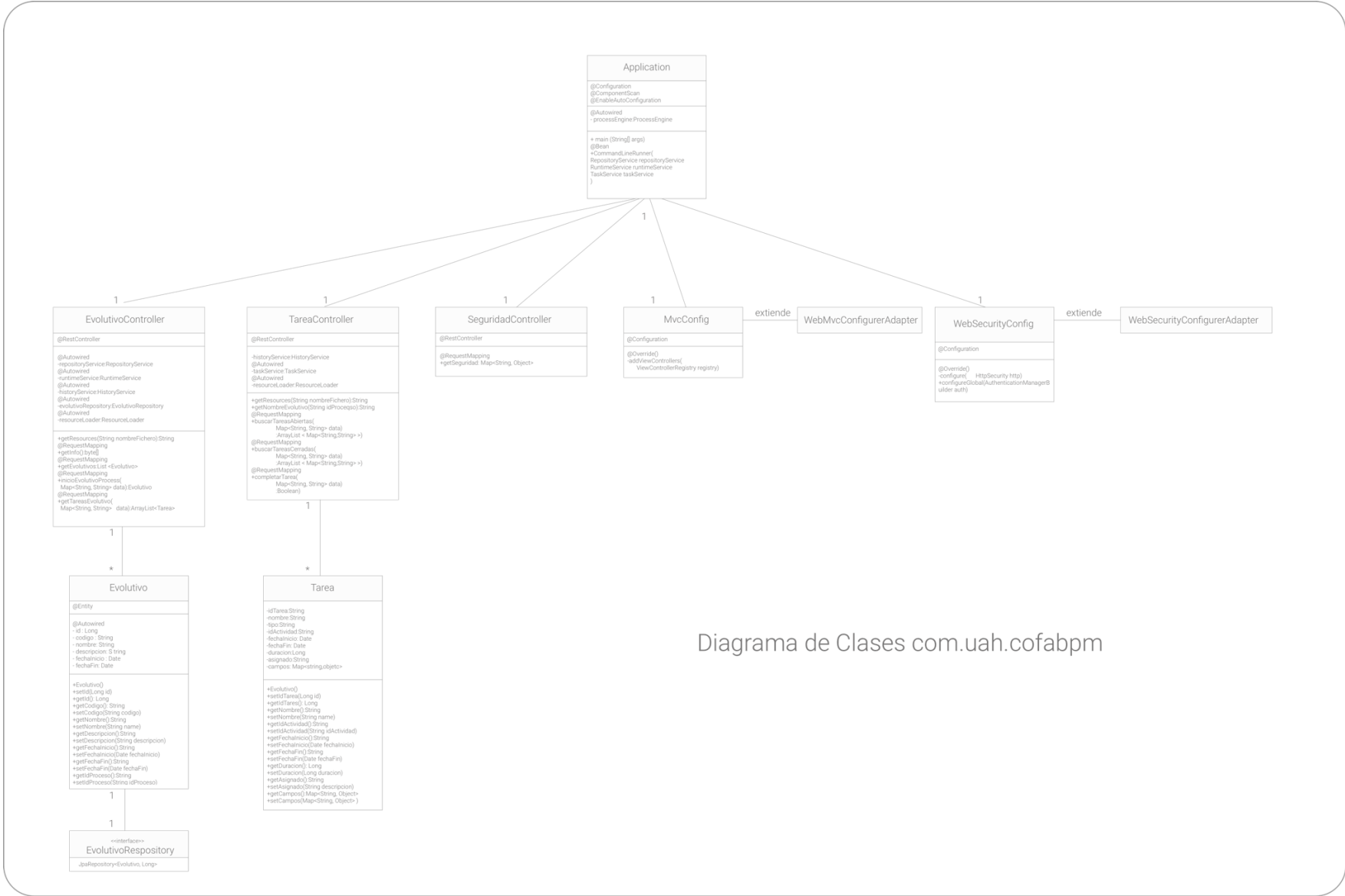


Diagrama de Clases com.uah.cofabpm

Figura 5-36 Diagrama de Clases

La aplicación contiene tres controladores que exponen sus operaciones a través de servicios REST (contienen la anotación `@RestController`).

El primer controlador es el de seguridad (`SeguridadController.java`). Este controlador expone un servicio que devuelve el usuario y rol de la sesión de Spring Security conectada.

El controlador `EvolutivoController.java` maneja la lógica de negocio relacionada con los evolutivos. Este controlador es el encargado de exponer las siguientes operaciones:

1. `api/evolutivos`

Devuelve la información de todos los evolutivos. Para ello hace uso de la query del histórico de proceso y busca todas las tareas finalizadas de “Apertura de Proyecto”. El hecho de utilizar el histórico en vez de ir contra el motor en ejecución tiene que ver con la optimización de Activiti. Como buena práctica, aconsejan minimizar al máximo las consultas sobre Process Execution para no penalizarlo.

2. `api/abrirevolutivo`

Realiza la apertura de un evolutivo. Tal y como se describió en el apartado de la creación de BPM, este tiene unas tareas encargadas de enviar información por email. Estos emails tienen una plantilla que consta de cabecera, cuerpo del mensaje y pie. La cabecera y el pie son invariables y el cuerpo del mensaje es redactado dentro del BPM haciendo uso de los campos que se van informando.

En esta clase, se cargan los recursos `templateHeader` y `templateFooter` y se pasan como variables al BPM para su uso.

Para la apertura del proyecto, es necesario instanciar un nuevo BPM del tipo Evolutivo y comenzarlo.

```
ProcessInstance procesoNuevo = runtimeService.startProcessInstanceByKey("BPM_Evolutivo");
```

Una vez creado un nuevo proceso, se complete la primera tarea que es alta del evolutivo. Para completarla se pasa los templates del email y un objeto JPA del tipo Evolutivo. Aquí es donde se ve la potencia de Activiti manejando datos complejos. La tarea es reclamada por el usuario y se completa.

```
Map<String, Object> taskVariables = new HashMap<String, Object>();  
taskVariables.put("evolutivo", evolutivo);  
taskVariables.put("templateHeader", templateHeader);  
taskVariables.put("templateFooter", templateFooter);  
taskService.setVariablesLocal(tarea.getId(), taskVariables);  
taskService.claim(tarea.getId(), usuario);  
taskService.complete(tarea.getId(), taskVariables);
```

3. api/evolutivoesado

Dado un id de proceso, devuelve toda la información de ese evolutivo. La salida son todas las tareas y los campos con los que se han informado.

Para ello se hace uso de dos queries del motor.

La primera para conseguir todas las actividades (instancias de tareas) que se han realizado en este proceso.

```
List<HistoricActivityInstance> activities = historyService.createHistoricActivityInstanceQuery()  
    .processInstanceId(idProceso).orderByHistoricActivityInstanceStartTime().asc().list();
```

Después para cada actividad se busca las variables que se han informado en el formulario.

```
List<HistoricVariableInstance> variablesTarea = historyService  
    .createHistoricVariableInstanceQuery().taskId(activity.getTaskId()).list();
```

4. api/información

Devuelve la imagen en formato del BPM cargado y muestra información por consola. Esta imagen se trata de un recurso renderizado por el motor de Activiti y se usa para conseguirlo este método

```
String diagramResourceName = processDefinition.getDiagramResourceName();  
  
InputStream imageStream = repositoryService  
    .getResourceAsStream(processDefinition.getDeploymentId(),
```



Figura 5.32. api/información devolviendo imagen del BPM

Lista devuelta por el rutimeServe: [ProcessInstance[27], Execution[id '28'] - activity 'UT2_Entrega_DF - parent '27', ProcessInstance[5], Execution[id '6'] - activity 'UT2_Entrega_DF - parent '5']
Número de procesos definidos : 1
Número de tareas : 2
Número de tareas después de empezar: 2
Nombre de tareas: [Task[id=24, name=Entrega Diseño Físico], Task[id=46, name=Entrega Diseño Físico]]
Lista devuelta por el rutimeServe: [ProcessInstance[27], Execution[id '28'] - activity 'UT2_Entrega_DF - parent '27', ProcessInstance[5], Execution[id '6'] - activity 'UT2_Entrega_DF - parent '5']

5. api/prueba

Por último, este servicio expuesto de prueba se ha usado con diferentes finalidades durante el desarrollo. Por ejemplo, hacer determinadas pruebas a queries intermedias. No tiene validez en producción, pero se ha dejado expuesto para futuros desarrollos.

La entidad de JPA utilizada se trata del POJO **Evolutivo.java**. Contiene la información del Evolutivo que se informa en la primera tarea del proceso, cuando se da de alta.

Esta entidad lleva asociada la clase `EvolutivoRepository.java` que extiende la clase `JPARepository` para su grabación en el repositorio. Usa metadatos necesarios para la grabación.

El siguiente controlador REST expone operaciones sobre las tareas (**TareaController.java**). También hace uso un POJO para almacenar la información de las tareas, pero este no se ha configurado para JPA. Al igual que la clase anterior, también carga las plantillas de los emails que son pasados a la hora de completar las tareas.

Los servicios que expone son:

1. api/buscartareasabiertas

Utiliza como entrada el grupo de un usuario y busca todas las tareas de cualquier evolutivo que puedan estar asociadas a este grupo de usuarios. Para ello se apoya en la query de tareas usando el método: `()taskCandidateGroup(grupo)`.

2. api/buscartareacerradas

La consulta de tareas cerradas se hace usando como entrada el usuario. Devuelve todas las tareas de todos los evolutivos en los que el usuario ha participado. Para ello se apoya en el método: `()taskAssignee(usuario)`.

3. api/completartarea

Este servicio se encarga de ir avanzando el BPM completando las tareas. De entrada, recibe el id de la tarea, el usuario y el grupo y los campos para completar la tarea.

Reclama la tarea para ese usuario y la completa pasando al BPM los campos recibidos.

El método utilizado es: `taskService.complete(tareaid,taskVariables)`;

El POJO utilizado para modelar las tareas es **Tarea.java** y contiene la información de la tarea: campos, inicio, fin, usuario que la ha llevado a cabo. Este POJO no se ha modelado como una `@Entity` de JPA porque el uso de Hibernate en el caso del Evolutivo era meramente teórico para demostrar su uso.

La última clase por comentar es **MVConfig.java**

Se trata de una clase `@Configuration` que extiende la clase `WebMvcConfigurationAdapter` de `SpringMVC`.

En ella se hace un `@Override` al método `addViewControllers` en el cual se registran las dos vistas utilizadas por el Front. Cada una de estas vistas es renderizada a través de `Thymeleaf`. Se trata de un motor de renderizado en el lado del servidor.

Se asocian dos direcciones la dirección `/login` a la vista `login` y la dirección `/` a la vista `index`. Estas vistas coinciden con sendos archivos HTML que se encuentran en la subcarpeta `templates` de `resources` donde va a buscarla.

El funcionamiento es el siguiente desde `SpringMVC` se puede acceder al `login` o al `index`. Para acceder al `index`, es necesario haber pasado por el `login` tal y como se vió en el capítulo anterior con la securización de Spring. Cuando el `login` es satisfactorio se va a la ruta raíz `/` desde donde se carga el `index.html`.

Una vez servido el `index.html` su contenido es una SPA en Angular que ya no hace recargas totales de la aplicación, sino que usa sus rutas y sus propios controladores recargando partes parciales de la pantalla. En este punto y tras la carga inicial del `index`, ni `Thymeleaf`, ni `SpringMVC` está haciendo trabajo en la parte front.

Esta separación de la parte de `login` y la parte de angular es una medida más de seguridad para no cargar ningún recurso de la aplicación hasta pasar la barrera de entrada.

5.4.3.5. FrontEnd del Sistema

La parte Front, tal y como se ha explicado consta de dos partes: la pantalla de login y la aplicación de Angular orquestando ambas desde SpringMVC y renderizando la pantalla de login y el punto de entrada a la aplicación Angular desde Thymeleaf..

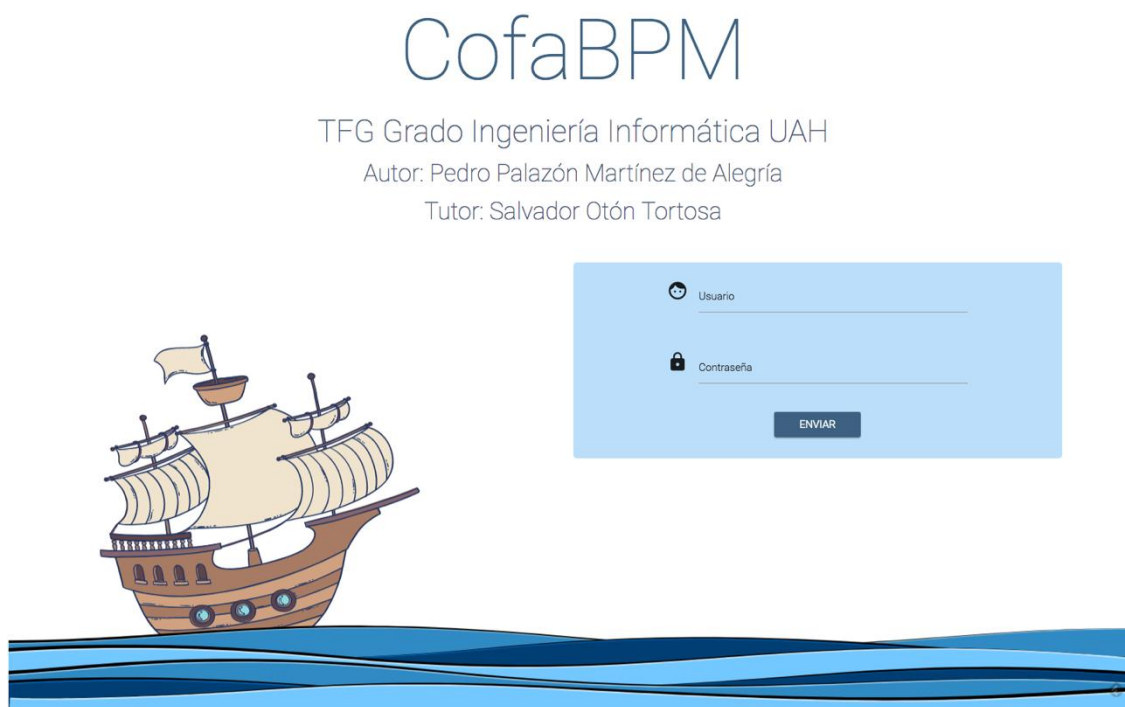


Figura 5-37 Pantalla de Login

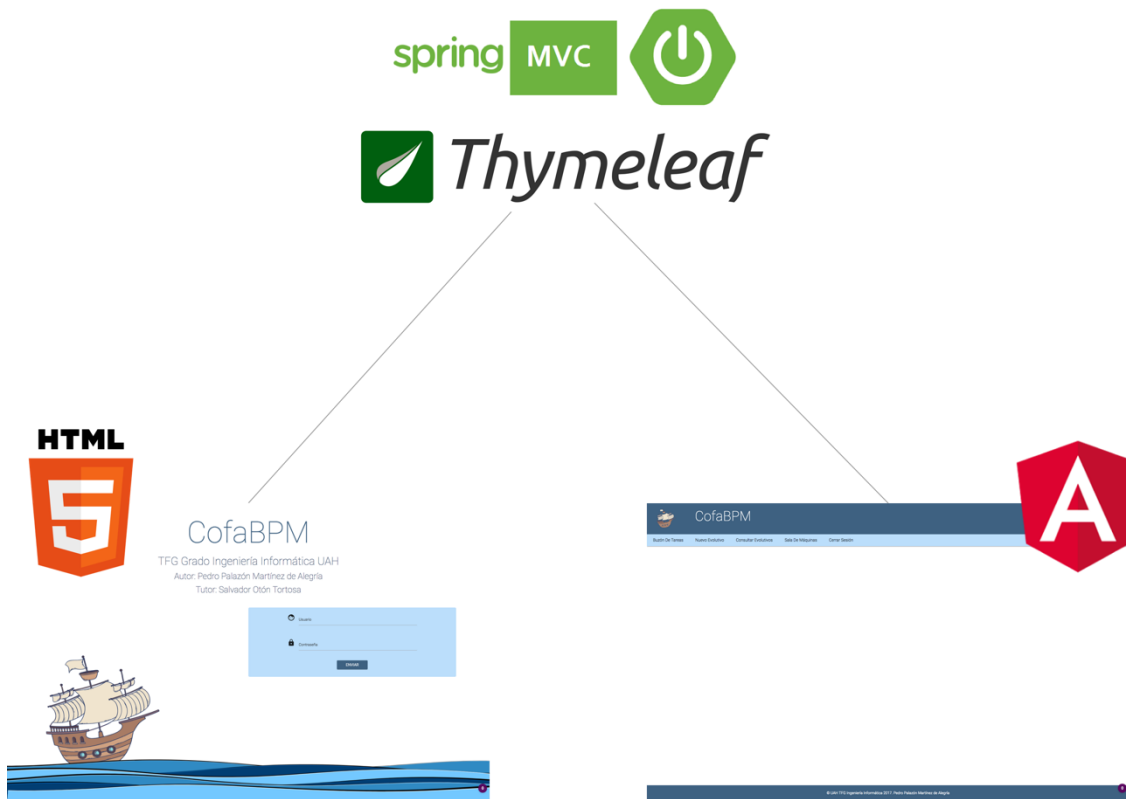


Figura 5-38 Uso de SpringMVC

Uxer Experience / User Interface

A la hora de abordar este proyecto, en la parte de conceptualización se tuvo claro que CofaBPM no debía ser una herramienta más de gestión de ciclo de vida. Con innumerables pestañas y un sin fin de campos a informar de forma rígida.

Esta herramienta se crea con el fin de **facilitar** la comunicación entre los participantes. Es por ello que se ha optado por un diseño visual limpio y cómodo. Haciendo uso de colores cálidos. Con mucho espacio entre los componentes. Usando el mínimo número de botones. Potenciando el lenguaje visual con micro animaciones que marquen las transiciones entre estados de la pantalla e interacciones con el usuario.

El diseño se ha realizado teniendo en cuenta las guías de diseño Material Design publicadas por Google.

Uno de los Framework de UI más importantes que siguen estas guías es Materialize.

Materialize es desarrollado por un grupo de estudiantes de la Carnegie Mellon University. Para muchos desarrolladores Materialize es mucho mejor que la propia librería Material de Google. Incluso encontrándose en versiones tempranas (v0.100.2).

La aplicación tiene tres áreas definidas:

1. Cabecera de la Aplicación

A su vez tiene dos secciones: una superior que contiene el logotipo y nombre a la izquierda y la información del usuario identificado a la derecha. La sección inferior es la barra de menú de la aplicación. Ambas son invariables durante todas las vistas.

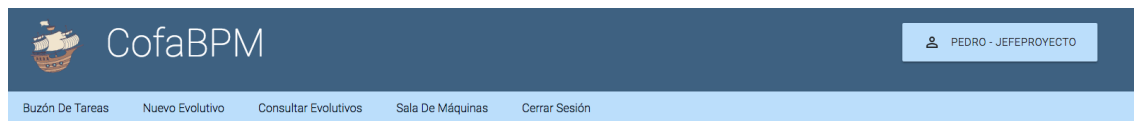


Figura 5-39 Cabecera de la Aplicación

2. Vista Central

Parte central donde se irán cargando las diferentes vistas de la aplicación dependiendo del estado

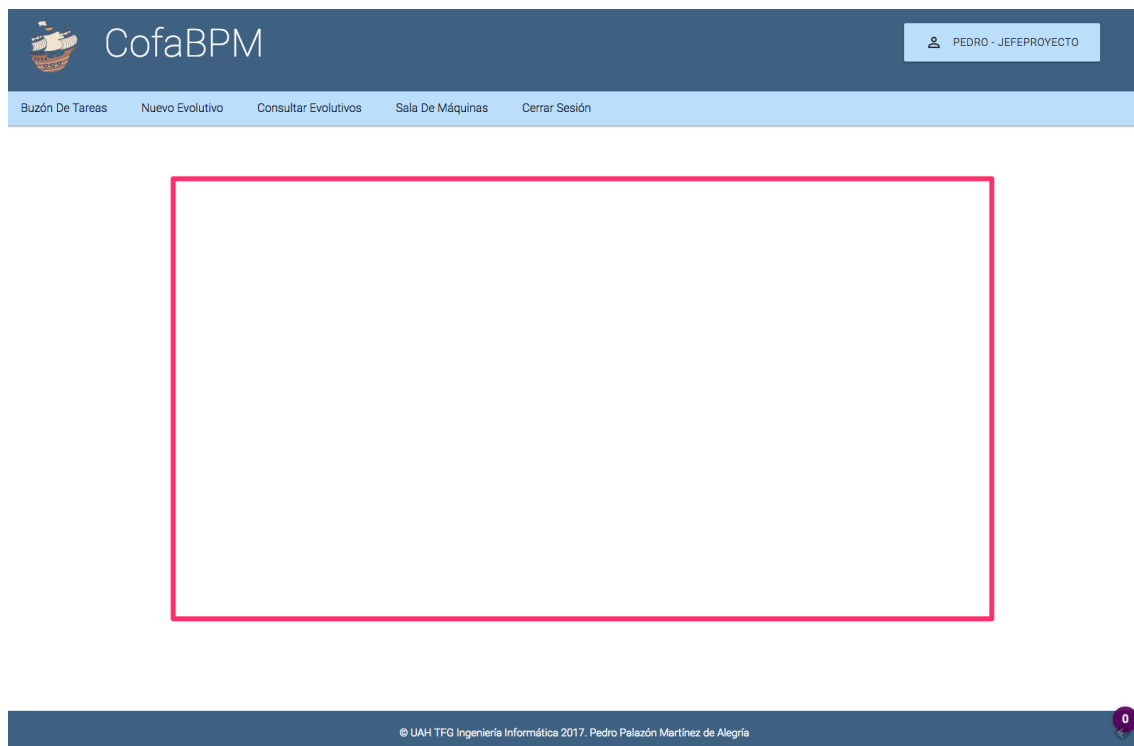


Figura 5-40 Cuerpo de la Aplicación

3. Pie de la Aplicación

Información de la fecha y creador.



Figura 5-41 Pie de la Aplicación

Estructura del Front CofaBPM

Una SPA Angular por sí, está basada en una arquitectura Modelo Vista Controlador. En el caso de CofaBPM cada controlador maneja una vista (parte parcial de HTML cargada) pero se ha procurado exponer en los ficheros de JavaScript la mínima lógica de negocio. Dejando está función a los controladores de Java. Los controller de Angular, básicamente tienen un uso para el manejo del comportamiento de la pantalla. A cada vista/controlador le corresponde un estado. Estos estados son orquestados desde el fichero app.js el módulo de Angular propiamente dicho.

En muchas ocasiones las aplicaciones Angular son escritas casi como si código espagueti se tratará. Todos los controladores uno, detrás de otro en el mismo fichero. Como buena práctica de desarrollo de aplicaciones Angular, está la de separar cada controlador en un fichero y asociarlo a una sola vista. Esto produce un código más legible y mantenible a futuro. Base primordial del **código limpio**.

Además, la aplicación Angular contiene dos ficheros más:

- **service.js** aquí se realiza las llamadas asíncronas a todos los servicios REST utilizados por la aplicación.
- **filter.js** que alberga directivas de filtros que pueden usarse en las pantallas. Por ejemplo, transformación a camelCase un string.

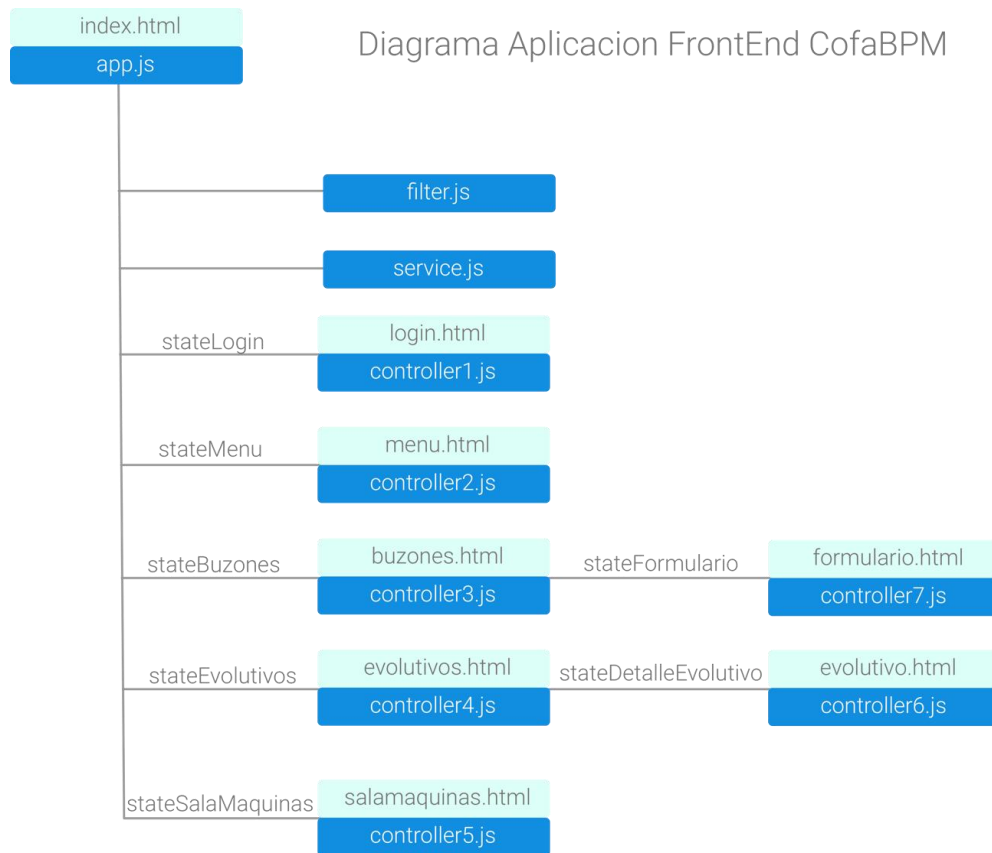


Figura 5-42 Diagrama de la Aplicación Front

En el esquema puede verse un stateLogin. Este estado no es el login real de la aplicación. Es estado no se usa, se mantiene dentro de la aplicación por si en algún momento se quisiese usar esta de forma independiente de SpringSecurity.

stateMenu

Una vez autenticados de forma correcta se accede a la pantalla de menú. En ella se muestra un pequeño manual “onboarding” de uso y se puede ir a cualquiera de las opciones disponibles.

Se trata de un componente UI de tipo carrusel donde se va mostrando cada una de las opciones disponibles.

No hace uso de ningún servicio REST.

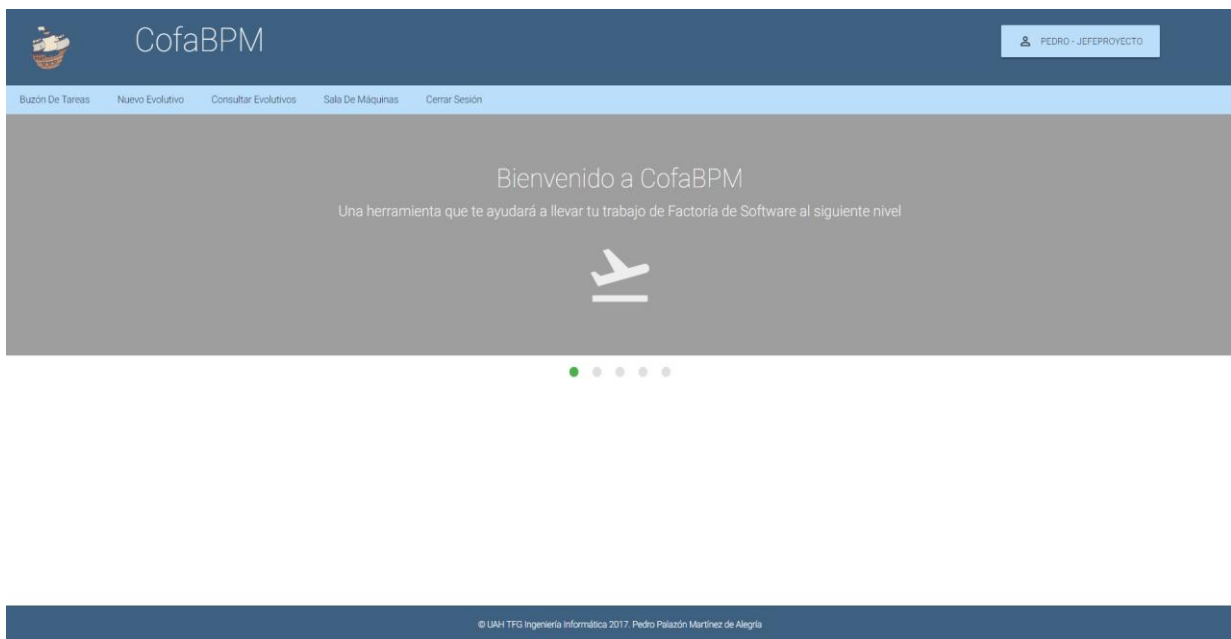


Figura 5-43 Menu

stateBuzones

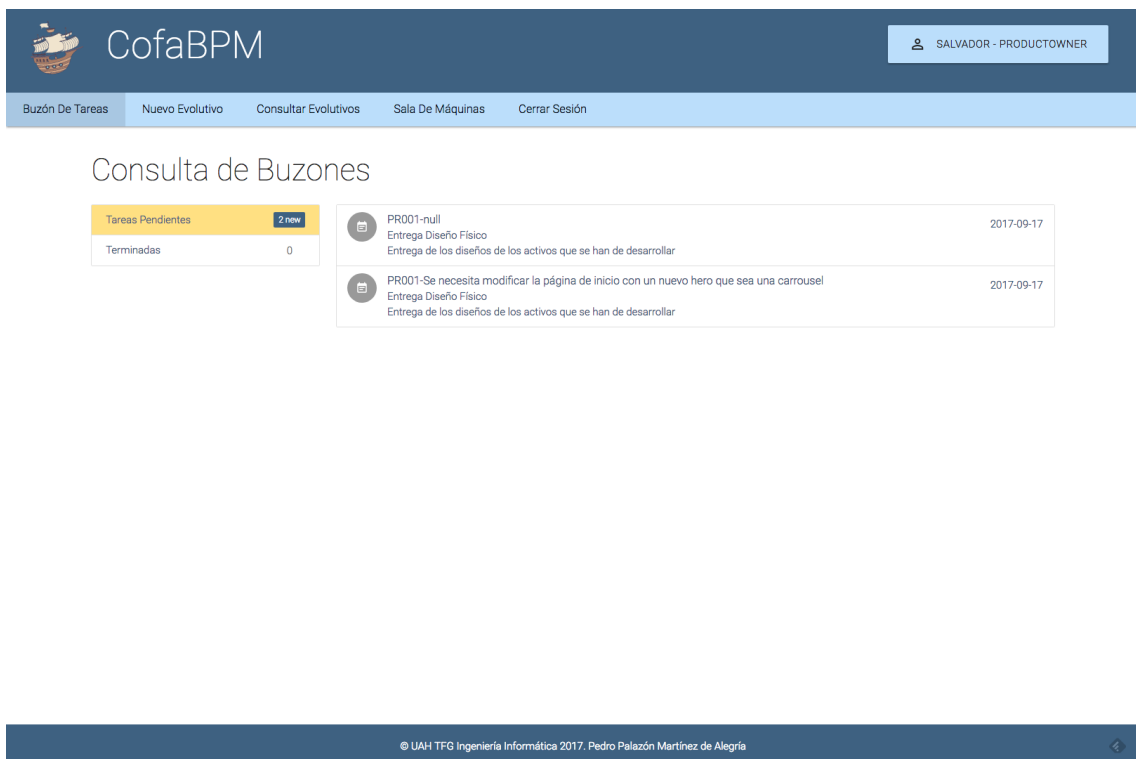
En este estado pueden consultarse los buzones de las tareas pendientes y de las tareas terminadas. Si se pulsa una tarea pendiente se accede a su formulario de trabajo para ser completada. En este punto se pasa al siguiente al estado stateFormulario. A este estado, le pasa como parámetros el proceso, la tarea y el tipo de tarea a completar.

Las tareas pendientes son a nivel de rol. Es decir “dos jefes de proyecto” van a ver las mismas tareas. Cuando la tarea se completa por uno de ellos, esta le desaparece al otro.

Por otro lado, las tareas terminadas si que son a nivel individual y cada usuario ve exclusivamente las suyas.

Los servicios expuestos que utiliza este estado son:

- /api/buscartareasabiertas
- /api/buscartareascerradas



CofaBPM SALVADOR - PRODUCTOWNER

Buzón De Tareas | Nuevo Evolutivo | Consultar Evolutivos | Sala De Máquinas | Cerrar Sesión

Consulta de Buzones

Tareas Pendientes	2 new
Terminadas	0

PR001-null Entrega Diseño Físico Entrega de los diseños de los activos que se han de desarrollar	2017-09-17
PR001-Se necesita modificar la página de inicio con un nuevo hero que sea una carousel Entrega Diseño Físico Entrega de los diseños de los activos que se han de desarrollar	2017-09-17

© UAH TFG Ingeniería Informática 2017. Pedro Palazón Martínez de Alegría

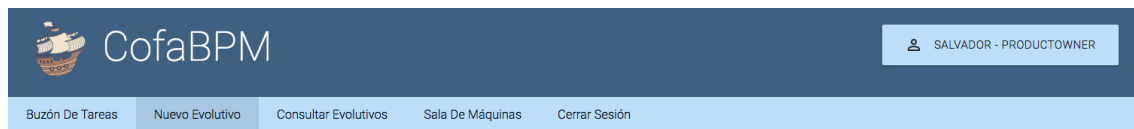
Figura 5-44 Consulta de Buzones

stateFormulario

Este estado tiene como entrada el tipo de tarea a completar. Dependiendo de este parámetro, muestra el formulario necesario y modifica el título de la vista. En total el evolutivo tiene 11 formularios. Al usar la misma vista, se consigue reutilizar el máximo posible el código. Algunos formularios son incluso el mismo.

Los servicios expuestos que utiliza este estado son:

- /api/abrirrevolutivo
- /api/completartarea



Formulario Tarea de Usuario: Entrega Diseño Físico

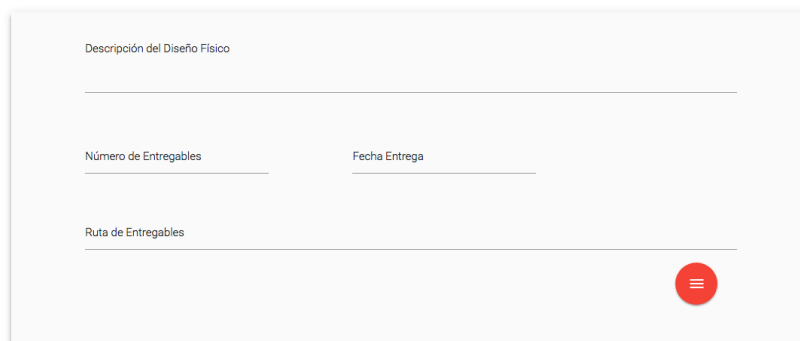


Figura 5-45 Formulario tareas de usuario: Entrega Diseño Físico

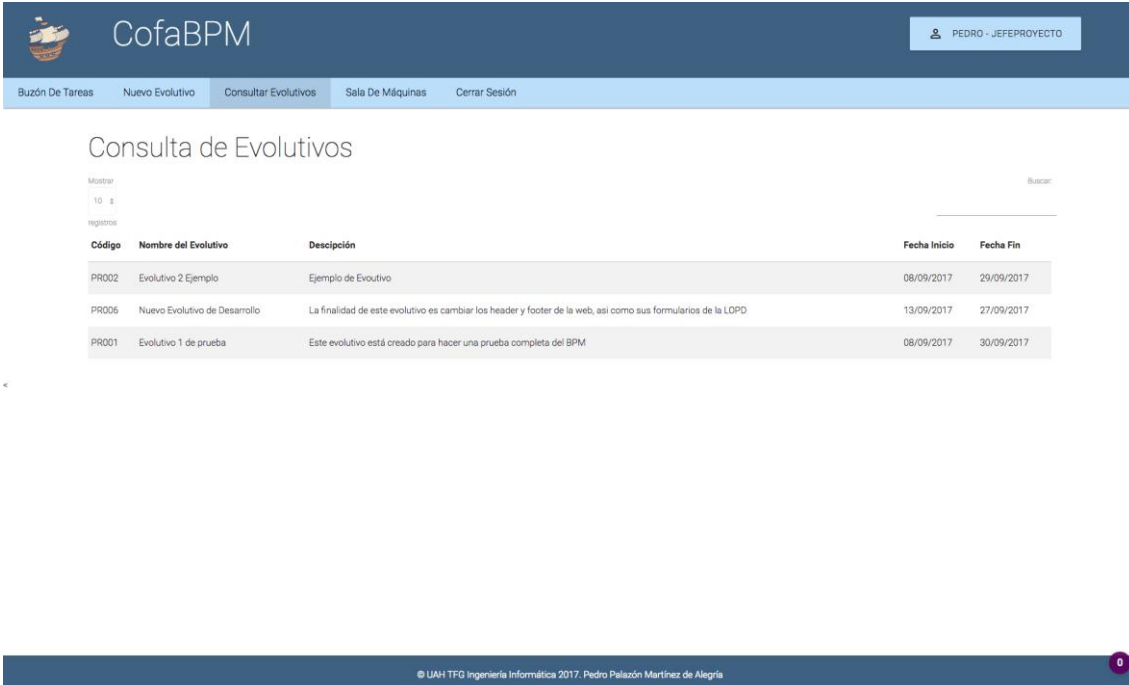
stateEvolutivos

Desde este estado se consulta en una tabla paginable y con posibilidad de filtrado los evolutivos dados de alta en el sistema. Para realizar esta tabla se ha utilizado la librería angularDataTables. Se trata de una adaptación para Angular de la mundialmente extendida Datatables de jQuery.

Si selecciona una fila de evolutivo, se pasa al siguiente estado: stateDetalleEvolutivo. Al pasar a este estado, se envía como parámetros la información de la tarea que contiene la pantalla.

El servicio expuesto que utiliza este estado es:

- /api/evolutivos



Mostrar 10 registros

Código	Nombre del Evolutivo	Descripción	Fecha Inicio	Fecha Fin
PR002	Evolutivo 2 Ejemplo	Ejemplo de Evolutivo	08/09/2017	29/09/2017
PR006	Nuevo Evolutivo de Desarrollo	La finalidad de este evolutivo es cambiar los header y footer de la web, así como sus formularios de la LOPD	13/09/2017	27/09/2017
PR001	Evolutivo 1 de prueba	Este evolutivo está creado para hacer una prueba completa del BPM	08/09/2017	30/09/2017

© UAH TFG Ingeniería Informática 2017. Pedro Palazón Martínez de Alegría

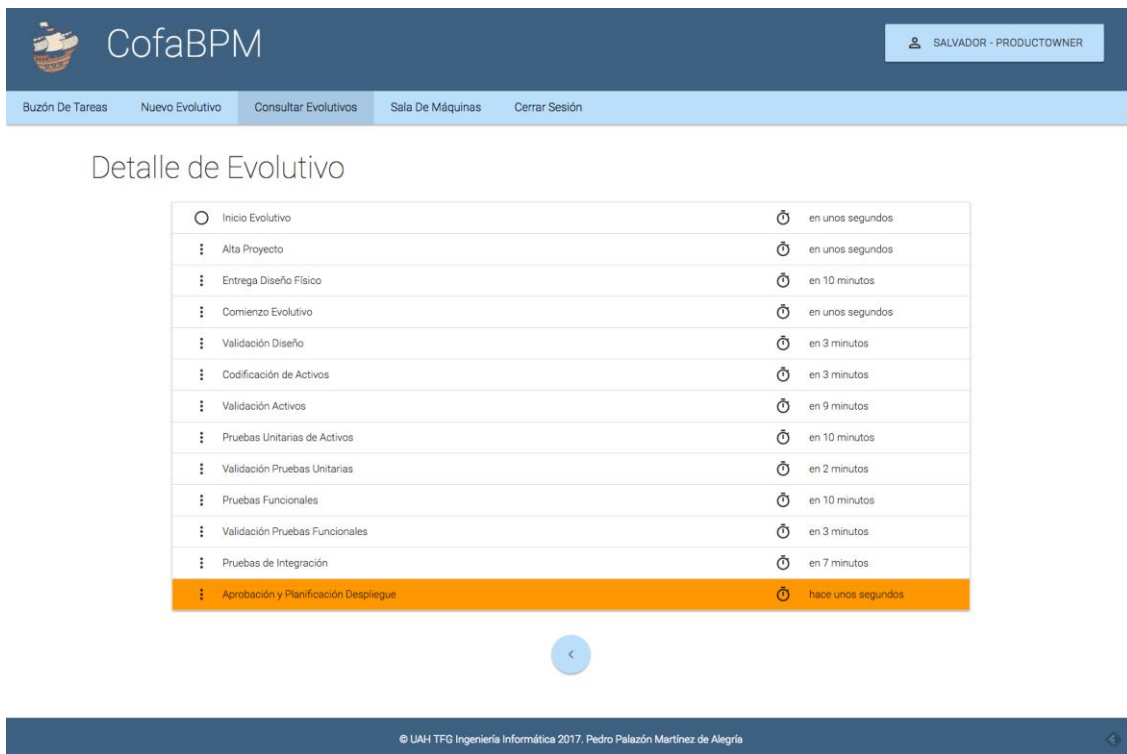
Figura 5-46 Consulta de Evolutivos

stateDetalleEvolutivo

En este detalle se muestra cada una de las tareas que conforman el evolutivo. Se ha utilizado un componente de acordeón desplegable para ver la información de cada tarea. Cuando una tarea del evolutivo esté sin terminar, esta aparece en color anaranjado. Se puede ver la información de cuánto tiempo se ha tardado en completar, fecha y hora de comienzo, fecha y hora de fin y cada uno de los campos informados en los formularios.

El servicio expuesto que utiliza este estado es:

- /api/evolutivoestado



Detalle de Evolutivo

○ Inicio Evolutivo	🕒 en unos segundos
⋮ Alta Proyecto	🕒 en unos segundos
⋮ Entrega Diseño Físico	🕒 en 10 minutos
⋮ Comienzo Evolutivo	🕒 en unos segundos
⋮ Validación Diseño	🕒 en 3 minutos
⋮ Codificación de Activos	🕒 en 3 minutos
⋮ Validación Activos	🕒 en 9 minutos
⋮ Pruebas Unitarias de Activos	🕒 en 10 minutos
⋮ Validación Pruebas Unitarias	🕒 en 2 minutos
⋮ Pruebas Funcionales	🕒 en 10 minutos
⋮ Validación Pruebas Funcionales	🕒 en 3 minutos
⋮ Pruebas de Integración	🕒 en 7 minutos
⋮ Aprobación y Planificación Despliegue	🕒 hace unos segundos

© UAH TFG Ingeniería Informática 2017. Pedro Palazón Martínez de Alegría

Figura 5-47 Detalle del Evolutivo

stateSalaMaquinas

Este estado utiliza para mostrar el estado del motor BPM. Aparece la información relacionada con el motor: el estado, los procesos cargados, procesos que tienen instancias, el número de tareas completadas y el número de tareas en curso.

Además, muestra en un pop-up los diagramas de los BPMS cargados a título informativo.

Para la creación de esta pantalla se ha utilizado un servicio rest nativo de Activiti-REST cargado.



CofaBPM

SALVADOR - PRODUCTOWNER

Buzón De Tareas Nuevo Evolutivo Consultar Evolutivos **Sala De Máquinas** Cerrar Sesión

Sala de Máquinas

Estado del Motor Activiti BPM:
Correcto y en Ejecución

1	2	2	0	11
Procesos Definidos	Evolutivos en Ejecucion	Tareas Pendientes	Evolutivos Completados	Tareas Completadas Hoy

BPM_Evolutivo (v1)
BPM Encargado de modelar el servicio del ciclo de vida de un evolutivo desde su entrada en Factoría hasta su entrega a cliente

© UAH TFG Ingeniería Informática 2017. Pedro Palazón Martínez de Alegría

Figura 5-48 Sala de Máquinas

El servicio expuesto que utiliza este estado es:

- /activiti

stateCerrarSesion

Este último es un estado sin controlador ni vista. Funcionalmente anula la cookie de la sesión, borra los datos almacenados en la localStorage. Llama al servicio de logout() de SpringSecurity y redirige la aplicación a un pantalla de salida. Esta pantalla está fuera de la aplicación Angular descargando todo el contexto de la misma.

El servicio expuesto que utiliza este estado es:

- /logout

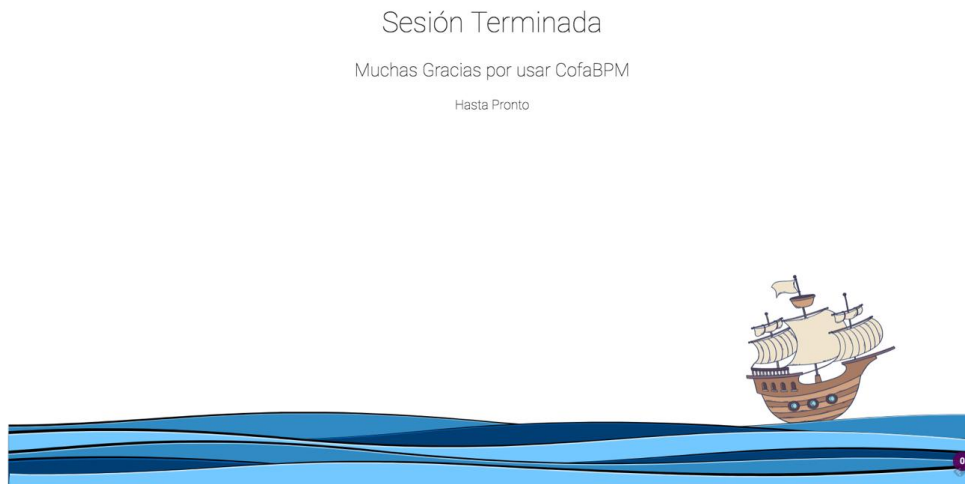


Figura 5-49 Finalización de la Sesión

Mensajes de Error e Informativos

Si alguna llamada de los servicios REST es utilizada, se muestra un pop-up informativo con el mensaje de error pertinente.

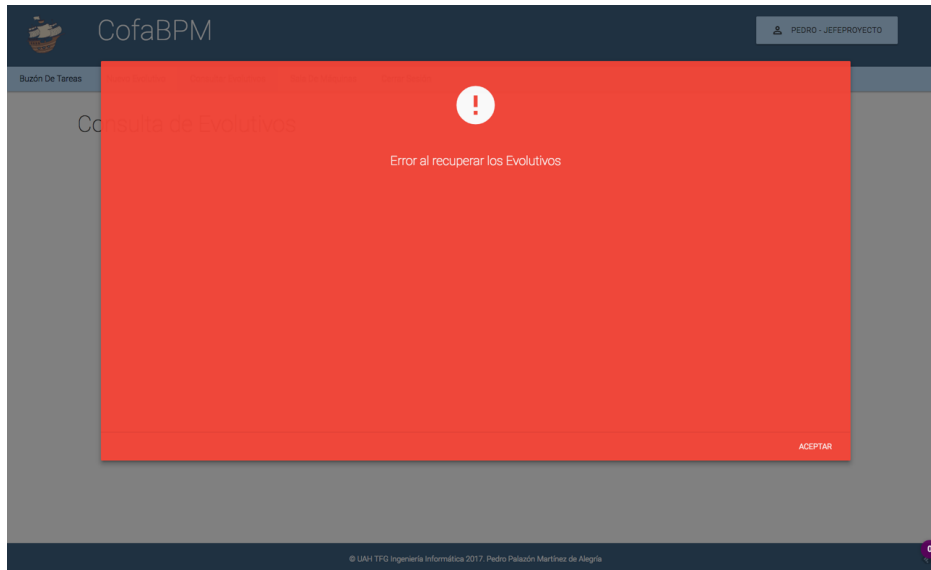


Figura 5-50 Mensaje de error

Del mismo modo cuando una acción de usuario tiene éxito, por ejemplo el abrir un evolutivo o completar una tarea asignada, aparece un pop-up de confirmación.

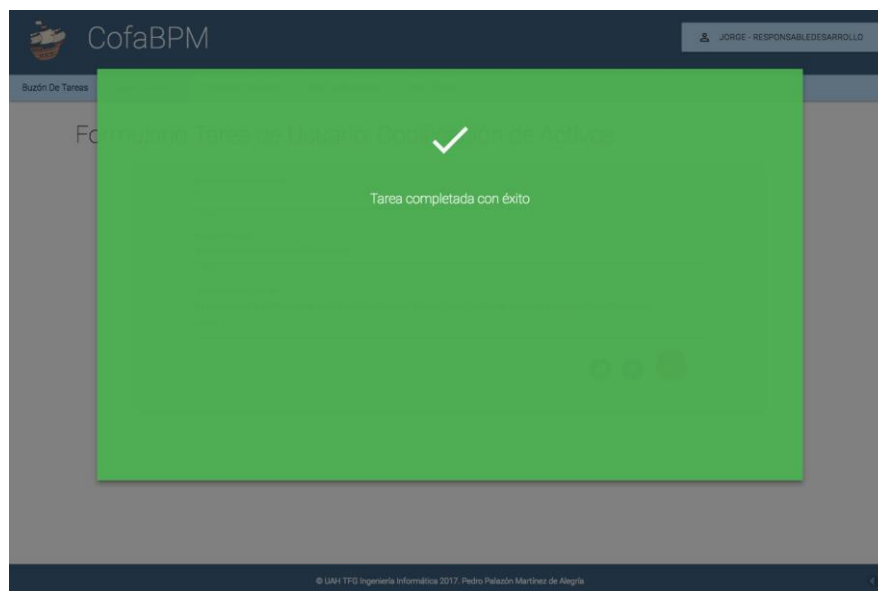


Figura 5-51 Mensaje de éxito

La pantalla relacionada con la Sala de Máquinas tiene una vista personalizada en caso de producirse un error en el motor.

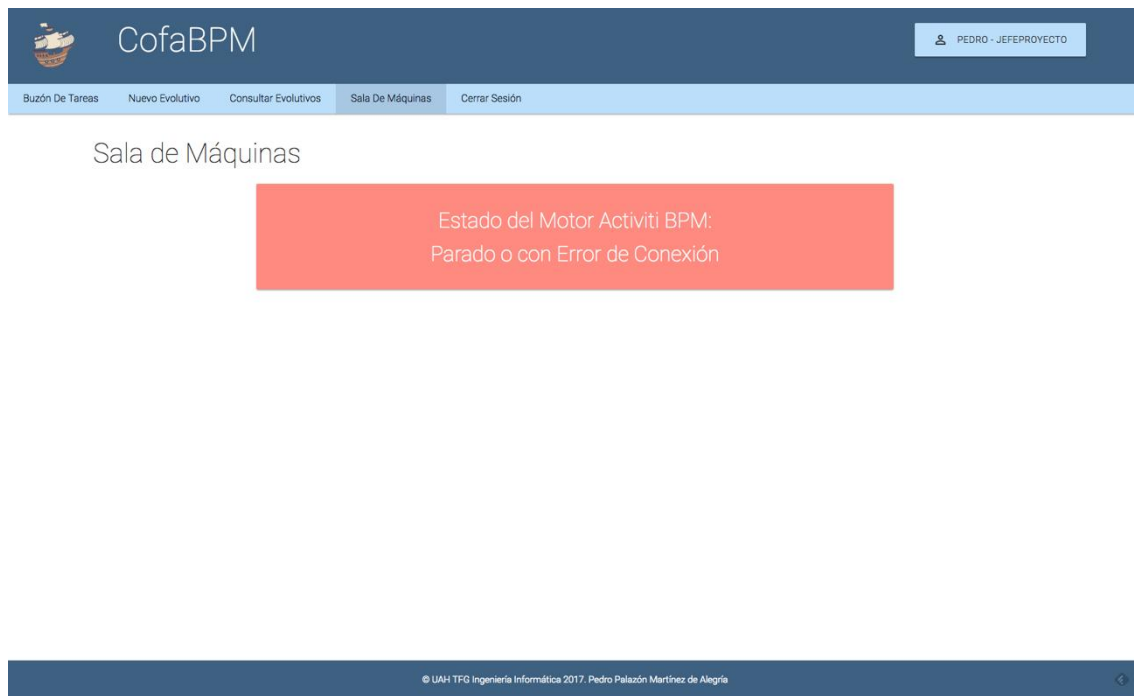


Figura 5-52 Mensaje de estado de conexión

Además, en caso de producirse un acceso no autorizado y se detecta se cierra la aplicación y se redirige a esta pantalla. Este control funciona si por algún motivo SpringSecurity no ha eliminado la cookie de sesión o se intenta un acceso por fuera. Hay un control más que es relacionado con la información de usuario guardada en el localStorage de Angular.

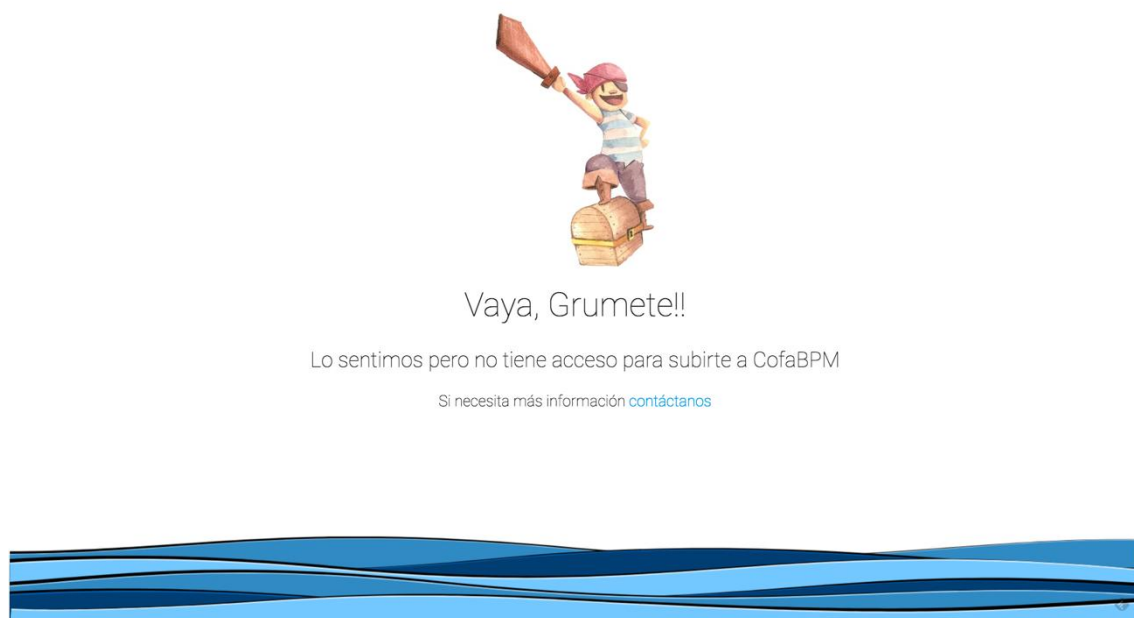


Figura 5-53 Mensaje de control de acceso

Envío de Emails del sistema

Durante la evolución del BPM pueden producirse no conformidades o incidencias que son informadas a través de una EmailTask lanzada por el propio BPM. Lo que le hace llegar a los usuarios un email con la información.

Este email es una plantilla de carácter responsive, para poder verse en el mayor número de dispositivos posibles.

Los emails constan de una cabecera y un pie, que son cargados como plantillas desde archivos de recurso. El contenido del mensaje es generado en tiempo de ejecución por el BPM y utilizando para ello la información introducida en los formularios.

En total son 6 los puntos del BPM donde una tárea de envío de Email puede ser lanzada:

1. Comienzo del evolutivo.
2. Errores durante la validación del diseño.
3. Errores durante las pruebas unitarias.
4. Errores durante las pruebas funcionales.
5. Informe de paralización del despliegue por las pruebas de integración.
6. Informe de paralización de despliegue por la no conformidad final.
7. Informe de evolutivo implantado con éxito.

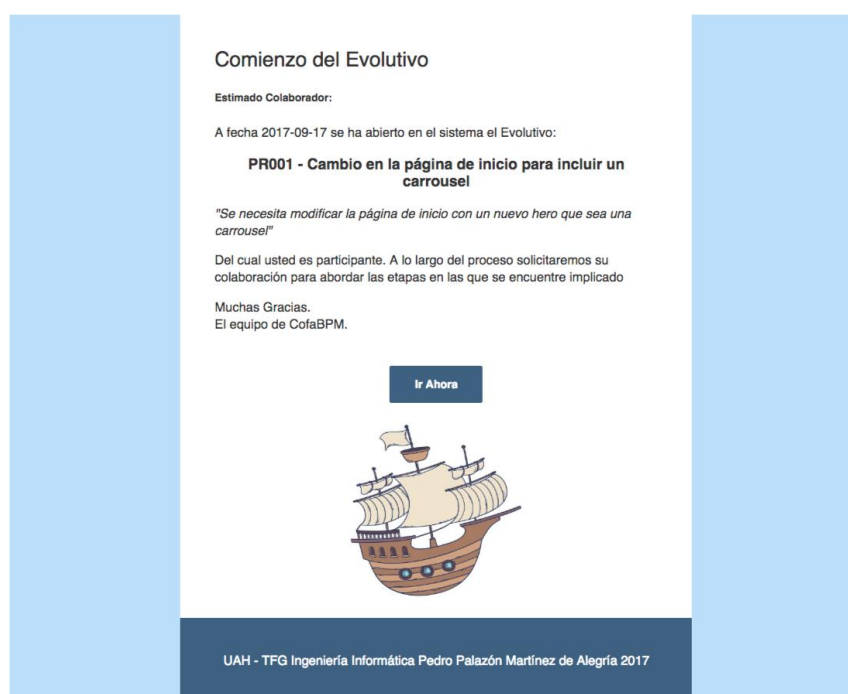


Figura 5-54 Plantilla de email enviada

5.5. Normativa

La normativa que aplica a este proceso de desarrollo evolutivo y que tiene que tenerse en cuenta en el marco de los requerimientos de seguridad y control, son las siguientes:

- Ley orgánica de Protección de datos de carácter personal
- Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico
- Ley 59/2003, de 19 de diciembre, de firma electrónica.
- Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia

En el caso de que el cliente para el que se desarrolla la versión de Sw evolutiva sea una AAPP:

- Ley 39/2015, de 1 de octubre, del Procedimiento Administrativo Común de las Administraciones Públicas.
- Ley 40/2015, de 1 de octubre, de Régimen Jurídico del Sector Público.
- Esquema Nacional de Seguridad
- Esquema Nacional de Interoperabilidad

1.- Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal

Se trata de la ley orgánica española que tiene por objeto establecer qué se considera dato de carácter personal, el nivel de criticidad del dato, así como las medidas a proporcionar en servicios tecnológicos y bases de datos para que no se viole el control de acceso y no exista fuga de datos.

Esta ley, además exige la existencia de un documento de seguridad dentro de la organización que regule las medidas, soportes, formatos y procedimientos de control de acceso tanto a la información física como lógica.

La misma ley exige la presentación clara y comprensible de los avisos legales y de protección de datos presentados al ciudadano, así como la proporcionalidad del dato, exigiendo así que se soliciten sólo los datos requeridos, pertinentes y no excesivos para el tratamiento o prestación del servicio. De igual forma, se exige cumplir con el principio de calidad, obligando a las entidades responsables del dato, a mantenerlo actualizado y a no utilizarlo con otra competencia o uso no autorizado por el interesado.

Por último, cabe destacar, que se exige la no transferencia de datos a un tercero, si esta no ha sido consentida expresamente por el propietario del dato. Sólo se consiente si es para la prestación de un servicio por parte de un tercero y existe un compromiso contractual que lo avala.

2.- Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico

Esta ley incorpora al ordenamiento legislativo la Directiva 2000/31/CE del Consejo y del Parlamento Europeo en la que se regulan determinados aspectos jurídicos de los Servicios de la Sociedad de la Información, en particular los relativos al comercio electrónico. Dada la extraordinaria expansión de las TIC y en especial los servicios en Internet, hace necesario establecer un marco jurídico adecuado que generen en todos los stakeholders la confianza, garantía y seguridad necesaria para el empleo de este nuevo medio.

En este sentido, esta ley establece tanto a los proveedores de servicios de intermediación, como a las empresas que ofrecen sus productos y servicios a través de una página web, las reglas necesarias para que el uso y disfrute de esta red, así como la posible actividad económica generada en torno a la compra sea una experiencia segura y confiable, y cuente con todas las garantías con las que cuenta una compra hecha de forma presencial y en un comercio tradicional.

3.- Ley 59/2003, de 19 de diciembre, de firma electrónica.

La presente ley regula la forma y los requisitos técnicos a cumplir tanto por parte de los prestadores de servicios de certificación, como las firmas electrónicas que se realicen sobre información, documentos ó datos, para que estos tengan plena validez legal y puedan ser considerados como jurídicamente válidos y sea totalmente equiparable un documento en soporte papel, firmado con rúbrica y de forma tradicional, a la firma realizada de forma electrónica sobre un documento electrónico.

Para que la firma electrónica sea válida tiene que contar con la base tecnológica exigida en esta ley, así como lo contenido en la ley 39 y 40 de 2015, en el caso de que se trate de documentos públicos administrativos.

4. Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.

La disposición final segunda de la Ley 27/1995, de 11 de octubre, de incorporación al Derecho español de la Directiva 93/98/CEE, del Consejo, de 29 de octubre, relativa a la armonización del plazo de protección del derecho de autor y de determinados derechos afines, autorizó al Gobierno para que, antes del 30 de junio de 1996, aprobara un texto que refundiese las disposiciones legales vigentes en materia de propiedad intelectual, regularizando, aclarando y armonizando los textos que hubieran de ser refundidos.

El alcance temporal de esta habilitación legislativa es el relativo a las disposiciones legales que se encontrarán vigentes a 30 de junio de 1996.

En consecuencia, se ha elaborado un texto refundido que se incorpora como anexo a este Real Decreto Legislativo, y que tiene por objeto dar cumplimiento al mandato legal.

En su virtud, a propuesta de la Ministra de Cultura, de acuerdo con el Consejo de Estado y previa deliberación del Consejo de Ministros en su reunión del día 12 de abril de 1996

[4.- Ley 39/2015, de 1 de octubre, del Procedimiento Administrativo Común de las Administraciones Públicas.](#)

[5.- Ley 40/2015, de 1 de octubre, de Régimen Jurídico del Sector Público.](#)

CAPÍTULO 6

RESUPUESTO Y MEDIOS EMPLEADOS



6. PRESUPUESTO Y MEDIOS EMPLEADOS

En este capítulo vamos a realizar una cuantificación de los medios empleados y el tiempo incurrido a la hora de crear CofaBPM.

6.1. Programas Utilizados

Todo el software utilizado para la creación de este proyecto es software gratuito y no ha sido necesario pagar ningún tipo de licencia. A excepción de las herramientas de ofimática y modelado.

- Sistema Operativo Ubuntu 16.04.3 LTS.
- Eclipse IDE JAVA EE Developers Release Neon.
- Java SE Development Kit 8.
- Atom Editor 64-bit.
- Postman Linux X64.
- Repositorio GIT albergado en BitBucket
- Microsoft Word 2017
- Sketch v46.

6.2. Tiempos Incurridos

El anteproyecto fue planificado para un ingeniero de software dedicado al 100% (8horas/día) con el siguiente diagrama de Gantt.

El coste total de días planificados eran 101 (808 horas).

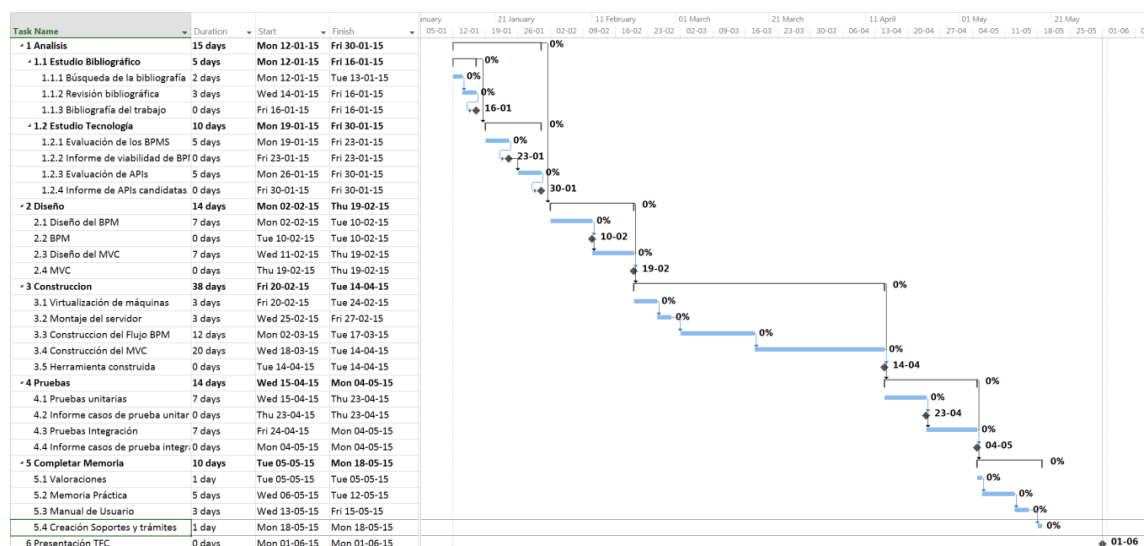


Figura 6-1 Diagrama de Gantt Inicial

El tiempo real ha sido ligeramente inferior y gran parte de este éxito viene provocado por la fase 1.2 de estudio tecnológico. La elección de las herramientas más idóneas para la construcción ha acelerado el proyecto.

El tiempo ganado ha sido en el punto 3, construcción.

- La tarea 3.1 Virtualización de la máquina ha quedado reducida de 3 días a 1. La creación de los droplet en Digital Ocean han facilitado la tarea.
- La tarea 3.2 Montaje del servidor que eran 3 días, ha quedado reducido a 1 día por el uso de SpringBoot.
- La tarea 3.4 Construcción del MVC también ha sido agilizada de 20 días a 12 por el uso de la tecnología Angular y componentes UI de Materialize.

En total ha habido un ahorro de 12 días, alrededor de 96 horas.

Por tanto, el proyecto en su totalidad ha sido realizado en **89 días (712 horas)**.

6.3. Costes

Además del tiempo incurrido, hay que tener en cuenta los costes de poner la herramienta en producción.

Estos costes vienen derivados de la compra del dominio, servicio de correo y cloud donde ejecutar la herramienta.

La compra del dominio y del correo son 4€ anuales.

El precio de la máquina es de 10\$/mes. Esto corresponde a unos 8,5€/mes que en tiempo de computación son unos 12 céntimos de € por hora.

Por tanto el coste total sería:

Detalle	Precio
Hosting y Correo	4€
Servicio Cloud (12cts./hora)*8760h	102€
TOTAL	106€

6.4. Beneficios Esperados

Poder ser capaces de implantar una herramienta de estas características en producción por unos 100€ anuales era impensable hace apenas 10 años. Lo que supone una muy buena noticia para la democratización de las tecnologías y la ganancia de competitividad en las pequeñas y medianas empresas.

Este tipo de herramientas basadas en procesos, hasta hace muy poco eran exclusivas de grandes corporaciones blindadas con millones de euros. Ahora una pequeña empresa, por menos de 9€ mensuales puede permitirse utilizar en su día a día una solución tan potente.

Como se dijo en capítulos anteriores, CofaBPM pretende ser una herramienta que elimine el gap de comunicaciones con el cliente. Creando una red valor que sea un factor diferenciador de sus competidores.

Teniendo en cuenta que una facturación media de un técnico cualificado puede rondar los 60€/h. Esta herramienta puede ser puesta en producción a cambio de menos de dos horas de este técnico cualificado.

Teniendo en cuenta que el número de horas que se gana mensualmente es muy superior: agilidad de las comunicaciones, eliminación de reuniones innecesarias, interminables hilos conversacionales de correo, errores de comunicación, etc. El retorno de inversión a corto y medio plazo queda más que probado.

CAPÍTULO 7

CONCLUSIONES Y FUTURAS LÍNEAS DE ACCIÓN



7. CONCLUSIONES Y FUTURAS LÍNEAS DE ACTUACIÓN

7.1. Conclusiones

El centro neurálgico de una empresa como es una factoría de sw, está formado por sus procesos y por las decisiones que se toman en los mismos. Una factoría tiene la característica de querer desarrollar sw como si de un proceso industrial se tratara, de forma, que es indudable que la automatización de los procesos y la formulación de workflows que se sustentan en desarrollos sw basados en bpm, permite contar con este modelo de gestión, si la necesidad de estar continuamente documentando las tareas.

A modo de resumen, podemos concluir que la primera conclusión es que la automatización del proceso con una herramienta BPM, potencia el concepto de industrialización del sw que es el objetivo principal de una Factoría Sw.

En este caso y gracias a la implantación de un workflow, se cuenta con una única fuente de información actualizada a la que acuden todos los procesos. Esta fuente de información por ser única, se encuentra sincronizada, actualizada, integrada y expurgada de forma automática, lo que garantiza una buena comunicación y una correcta toma de decisiones dentro de los procesos.

La segunda conclusión sería que los procesos se ven beneficiados por la utilización por parte de todos ellos de una única fuente de datos e información sincronizada.

Por otra parte, en la era industrial, de fabricación y venta en la que nos encontramos, en la que la tecnología tiene mucho que decir a favor de la hegemonía de las compañías y en la que los equipos de trabajo se encuentran deslocalizados, llegando incluso a incorporar en la cadena de valor a los propios clientes, otra conclusión a la que podemos llegar es que se mejora la colaboración del equipo y se favorece la compartición de ideas.

Por otra parte, es indudable que se mejora la calidad, el hecho de que el proceso sea repetible, se encuentre normalizado y que las tareas de trabajo respondan al orden, tiempo y estándares que la herramienta ha marcado, hace que la compañía se sitúe en niveles de calidad óptimos. La automatización del proceso y la comunicación asíncrona, hace que se produzcan menos tiempos muertos, y que no existan interrupciones por errores o cambios en el proceso.

Es cierto que otra de las conclusiones a la que podemos llegar es a que se obtiene un gran ahorro de costes, esto es debido a que se aumenta la eficiencia y la precisión en el proceso

Como conclusión también contamos con que aumenta la ventaja competitiva de la compañía. Ser más rápido, tener el proceso normalizado, ahorras costes y todas las conclusiones expuestas con anterioridad, hacen que se aumente la competitividad en el mercado, ya que se cumplen en mejor medida con las expectativas del mercado.

Por otra, parte y a modo de conclusión, podemos concluir que además con este ejemplo práctico se pone de manifiesto que con la tecnología elegida, se eliminan dos grandes desventajas de la automatización de procesos, como son, el coste de inversión y la dependencia tecnológica.

El coste de inversión se ve minimizado por el empleo de tecnologías de sw libre con costes poco representativos, y la dependencia tecnológica también se ve aplacada. Esto es así ya que el empleo de una tecnología implantada y muy estandarizada entre los principales motores de bpm hace que la tecnología no dependa del proveedor elegido ni verse cauto en contratos de mantenimiento ni encorsetados por desarrollos específicos.

Por otra parte, y ya para finalizar, otra de las conclusiones es la resistencia a la obsolescencia tecnológica. En cualquier tipo de herramienta existe el riesgo de obsolescencia, por lo que es importante que se tenga en cuenta antes del desarrollo y de abordar su presupuesto, tanto la amortización de la inversión y el Retorno de la Inversión, antes de nada.

En el proyecto que se ha presentado en este trabajo, puede verse que el retorno de la inversión es altísimo, es decir, la diferencia existente entre el beneficio obtenido en términos económicos con respecto a la inversión realizada, es enorme. Esta también es una conclusión obtenida tras la realización del presente trabajo.

7.2. Acciones Futuras

Atendiendo a su naturaleza se pueden categorizar las actuaciones de mejora de la siguiente manera:

A Nivel de Negocio

- Ampliar el catálogo de servicios de la factoría, por ejemplo, para abordar flujos de incidencias o proyectos basados en metodologías ágiles.
- Crear un cuadro de mando para las métricas de los proyectos.

A nivel técnico

- Integrar la firma electrónica de una CA como por ejemplo la FNMT o DNI-e. Para que los participantes firmen cada una de las tareas que completen. De ese modo podrían tener carácter de validez legal y no repudio en caso de existir contiendas legales relacionadas con el proyecto.
- Integración con un gestor documental para almacenar documentos (Alfresco sería el más idóneo). Se podría utilizar en la entrega de los diseños a implementar u otros documentos de análisis.
- Integración con una herramienta de control de versiones como por ejemplo GIT. Se utilizaría en las entregas de los artefactos codificados.
- Integración con herramientas de testing o calidad como SonarQube o Testlink. Se utilizaría en las fases de pruebas unitarias y pruebas funcionales.
- Integración con una herramienta de integración continua como Jenkins o Bamboo. Se utilizaría para realizar los despliegues de forma automática una vez aprobada su fecha de despliegue.

CAPÍTULO 8

BIBLIOGRAFÍA



8. BIBLIOGRAFÍA

8.1 Relación de Referencias Bibliográficas

- [1] Antonio Hidalgo Nuchera. La gestión de la tecnología como factor estratégico de la competitividad industrial. Revisata ECONOMÍA INDUSTRIAL N.º 330. Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid. 1999.
- [2] Sistemas de Gestión de la Calidad. Requisitos ISO 9001:2000, COPANT/ISO 9001:2000 NMX-CC-9001-IMNC-2000.
- [3] Nave, Dave. How To Compare Six Sigma, Lean and the Theory of Constraints. Quality Progress, Marzo 002, pp. 73-78.
- [4] Object Management Group, OMG Model Driven Architecture.
<http://www.omg.org/mda/>
- [5] Hansen, M., SOA using Java Web Services, Prentice Hall, 2007.
- [6] E. Porter. Competitive Strategy: Techniques for Analyzing Industries and Competitors. 1980
- [7] Desarrollada a finales de 1980, la Biblioteca de Infraestructura de Tecnologías de la Información (**ITIL**®) se ha convertido en el estándar mundial de de facto en la Gestión de Servicios Informáticos.
- [8] ITIL. The Official Introduction to the ITIL Service Lifecycle - by Office of Government Commerce (OGC) TSO © 2007.
- [9] TIL Service Design, 2011 Edition by Office of Government Commerce (OGC) -The Stationery Office Copyright Crown Copyright © 2011.
- [10] IBM Software. Informe sobre liderazgo experto. Once hábitos para el éxito de los programas de BPM. Pasos probados para comenzar con la mejora del proceso.WSW14117-ESE.
- [11] Juan José Camargo Vega, Jorge Enrique Orálora Luna y Aura Beatriz Alvarado Gaona. Revista Ingeniero N 9. Todo alrededor del BPM. Editorial, Desde su origen la tradición intelectual y la científica. ISSN Virtual 2322-8415
- [12] Sistemas de Gestión de la Calidad. Requisitos ISO 9001:2000, COPANT/ISO 9001:2000 NMX-CC-9001-IMNC-2000.
- [13] Artículo ISO 9001 2015 ¿Qué diferencia hay entre proceso y procedimiento?. Escuela Europea de Excelencia. 22 enero, 2016
<HTTP://WWW.NUEVA-ISO-9001-2015.COM/2016/01/ISO-9001-2015-DIFERENCIA-PROCESO-PROCEDIMIENTO/>
- [14] Artículo de IX jornada de Gerencia de Proyectos de TI. Fábricas del Sw. ACIS.

- [HTTP://52.0.140.184/TYPO43/FILEADMIN/BASE_DE_CONOCIMIENTO/IX_JORNADA_GERENCIA/CONFERENCIAO_RLANDOALBERTOACOSTAVARGAS.PDF](http://52.0.140.184/TYPO43/FILEADMIN/BASE_DE_CONOCIMIENTO/IX_JORNADA_GERENCIA/CONFERENCIAO_RLANDOALBERTOACOSTAVARGAS.PDF)
- [15] [Garzás, J., & Piattini, M. \(2007\). Factorías de software: Experiencias, tecnologías y organización: Ra-ma.](#)
- [16] Thomas M. Pigosky, (1997). Practical Software Maintenance. Best practices for managing your Software Investment
- [17] The Object Management Group® (OMG®) is an international, open membership, not-for-profit technology standards consortium <http://www.omg.org/>
- [18] Garzás, J., & Piattini, M. (2010). Fábricas de software: experiencias, tecnologías y organización – 2ª edición
- [19] Artículo Mayo 07 sobre [Garzás, J., & Piattini, M. \(2007\). Factorías de software: Experiencias, tecnologías y organización: Ra-ma.](#)
<http://www.kybeleconsulting.com/articulos/concepto-y-evolucion-de-las-fabricas-software/>
- [20] Aoyama, M. (1996). Beyond software factories: concurrent-development process and an evolution of software process technology in Japan. Information and Software Technology, 38(3), 133-143.
- [21] Basili, V. R., Caldiera, G., & Cantone, G. (1992). A Reference Architecture for the Component Factory. ACM Transactions on Software Engineering and Methodology, 1(1), 53-80.
- [22] Greenfield, J., Short, K., Cook, S., Kent, S., & Crupi, J. (2004). Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools (1st ed.): Wiley.
- [23] Li, C., Li, H., & Li, M. (2001). A Software Factory Model Based on ISO 9000 and CMM for Chinese Small Organizations. Paper presented at the Second Asia-Pacific Conference on Quality Software (APAQS'01).
- [24] Object Management Group. <http://www.omg.org/spec/BPMN/2.0.2/PDF/>
- [25] Michael E. Porter. Competitive Strategy: Techniques for Analyzing Industries and Competitors. 1980.
- [26] Michael E. Porter. Estrategia Competitiva: Técnicas Para el Análisis de la Empresa y sus Competidores Pirámide, 2009.
- [27] Michael E. Porter. Ventaja Competitiva: Creacion y sostenimiento de un desarrollo superior. Piramide, 2010.
- [28] Nave, Dave. How To Compare Six Sigma, Lean and the Theory of Constraints. Quality Progress, pp. 73-78. Marzo 2002.
- [29] Object Management Group, OMG Model Driven Architecture.
<http://www.omg.org/mda/>

- [30] United States General Accounting Office. GAO: "Business Process Reengineering Assessment Guide". Version 3. May 1997
- [31] Rademakers.Tijs (2012) Activiti in Action

ANEXO A

CÓDIGO FUENTE



9. ANEXO A – Código Fuente

9.1 BPM generado

BPM_Evolutivo.bpmn20.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:activiti="http://activiti.org/bpmn" xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:omgdc="http://www.omg.org/spec/DD/20100524/DC" xmlns:omgdi="http://www.omg.org/spec/DD/20100524/DI"
typeLanguage="http://www.w3.org/2001/XMLSchema" expressionLanguage="http://www.w3.org/1999/XPath" targetNamespace="www.cofabpm.com">
<process id="BPM_Evolutivo" name="BPM_Evolutivo" isExecutable="true">
<documentation>Proceso para la puesta en producción de un evolutivo software desde la Factoría de Software a un cliente</documentation>
<startEvent id="startEvent1" name="Inicio Evolutivo"></startEvent>
<userTask id="UT1_Alta_Proyecto" name="Alta Proyecto" activiti:candidateGroups="JefeProyecto">
<documentation>Alta en el sistema el nuevo evolutivo</documentation>
<extensionElements>
<activiti:formProperty id="codigo" name="codigo" type="string"></activiti:formProperty>
<activiti:formProperty id="nombre" name="nombre" type="string"></activiti:formProperty>
<activiti:formProperty id="descripcion" name="descripcion" type="string"></activiti:formProperty>
<activiti:formProperty id="fechaInicio" name="fechaInicio" type="date" expression="yyyy-MM-dd"></activiti:formProperty>
<activiti:formProperty id="fechafin" name="fechafin" type="date" datePattern="yyyy-MM-dd"></activiti:formProperty>
</extensionElements>
</userTask>
<userTask id="UT2_Entrega_DF" name="Entrega Diseño Físico" activiti:candidateGroups="ProductOwner">
<documentation>Entrega de los diseños de los activos que se han de desarrollar</documentation>
<extensionElements>
<activiti:formProperty id="fechaEntregaDF" name="fechaEntregaDF" type="date" datePattern="yyyy-MM-DD"></activiti:formProperty>
<activiti:formProperty id="descripcion" name="descripcion" type="string"></activiti:formProperty>
<activiti:formProperty id="rutaEntrega" name="rutaEntrega" type="string"></activiti:formProperty>
<activiti:formProperty id="entregables" name="entregables" type="long"></activiti:formProperty>
</extensionElements>
</userTask>
<sequenceFlow id="sid-56BADC35-CD63-4926-B158-913FFF5D0E4D" sourceRef="UT1_Alta_Proyecto"
targetRef="UT2_Entrega_DF"></sequenceFlow>
<userTask id="UT3_Validacion_DF" name="Validación Diseño" activiti:candidateGroups="ResponsableAnalista">
<documentation>Se validan los diseños entregados</documentation>
<extensionElements>
<activiti:formProperty id="conforme" name="conforme" type="boolean" required="true"></activiti:formProperty>
<activiti:formProperty id="motivo" name="motivo" type="string"></activiti:formProperty>
</extensionElements>
</userTask>
<sequenceFlow id="sid-DA569EE0-BB34-4A88-B91B-ABF9DB6C0129" sourceRef="UT2_Entrega_DF"
targetRef="UT3_Validacion_DF"></sequenceFlow>
<exclusiveGateway id="GAT1_ConformidadDF" name="¿Es correcto el diseño?"></exclusiveGateway>
<sequenceFlow id="sid-D2A57FA3-C425-4A41-A382-E76B3E287998" sourceRef="UT3_Validacion_DF"
targetRef="GAT1_ConformidadDF"></sequenceFlow>
<userTask id="UT4_CodificacionActivos" name="Codificación de Activos" activiti:candidateGroups="ResponsableDesarrollo">
<documentation>Codificación de los activos</documentation>
<extensionElements>
<activiti:formProperty id="rutaEntrega" name="rutaEntrega" type="string"></activiti:formProperty>
<activiti:formProperty id="activosCodificados" name="activosCodificados" type="long"></activiti:formProperty>
<activiti:formProperty id="observaciones" name="observaciones" type="string"></activiti:formProperty>
</extensionElements>
</userTask>
<exclusiveGateway id="GAT2_ConformidadCO" name="¿Los activos esta completados?"></exclusiveGateway>
```

```

<userTask id="UT6_Puebas_Unitarias_Activos" name="Pruebas Unitarias de Activos" activiti:candidateGroups="ResponsableTesting">
  <documentation>Pruebas Unitarias de todos los activos del evolutivo</documentation>
  <extensionElements>
    <activiti:formProperty id="testRealizados" name="testRealizados" type="long"></activiti:formProperty>
    <activiti:formProperty id="testErroneos" name="testErroneos" type="long"></activiti:formProperty>
    <activiti:formProperty id="coberturaCubierta" name="coberturaCubierta" type="long"></activiti:formProperty>
    <activiti:formProperty id="rutaInformes" name="rutaInformes" type="string"></activiti:formProperty>
  </extensionElements>
</userTask>
<exclusiveGateway id="GAT3_PuebasUnitarias" name="¿Hay algún error de codificación?"></exclusiveGateway>
<sequenceFlow id="sid-A36D8BBC-76F8-4937-ADD9-5F5C6950D4FE" sourceRef="UT6_Puebas_Unitarias_Activos"
targetRef="GAT3_PuebasUnitarias"></sequenceFlow>
<serviceTask id="ET3_Errores_en_Puebas_Unitarias" name="Errores en Pruebas Unitarias" activiti:type="mail">
  <documentation>Aviso si hay algun error durante las pruebas unitarias</documentation>
  <extensionElements>
    <activiti:field name="to">
      <activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="from">
      <activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="subject">
      <activiti:expression><![CDATA[Evolutivo ${evolutivo.codigo}-${evolutivo.nombre} detectados errores durante las Pruebas Unitarias
realizadas]]></activiti:expression>
    </activiti:field>
    <activiti:field name="html">
      <activiti:expression><![CDATA[${templateHeader}
      <tr>
        <td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">
          <h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Errores
Pruebas Unitarias</h1>
          <h5>Estimado Colaborador:</h5>
          <p style="margin: 0;">Se han detectado errores durante las pruebas unitarias. Siendo el número de pruebas ejecutadas ${testRealizados} y
el número de pruebas fallidas ${testErroneos}. Con una cobertura de código del ${coberturaCubierta}%. Por favor revise los test incorrectos.</p>
          <p></p>
          <p style="margin: 0;">Muchas Gracias.</p>
          <p style="margin: 0;">El equipo de CofaBPM.</p>
        </td>
      </tr>
      ${templateFooter}></activiti:expression>
    </activiti:field>
  </extensionElements>
</serviceTask>
<serviceTask id="ET2_ErroresValidacionDF" name="Errores Validación Diseño" activiti:type="mail">
  <documentation>Aviso de Errores en la Validación del Diseño Físico</documentation>
  <extensionElements>
    <activiti:field name="to">
      <activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="from">
      <activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="subject">
      <activiti:expression><![CDATA[Evolutivo ${evolutivo.codigo}-${evolutivo.nombre} DISCONFORMIDAD con el Diseño Físico
entregado]]></activiti:expression>
    </activiti:field>
    <activiti:field name="html">
      <activiti:expression><![CDATA[${templateHeader}
      <tr>
        <td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">

```

```

        <h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Errores
Entrega Diseño Físico</h1>
        <h5>Estimado Colaborador:</h5>
        <p style="margin: 0;">Se han detectado errores en la validación del diseño físico entregado, por lo que se ha procedido a rechazar la
entrega. Por favor revise el motivo del rechazo y vuelva a enviar el DF corregido.</p>
        <h5>Motivo:</h5>
        <p><i>"${motivo}"</i></p>
        <p></p>
        <p style="margin: 0;">Muchas Gracias.</p>
        <p style="margin: 0;">El equipo de CofaBPM.</p>
    </td>
</tr>
    ${templateFooter}}]></activiti:expression>
</activiti:field>
</extensionElements>
</serviceTask>
<sequenceFlow id="sid-7DC9D0F3-E4FC-4A47-BE3E-50E49DC56F31" sourceRef="ET2_ErroresValidacionDF"
targetRef="UT2_Entrega_DF"></sequenceFlow>
<userTask id="UT7_Validacion_Pruebas_Unitarias" name="Validación Pruebas Unitarias" activiti:candidateGroups="JefeProyecto">
    <documentation>Se da conformidad a las pruebas unitarias tras ser revisadas</documentation>
    <extensionElements>
        <activiti:formProperty id="conforme" name="conforme" type="boolean"></activiti:formProperty>
        <activiti:formProperty id="motivo" name="motivo" type="string"></activiti:formProperty>
    </extensionElements>
</userTask>
<sequenceFlow id="sid-4C03116A-4E58-4C14-B349-57AF6DDE2982" sourceRef="UT5_Validacion_Activos"
targetRef="GAT2_ConformidadCO"></sequenceFlow>
<userTask id="UT5_Validacion_Activos" name="Validación Activos" activiti:candidateGroups="JefeProyecto">
    <documentation>Se da conformidad a los activos codificados</documentation>
    <extensionElements>
        <activiti:formProperty id="conforme" name="conforme" type="boolean"></activiti:formProperty>
        <activiti:formProperty id="motivo" name="motivo" type="string"></activiti:formProperty>
    </extensionElements>
</userTask>
<exclusiveGateway id="GAT4_ConformidadPU"></exclusiveGateway>
<exclusiveGateway id="GAT5_PruebasFuncionales"></exclusiveGateway>
<userTask id="UT9_Validacion_Pruebas_Funcionales" name="Validación Pruebas Funcionales" activiti:candidateGroups="JefeProyecto">
    <documentation>Se da conformidad a las pruebas funcionales realizadas</documentation>
    <extensionElements>
        <activiti:formProperty id="conforme" name="conforme" type="boolean"></activiti:formProperty>
        <activiti:formProperty id="motivo" name="motivo" type="string"></activiti:formProperty>
    </extensionElements>
</userTask>
<exclusiveGateway id="GAT6_ConformidadPF"></exclusiveGateway>
<userTask id="UT10_Pruebas_Integracion" name="Pruebas de Integración" activiti:candidateGroups="ProductOwner">
    <documentation>Se da conformidad a las pruebas de integración realizadas</documentation>
    <extensionElements>
        <activiti:formProperty id="conforme" name="conforme" type="boolean"></activiti:formProperty>
        <activiti:formProperty id="motivo" name="motivo" type="string"></activiti:formProperty>
    </extensionElements>
</userTask>
<serviceTask id="ET4_Errores_Pruebas_Funcionales" name="Informe Validación Pruebas Funcionales" activiti:type="mail">
    <documentation>Aviso de errores durante las pruebas funcionales</documentation>
    <extensionElements>
        <activiti:field name="to">
            <activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
        </activiti:field>
        <activiti:field name="from">
            <activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
        </activiti:field>
    </extensionElements>
</serviceTask>

```

```

<activiti:field name="subject">
  <activiti:expression><![CDATA[Evolutivo ${evolutivo.codigo}-${evolutivo.nombre} detectados errores durante las Pruebas Funcionales
realizadas]]></activiti:expression>
</activiti:field>
<activiti:field name="html">
  <activiti:expression><![CDATA[{$templateHeader}
  <tr>
    <td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">
      <h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Errores
Pruebas Funcionales</h1>
      <h5>Estimado Colaborador:</h5>
      <p style="margin: 0;">Se han detectado errores durante las pruebas funcionales. Siendo los casos de uso probados ${casosUso} y el
número de casos fallidas ${casosErroneos}. Por favor revise los casos de uso incorrectos.</p>
      <p style="margin: 0;">Muchas Gracias.</p>
      <p style="margin: 0;">El equipo de CofaBPM.</p>
    </td>
  </tr>
  {$templateFooter}></activiti:expression>
</activiti:field>
</extensionElements>
</serviceTask>
<sequenceFlow id="sid-AB238A46-69BE-466C-90DA-D96A8F573322" sourceRef="ET4_Errores_Pruebas_Funcionales"
targetRef="UT3_Validacion_DF"></sequenceFlow>
<exclusiveGateway id="GAT7_ConformidadPI"></exclusiveGateway>
<userTask id="UT11_Aprobacion_Despliegue" name="Aprobación y Planificación Despliegue" activiti:candidateGroups="ProductOwner">
  <documentation>Se da conformidad a la implantación</documentation>
  <extensionElements>
    <activiti:formProperty id="aprobadaImplantacion" name="aprobadaImplantacion" type="boolean"></activiti:formProperty>
    <activiti:formProperty id="motivo" name="motivo" type="string"></activiti:formProperty>
    <activiti:formProperty id="fechalPlantacion" name="fechalPlantacion" type="date" datePattern="yyyy-MM-dd"></activiti:formProperty>
  </extensionElements>
</userTask>
<serviceTask id="ET5_2_Paralizacion_Despliegue" name="Informe Paralización Despliegue" activiti:type="mail">
  <documentation>Aviso de paralización del despliegue</documentation>
  <extensionElements>
    <activiti:field name="to">
      <activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="from">
      <activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="subject">
      <activiti:expression><![CDATA[AVISO IMPORTANTE: Evolutivo ${evolutivo.codigo}-${evolutivo.nombre} PARALIZACIÓN del
despliegue]]></activiti:expression>
    </activiti:field>
    <activiti:field name="html">
      <activiti:expression><![CDATA[{$templateHeader}
      <tr>
        <td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">
          <h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Paralización
del Evolutivo</h1>
          <h5>Estimado Colaborador:</h5>
          <p style="margin: 0;">Se ha procedido a paralizar la puesta en producción del evolutivo por el siguiente motivo:</p>
          <p><i>"${motivo}"</i></p>
          <p></p>
          <p></p>
          <p style="margin: 0;">Muchas Gracias.</p>
          <p style="margin: 0;">El equipo de CofaBPM.</p>
        </td>
      </tr>
    </activiti:expression>
  </activiti:field>
  </extensionElements>
</serviceTask>

```



```

    ${templateFooter}}></activiti:expression>
  </activiti:field>
</extensionElements>
</serviceTask>
<serviceTask id="ET1_Comienzo_Evolutivo" name="Comienzo Evolutivo" activiti:type="mail">
  <documentation>Aviso del comienzo del evolutivos</documentation>
  <extensionElements>
    <activiti:field name="to">
      <activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="from">
      <activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="subject">
      <activiti:expression><![CDATA[Comienzo del Evolutivo ${evolutivo.codigo}-${evolutivo.nombre}]]></activiti:expression>
    </activiti:field>
    <activiti:field name="html">
      <activiti:expression><![CDATA[${templateHeader}
        <tr>
          <td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">
            <h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Comienzo
del Evolutivo</h1>
            <h5>Estimado Colaborador:</h5>
            <p style="margin: 0;">A fecha ${evolutivo.getFechaInicio()} se ha abierto en el sistema el Evolutivo:</p>
            <h3 style="text-align: center;">${evolutivo.codigo} - ${evolutivo.nombre}</h3>
            <p><i>${evolutivo.descripcion}</i></p>
            <p style="margin: 0;">Del cual usted es participante. A lo largo del proceso solicitaremos su colaboración para abordar las etapas en las
que se encuentre implicado</p>
            <p></p>
            <p style="margin: 0;">Muchas Gracias.</p>
            <p style="margin: 0;">El equipo de CofaBPM.</p>
          </td>
        </tr>
      </activiti:expression>
    </activiti:field>
  </extensionElements>
</serviceTask>
<sequenceFlow id="sid-7D35818F-B66D-40B4-96C9-11CDB1FDB07C" sourceRef="UT1_Alta_Proyecto"
targetRef="ET1_Comienzo_Evolutivo"></sequenceFlow>
<userTask id="UT8_Pruebas_Funcionales" name="Pruebas Funcionales" activiti:candidateGroups="ResponsableAnalista">
  <extensionElements>
    <activiti:formProperty id="casosUso" name="casosUso" type="string"></activiti:formProperty>
    <activiti:formProperty id="casosErroneos" name="casosEroneos" type="string"></activiti:formProperty>
    <activiti:formProperty id="rutaEvidencias" name="rutaEvidencias" type="string"></activiti:formProperty>
    <activiti:formProperty id="observaciones" name="observaciones" type="string"></activiti:formProperty>
  </extensionElements>
</userTask>
<sequenceFlow id="sid-57F347EA-41BA-4ED6-AFC4-1AD66E6E2CEA" name="¿conformidad?" sourceRef="UT9_Validacion_Pruebas_Funcionales"
targetRef="GAT6_ConformidadPF"></sequenceFlow>
<endEvent id="sid-741D59CF-5A16-4C1F-A685-FC3543C5C732" name="Evolutivo Implantado"></endEvent>
<endEvent id="sid-A7F526EF-0199-4720-B92A-C6037744A4C9" name="Evolutivo Paralizado"></endEvent>
<sequenceFlow id="sid-0786241F-6418-4913-AD6F-9FC671A36E88" sourceRef="ET5_2_Paralizacion_Despliegue" targetRef="sid-A7F526EF-0199-
4720-B92A-C6037744A4C9"></sequenceFlow>
<exclusiveGateway id="GAT8_ConformidadDespliegue"></exclusiveGateway>
<sequenceFlow id="sid-6FDA76D5-DE75-493A-A2D2-99ED64BA3742" sourceRef="UT11_Aprobacion_Despliegue"
targetRef="GAT8_ConformidadDespliegue"></sequenceFlow>
<sequenceFlow id="sid-421293CB-E404-4FA5-89E1-D5910A8F2294" name="¿Son Correctas?" sourceRef="UT10_Pruebas_Integracion"
targetRef="GAT7_ConformidadPI"></sequenceFlow>
<sequenceFlow id="sid-66BFE900-40FE-4704-B7DE-9178C60DF2B2" name="No" sourceRef="GAT1_ConformidadDF"
targetRef="ET2_ErroresValidacionDF">

```

```

<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(conforme == false)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-ED9B3365-1BA1-47DD-8959-7A1A8159B7B9" sourceRef="ET3_Errores_en_Pruebas_Unitarias"
targetRef="UT4_CodificacionActivos"></sequenceFlow>
<sequenceFlow id="sid-8A04E411-48CA-4049-B041-F035696937CD" name="No" sourceRef="GAT4_ConformidadPU"
targetRef="UT6_Pruebas_Unitarias_Activos">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(conforme == false)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-5B03B468-404A-4459-A49A-909A630C8183" name="¿conformidad?" sourceRef="UT7_Validacion_Pruebas_Unitarias"
targetRef="GAT4_ConformidadPU"></sequenceFlow>
<sequenceFlow id="sid-7C53637E-E390-4210-8CBC-54C9779BC6CB" name="¿Hay algún error Funcional?" sourceRef="UT8_Pruebas_Funcionales"
targetRef="GAT5_PruebasFuncionales"></sequenceFlow>
<sequenceFlow id="sid-4F5D20EA-9760-4F67-8DD3-F7E4B26A815E" name="Si" sourceRef="GAT4_ConformidadPU"
targetRef="UT8_Pruebas_Funcionales">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(conforme == true)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-21EA5ED8-2CB6-4CE7-AB07-575BEB81A4EF" name="casosErroneos &gt; 0" sourceRef="GAT5_PruebasFuncionales"
targetRef="ET4_Errores_Pruebas_Funcionales">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(casosErroneos > 0)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-A76D6080-F7B5-4590-AED2-6FE4AE050902" name="si" sourceRef="GAT6_ConformidadPF"
targetRef="UT10_Pruebas_Integracion">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(conforme == true)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-D5E35CBC-AABF-4512-AA5E-56302FCFE3E2" name="No" sourceRef="GAT6_ConformidadPF"
targetRef="UT8_Pruebas_Funcionales">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(conforme == false)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-2B1F1758-D64B-4B1B-9555-88F9FDD30643" name="testErroneos &gt; 0" sourceRef="GAT3_PruebasUnitarias"
targetRef="ET3_Errores_en_Pruebas_Unitarias">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(testErroneos > 0)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-590354FF-8BBF-4AFE-BFA3-FA76671DAB30" name="testErroneos = 0" sourceRef="GAT3_PruebasUnitarias"
targetRef="UT7_Validacion_Pruebas_Unitarias">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(testErroneos == 0)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-5DA8EDD9-84B9-49CA-8FD5-03FE795CD784" name="casosErroneos = 0" sourceRef="GAT5_PruebasFuncionales"
targetRef="UT9_Validacion_Pruebas_Funcionales">
<conditionExpression xsi:type="tFormalExpression"><![CDATA[{$(casosErroneos == 0)}]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-1400CD61-B0A6-40BD-9FF6-B1E297323637" sourceRef="startEvent1" targetRef="UT1_Alta_Proyecto"></sequenceFlow>
<serviceTask id="ET5_1_InformeFinal" name="Informe Final Despliegue" activiti:type="mail">
<documentation>Aviso de paralización del despliegue</documentation>
<extensionElements>
<activiti:field name="to">
<activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
</activiti:field>
<activiti:field name="from">
<activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
</activiti:field>
<activiti:field name="subject">
<activiti:expression><![CDATA[Evolutivo ${evolutivo.codigo}-${evolutivo.nombre} implantado satisfactoriamente]]></activiti:expression>
</activiti:field>
<activiti:field name="html">
<activiti:expression><![CDATA[{$(templateHeader)}
<tr>
<td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">
<h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Implantación
del Evolutivo</h1>
<h5>Estimado Colaborador:</h5>

```

```

<p style="margin: 0;">A fecha ${fechalmlantacion} se ha implantado satisfactoriamente el Evolutivo:</p>
<h3 style="text-align: center;">${evolutivo.nombre}</h3>
<h5>Observaciones:</h5>
<p><i>${descripcion}</i></p>
<p style="margin: 0;"> Muchas gracias por su inestimable colaboración durante el proceso.</p>
<p> </p>
<p style="margin: 0;">Un saludo</p>
<p style="margin: 0;">El equipo de CofaBPM.</p>
</td>
</tr>
${templateFooter}}</activiti:expression>
</activiti:field>
</extensionElements>
</serviceTask>
<sequenceFlow id="sid-11C24E3F-B493-4C9C-A4A9-BAAE51F5F8AE" sourceRef="ET5_1_InformeFinal" targetRef="sid-741D59CF-5A16-4C1F-A685-FC3543C5C732"></sequenceFlow>
<sequenceFlow id="sid-ABE2D226-8471-4FA1-8509-7CA9B7392AE4" name="Si" sourceRef="GAT1_ConformidadDF" targetRef="UT4_CodificacionActivos">
  <conditionExpression xsi:type="tFormalExpression"><![CDATA[${conforme == true}]]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-956A4D78-48E9-4F3B-828C-039486D22FAA" sourceRef="UT4_CodificacionActivos" targetRef="UT5_Validacion_Activos"></sequenceFlow>
<sequenceFlow id="sid-AAAF82F0-4464-4AB5-9E4D-18606C24B335" name="No" sourceRef="GAT2_ConformidadCO" targetRef="UT4_CodificacionActivos">
  <conditionExpression xsi:type="tFormalExpression"><![CDATA[${conforme == false}]]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-A846C32B-2FDF-4470-BE94-928BB670A600" sourceRef="GAT2_ConformidadCO" targetRef="UT6_Pruebas_Unitarias_Activos">
  <conditionExpression xsi:type="tFormalExpression"><![CDATA[${conforme == true}]]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-4413BB52-DD96-426A-9B56-F8F9909BB0ED" name="Conformidad Despliegue" sourceRef="GAT8_ConformidadDespliegue" targetRef="ET5_1_InformeFinal">
  <conditionExpression xsi:type="tFormalExpression"><![CDATA[${aprobadaImplantacion == true}]]></conditionExpression>
</sequenceFlow>
<sequenceFlow id="sid-DF776A4E-461A-4D8B-9C9C-80928A4BDB83" name="Disconformidad Despliegue" sourceRef="GAT8_ConformidadDespliegue" targetRef="ET5_2_Paralizacion_Despliegue">
  <conditionExpression xsi:type="tFormalExpression"><![CDATA[${aprobadaImplantacion == false}]]></conditionExpression>
</sequenceFlow>
<serviceTask id="ET5_3_Paralizacion_Despliegue" name="Informe Paralización Despliegue" activiti:type="mail">
  <documentation>Aviso de paralización del despliegue</documentation>
  <extensionElements>
    <activiti:field name="to">
      <activiti:string><![CDATA[palazon@gmail.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="from">
      <activiti:string><![CDATA[info@cofabpm.com]]></activiti:string>
    </activiti:field>
    <activiti:field name="subject">
      <activiti:expression><![CDATA[AVISO IMPORTANTE: Evolutivo ${evolutivo.nombre} PARALIZACIÓN del despliegue por errores en las Pruebas de Integración]]></activiti:expression>
    </activiti:field>
    <activiti:field name="html">
      <activiti:expression><![CDATA[${templateHeader}
<tr>
  <td style="padding: 40px; font-family: sans-serif; font-size: 15px; line-height: 20px; color: #333333;">
    <h1 style="margin: 0 0 10px 0; font-family: sans-serif; font-size: 24px; line-height: 27px; color: #333333; font-weight: normal;">Paralización del Evolutivo</h1>
    <h5>Estimado Colaborador:</h5>
    <p style="margin: 0;">Se ha procedido a paralizar la puesta en producción del evolutivo. Los problemas encontrados durante las pruebas de integración son:</p>

```

```

        <p><i>"${motivo}"</i></p>
        <p></p>
        <p></p>
        <p style="margin: 0;">Muchas Gracias.</p>
        <p style="margin: 0;">El equipo de CofaBPM.</p>
    </td>
</tr>
    ${templateFooter}}></activiti:expression>
</activiti:field>
</extensionElements>
</serviceTask>
<sequenceFlow id="sid-56D6D3A4-58EE-454B-86B9-96048FCDA9E" name="Si" sourceRef="GAT7_ConformidadPI"
targetRef="UT11_Aprobacion_Despliegue">
    <conditionExpression xsi:type="tFormalExpression"><![CDATA[${conforme == true}]]></conditionExpression>
</sequenceFlow>
<endEvent id="sid-ED53C347-F389-4B77-A9E9-CDF21FDEAB5C" name="Evolutivo Paralizado"></endEvent>
<sequenceFlow id="sid-FED4C7E9-FC30-47B5-B6C9-4B8C0CB67AFE" sourceRef="ET5_3_Paralizacion_Despliegue" targetRef="sid-ED53C347-F389-
4B77-A9E9-CDF21FDEAB5C"></sequenceFlow>
<sequenceFlow id="sid-49EA735D-884F-4702-A929-141B24CA6425" sourceRef="GAT7_ConformidadPI" targetRef="ET5_3_Paralizacion_Despliegue">
    <conditionExpression xsi:type="tFormalExpression"><![CDATA[${conforme == true}]]></conditionExpression>
</sequenceFlow>
</process>
<bpmndi:BPMNDiagram id="BPMNDiagram_BPM_Evolutivo">
<bpmndi:BPMNPlane bpmnElement="BPM_Evolutivo" id="BPMNPlane_BPM_Evolutivo">
    <bpmndi:BPMNShape bpmnElement="startEvent1" id="BPMNShape_startEvent1">
        <omgdc:Bounds height="30.0" width="30.0" x="37.037037624257394" y="400.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="UT1_Alta_Proyecto" id="BPMNShape_UT1_Alta_Proyecto">
        <omgdc:Bounds height="80.0" width="99.9999999999997" x="120.0" y="375.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="UT2_Entrega_DF" id="BPMNShape_UT2_Entrega_DF">
        <omgdc:Bounds height="80.0" width="100.0" x="321.15542108357954" y="375.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="UT3_Validacion_DF" id="BPMNShape_UT3_Validacion_DF">
        <omgdc:Bounds height="80.0" width="100.0" x="503.23681669134817" y="375.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="GAT1_ConformidadDF" id="BPMNShape_GAT1_ConformidadDF">
        <omgdc:Bounds height="40.0" width="40.0" x="681.1554498451151" y="395.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="UT4_CodificacionActivos" id="BPMNShape_UT4_CodificacionActivos">
        <omgdc:Bounds height="80.0" width="100.00000000000023" x="865.6644292098593" y="375.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="GAT2_ConformidadCO" id="BPMNShape_GAT2_ConformidadCO">
        <omgdc:Bounds height="40.0" width="40.0" x="1195.2479055635274" y="395.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="UT6_Pruebas_Unitarias_Activos" id="BPMNShape_UT6_Pruebas_Unitarias_Activos">
        <omgdc:Bounds height="80.0" width="100.0" x="1432.4519923350679" y="375.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="GAT3_PruebasUnitarias" id="BPMNShape_GAT3_PruebasUnitarias">
        <omgdc:Bounds height="40.0" width="40.0" x="1602.295027463812" y="395.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="ET3_Errores_en_Pruebas_Unitarias" id="BPMNShape_ET3_Errores_en_Pruebas_Unitarias">
        <omgdc:Bounds height="80.0" width="100.0" x="1242.608957206324" y="155.3261196507905"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="ET2_ErroresValidacionDF" id="BPMNShape_ET2_ErroresValidacionDF">
        <omgdc:Bounds height="80.0" width="100.0" x="465.97835895237154" y="155.3261196507905"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="UT7_Validacion_Pruebas_Unitarias" id="BPMNShape_UT7_Validacion_Pruebas_Unitarias">
        <omgdc:Bounds height="79.9999999999994" width="100.0" x="1846.6549780705093" y="375.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>

```

```
<bpmndi:BPMNShape bpmnElement="UT5_Validacion_Activos" id="BPMNShape_UT5_Validacion_Activos">
  <omgdc:Bounds height="80.0" width="100.00000000000023" x="1035.5074643386033" y="375.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="GAT4_ConformidadPU" id="BPMNShape_GAT4_ConformidadPU">
  <omgdc:Bounds height="40.0" width="40.0" x="2088.2733864161833" y="395.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="GAT5_PruebasFuncionales" id="BPMNShape_GAT5_PruebasFuncionales">
  <omgdc:Bounds height="40.0" width="40.0" x="2563.747756828339" y="395.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="UT9_Validacion_Pruebas_Funcionales" id="BPMNShape_UT9_Validacion_Pruebas_Funcionales">
  <omgdc:Bounds height="80.0" width="100.0" x="2761.3532382362755" y="375.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="GAT6_ConformidadPF" id="BPMNShape_GAT6_ConformidadPF">
  <omgdc:Bounds height="40.0" width="40.0" x="2958.295107668732" y="395.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="UT10_Pruebas_Integracion" id="BPMNShape_UT10_Pruebas_Integracion">
  <omgdc:Bounds height="80.0" width="100.0" x="3075.0" y="375.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="ET4_Errores_Pruebas_Funcionales" id="BPMNShape_ET4_Errores_Pruebas_Funcionales">
  <omgdc:Bounds height="80.0" width="100.0" x="1484.2273655519982" y="604.0460208641853"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="GAT7_ConformidadPI" id="BPMNShape_GAT7_ConformidadPI">
  <omgdc:Bounds height="40.0" width="40.0" x="3335.396795770736" y="395.0000140987919"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="UT11_Aprobacion_Despliegue" id="BPMNShape_UT11_Aprobacion_Despliegue">
  <omgdc:Bounds height="80.0" width="100.0" x="3465.0" y="375.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="ET5_2_Paralizacion_Despliegue" id="BPMNShape_ET5_2_Paralizacion_Despliegue">
  <omgdc:Bounds height="80.00000000000006" width="100.0" x="3883.153137263805" y="209.8430386974872"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="ET1_Comienzo_Evolutivo" id="BPMNShape_ET1_Comienzo_Evolutivo">
  <omgdc:Bounds height="80.0" width="100.0" x="120.0" y="155.3261196507905"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="UT8_Pruebas_Funcionales" id="BPMNShape_UT8_Pruebas_Funcionales">
  <omgdc:Bounds height="80.0" width="100.0" x="2220.0" y="375.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="sid-741D59CF-5A16-4C1F-A685-FC3543C5C732" id="BPMNShape_sid-741D59CF-5A16-4C1F-A685-FC3543C5C732">
  <omgdc:Bounds height="28.0" width="28.0" x="4110.0" y="401.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="sid-A7F526EF-0199-4720-B92A-C6037744A4C9" id="BPMNShape_sid-A7F526EF-0199-4720-B92A-C6037744A4C9">
  <omgdc:Bounds height="28.0" width="28.0" x="4110.000077261379" y="235.84304313095646"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="GAT8_ConformidadDespliegue" id="BPMNShape_GAT8_ConformidadDespliegue">
  <omgdc:Bounds height="40.0" width="40.0" x="3660.0" y="395.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="ET5_1_InformeFinal" id="BPMNShape_ET5_1_InformeFinal">
  <omgdc:Bounds height="80.00000000000006" width="100.0" x="3883.1532102608285" y="375.0000140987919"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="ET5_3_Paralizacion_Despliegue" id="BPMNShape_ET5_3_Paralizacion_Despliegue">
  <omgdc:Bounds height="80.00000000000006" width="100.0" x="3883.153137263805" y="60.0"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNShape bpmnElement="sid-ED53C347-F389-4B77-A9E9-CDF21FDEAB5C" id="BPMNShape_sid-ED53C347-F389-4B77-A9E9-CDF21FDEAB5C">
  <omgdc:Bounds height="28.0" width="28.0" x="4110.0" y="86.00000000000003"></omgdc:Bounds>
</bpmndi:BPMNShape>
<bpmndi:BPMNEdge bpmnElement="sid-ED9B3365-1BA1-47DD-8959-7A1A8159B7B9" id="BPMNEdge_sid-ED9B3365-1BA1-47DD-8959-7A1A8159B7B9">
  <omgdi:waypoint x="1242.608957206324" y="195.28293745866202"></omgdi:waypoint>
```

```
<omgdi:waypoint x="915.0" y="195.0"></omgdi:waypoint>
<omgdi:waypoint x="915.543623898976" y="375.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-8A04E411-48CA-4049-B041-F035696937CD" id="BPMNEdge_sid-8A04E411-48CA-4049-B041-
F035696937CD">
<omgdi:waypoint x="2108.2386206592537" y="434.96523424307026"></omgdi:waypoint>
<omgdi:waypoint x="2108.0" y="572.0"></omgdi:waypoint>
<omgdi:waypoint x="1482.0" y="572.0"></omgdi:waypoint>
<omgdi:waypoint x="1482.336835052248" y="455.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-57F347EA-41BA-4ED6-AFC4-1AD66E6E2CEA" id="BPMNEdge_sid-57F347EA-41BA-4ED6-AFC4-
1AD66E6E2CEA">
<omgdi:waypoint x="2861.3532382362755" y="415.0"></omgdi:waypoint>
<omgdi:waypoint x="2958.295107668732" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-1400CD61-B0A6-40BD-9FF6-B1E297323637" id="BPMNEdge_sid-1400CD61-B0A6-40BD-9FF6-
B1E297323637">
<omgdi:waypoint x="67.0370376242574" y="415.0"></omgdi:waypoint>
<omgdi:waypoint x="120.0" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-4F5D20EA-9760-4F67-8DD3-F7E4B26A815E" id="BPMNEdge_sid-4F5D20EA-9760-4F67-8DD3-
F7E4B26A815E">
<omgdi:waypoint x="2128.2733864161833" y="415.0"></omgdi:waypoint>
<omgdi:waypoint x="2220.0" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-56D6D3A4-58EE-454B-86B9-96048FCDCA9E" id="BPMNEdge_sid-56D6D3A4-58EE-454B-86B9-
96048FCDCA9E">
<omgdi:waypoint x="3375.3967940040056" y="415.0000123320616"></omgdi:waypoint>
<omgdi:waypoint x="3465.0" y="415.0000044168261"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-2B1F1758-D64B-4B1B-9555-88F9FDD30643" id="BPMNEdge_sid-2B1F1758-D64B-4B1B-9555-
88F9FDD30643">
<omgdi:waypoint x="1622.2682427045545" y="395.0267847592575"></omgdi:waypoint>
<omgdi:waypoint x="1622.0" y="195.0"></omgdi:waypoint>
<omgdi:waypoint x="1342.608957206324" y="195.2766162326003"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-ABE2D226-8471-4FA1-8509-7CA9B7392AE4" id="BPMNEdge_sid-ABE2D226-8471-4FA1-8509-
7CA9B7392AE4">
<omgdi:waypoint x="721.1554498451151" y="415.0"></omgdi:waypoint>
<omgdi:waypoint x="865.6644292098593" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-AAAF82F0-4464-4AB5-9E4D-18606C24B335" id="BPMNEdge_sid-AAAF82F0-4464-4AB5-9E4D-
18606C24B335">
<omgdi:waypoint x="1215.2159688281972" y="434.9680632646698"></omgdi:waypoint>
<omgdi:waypoint x="1215.0" y="570.0"></omgdi:waypoint>
<omgdi:waypoint x="915.0" y="570.0"></omgdi:waypoint>
<omgdi:waypoint x="915.4929636073151" y="455.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-DF776A4E-461A-4D8B-9C9C-80928A4BDB83" id="BPMNEdge_sid-DF776A4E-461A-4D8B-9C9C-
80928A4BDB83">
<omgdi:waypoint x="3680.0" y="395.0"></omgdi:waypoint>
<omgdi:waypoint x="3680.0" y="249.84303869748726"></omgdi:waypoint>
<omgdi:waypoint x="3883.153137263805" y="249.84303869748723"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-56BADC35-CD63-4926-B158-913FFF5D0E4D" id="BPMNEdge_sid-56BADC35-CD63-4926-B158-
913FFF5D0E4D">
<omgdi:waypoint x="219.9999999999997" y="415.0"></omgdi:waypoint>
<omgdi:waypoint x="321.15542108357954" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
```

```
<bpmndi:BPMNEdge bpmnElement="sid-D5E35CBC-AABF-4512-AA5E-56302FCFE3E2" id="BPMNEdge_sid-D5E35CBC-AABF-4512-AA5E-56302FCFE3E2">
  <omgdi:waypoint x="2978.2679457439285" y="395.02716192480335"></omgdi:waypoint>
  <omgdi:waypoint x="2978.0" y="198.0"></omgdi:waypoint>
  <omgdi:waypoint x="2270.0" y="198.0"></omgdi:waypoint>
  <omgdi:waypoint x="2270.0" y="375.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-590354FF-8BBF-4AFE-BFA3-FA76671DAB30" id="BPMNEdge_sid-590354FF-8BBF-4AFE-BFA3-FA76671DAB30">
  <omgdi:waypoint x="1642.295027463812" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="1846.6549780705093" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-956A4D78-48E9-4F3B-828C-039486D22FAA" id="BPMNEdge_sid-956A4D78-48E9-4F3B-828C-039486D22FAA">
  <omgdi:waypoint x="965.6644292098596" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="1035.5074643386033" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-6FDA76D5-DE75-493A-A2D2-99ED64BA3742" id="BPMNEdge_sid-6FDA76D5-DE75-493A-A2D2-99ED64BA3742">
  <omgdi:waypoint x="3565.0" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="3660.0" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-DA569EE0-BB34-4A88-B91B-ABF9DB6C0129" id="BPMNEdge_sid-DA569EE0-BB34-4A88-B91B-ABF9DB6C0129">
  <omgdi:waypoint x="421.15542108357954" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="503.23681669134817" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-A36D8BBC-76F8-4937-ADD9-5F5C6950D4FE" id="BPMNEdge_sid-A36D8BBC-76F8-4937-ADD9-5F5C6950D4FE">
  <omgdi:waypoint x="1532.4519923350679" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="1602.295027463812" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-5DA8EDD9-84B9-49CA-8FD5-03FE795CD784" id="BPMNEdge_sid-5DA8EDD9-84B9-49CA-8FD5-03FE795CD784">
  <omgdi:waypoint x="2603.747756828339" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="2761.3532382362755" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-A76D6080-F7B5-4590-AED2-6FE4AE050902" id="BPMNEdge_sid-A76D6080-F7B5-4590-AED2-6FE4AE050902">
  <omgdi:waypoint x="2998.295107668732" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="3075.0" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-5B03B468-404A-4459-A49A-909A630C8183" id="BPMNEdge_sid-5B03B468-404A-4459-A49A-909A630C8183">
  <omgdi:waypoint x="1946.6549780705093" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="2088.2733864161833" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-0786241F-6418-4913-AD6F-9FC671A36E88" id="BPMNEdge_sid-0786241F-6418-4913-AD6F-9FC671A36E88">
  <omgdi:waypoint x="3983.153137263805" y="249.84303985901207"></omgdi:waypoint>
  <omgdi:waypoint x="4110.000077261379" y="249.8430428057295"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-66BFE900-40FE-4704-B7DE-9178C60DF2B2" id="BPMNEdge_sid-66BFE900-40FE-4704-B7DE-9178C60DF2B2">
  <omgdi:waypoint x="701.1413280193575" y="395.0141218257576"></omgdi:waypoint>
  <omgdi:waypoint x="701.0" y="195.0"></omgdi:waypoint>
  <omgdi:waypoint x="565.9783589523715" y="195.23798951397407"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-7C53637E-E390-4210-8CBC-54C9779BC6CB" id="BPMNEdge_sid-7C53637E-E390-4210-8CBC-54C9779BC6CB">
```

```
<omgdi:waypoint x="2320.0" y="415.0"></omgdi:waypoint>
<omgdi:waypoint x="2563.747756828339" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-49EA735D-884F-4702-A929-141B24CA6425" id="BPMNEdge_sid-49EA735D-884F-4702-A929-141B24CA6425">
  <omgdi:waypoint x="3355.896795770736" y="395.5000140987919"></omgdi:waypoint>
  <omgdi:waypoint x="3355.896795770736" y="100.00000000000003"></omgdi:waypoint>
  <omgdi:waypoint x="3883.153137263805" y="100.00000000000003"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-A846C32B-2FDF-4470-BE94-928BB670A600" id="BPMNEdge_sid-A846C32B-2FDF-4470-BE94-928BB670A600">
  <omgdi:waypoint x="1235.2479055635274" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="1432.4519923350679" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-421293CB-E404-4FA5-89E1-D5910A8F2294" id="BPMNEdge_sid-421293CB-E404-4FA5-89E1-D5910A8F2294">
  <omgdi:waypoint x="3175.0" y="415.0000030596762"></omgdi:waypoint>
  <omgdi:waypoint x="3335.396796994606" y="415.0000128749215"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-21EA5ED8-2CB6-4CE7-AB07-575BEB81A4EF" id="BPMNEdge_sid-21EA5ED8-2CB6-4CE7-AB07-575BEB81A4EF">
  <omgdi:waypoint x="2583.6826631065405" y="434.93490627820165"></omgdi:waypoint>
  <omgdi:waypoint x="2583.0" y="644.0"></omgdi:waypoint>
  <omgdi:waypoint x="1584.2273655519982" y="644.0438268298125"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-AB238A46-69BE-466C-90DA-D96A8F573322" id="BPMNEdge_sid-AB238A46-69BE-466C-90DA-D96A8F573322">
  <omgdi:waypoint x="1484.2273655519982" y="644.0436757979039"></omgdi:waypoint>
  <omgdi:waypoint x="553.0" y="644.0"></omgdi:waypoint>
  <omgdi:waypoint x="553.195451330414" y="455.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-FED4C7E9-FC30-47B5-B6C9-4B8C0CB67AFE" id="BPMNEdge_sid-FED4C7E9-FC30-47B5-B6C9-4B8C0CB67AFE">
  <omgdi:waypoint x="3983.153137263805" y="100.00000000000003"></omgdi:waypoint>
  <omgdi:waypoint x="4110.0" y="100.00000000000003"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-4C03116A-4E58-4C14-B349-57AF6DDE2982" id="BPMNEdge_sid-4C03116A-4E58-4C14-B349-57AF6DDE2982">
  <omgdi:waypoint x="1135.5074643386035" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="1195.2479055635274" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-D2A57FA3-C425-4A41-A382-E76B3E287998" id="BPMNEdge_sid-D2A57FA3-C425-4A41-A382-E76B3E287998">
  <omgdi:waypoint x="603.2368166913482" y="415.0"></omgdi:waypoint>
  <omgdi:waypoint x="681.1554498451151" y="415.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-4413BB52-DD96-426A-9B56-F8F9909BB0ED" id="BPMNEdge_sid-4413BB52-DD96-426A-9B56-F8F9909BB0ED">
  <omgdi:waypoint x="3699.999988861455" y="415.000011138544"></omgdi:waypoint>
  <omgdi:waypoint x="3883.1532102608285" y="415.0000113141557"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-7DC9D0F3-E4FC-4A47-BE3E-50E49DC56F31" id="BPMNEdge_sid-7DC9D0F3-E4FC-4A47-BE3E-50E49DC56F31">
  <omgdi:waypoint x="465.97835895237154" y="195.2136478125289"></omgdi:waypoint>
  <omgdi:waypoint x="371.0" y="195.0"></omgdi:waypoint>
  <omgdi:waypoint x="371.12716270474687" y="375.0"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-11C24E3F-B493-4C9C-A4A9-BAAE51F5F8AE" id="BPMNEdge_sid-11C24E3F-B493-4C9C-A4A9-BAAE51F5F8AE">
  <omgdi:waypoint x="3983.1532102608285" y="415.00001040504577"></omgdi:waypoint>
```



```
<omgdi:waypoint x="4110.0" y="415.0000010342489"></omgdi:waypoint>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="sid-7D35818F-B66D-40B4-96C9-11CDB1FDB07C" id="BPMNEdge_sid-7D35818F-B66D-40B4-96C9-
11CDB1FDB07C">
  <omgdi:waypoint x="170.0" y="375.0"></omgdi:waypoint>
  <omgdi:waypoint x="170.0" y="235.3261196507905"></omgdi:waypoint>
</bpmndi:BPMNEdge>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</definitions>
```

9.2 BackEnd del Sistema

Application.java

```
package com.uah.cofabpm;

import org.activiti.engine.ProcessEngine;
import org.activiti.engine.RepositoryService;
import org.activiti.engine.RuntimeService;
import org.activiti.engine.TaskService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
@ComponentScan
@EnableAutoConfiguration
public class Application {

    @Autowired
    private ProcessEngine processEngine;

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    @Bean
    public CommandLineRunner init(
        final RepositoryService repositoryService,
        final RuntimeService runtimeService,
        final TaskService taskService) {
        return new CommandLineRunner() {
            public void run(String... strings) throws Exception {
                String pName = processEngine.getName();
                String ver = ProcessEngine.VERSION;
                System.out.println("Nombre del Motor y Version: ProcessEngine [" +
                    pName + "] Version: [" + ver + "]");
            }
        };
    }
}
```

SeguridadController.java

```
package com.uah.cofabpm;

import java.util.HashMap;
import java.util.Map;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController

public class SeguridadController {

    @RequestMapping(value = "/api/seguridad", method = RequestMethod.GET)
    public Map<String, Object> getSeguridad() {
        Map<String, Object> login = new HashMap<String, Object>();
        Authentication authentication =
            SecurityContextHolder.getContext().getAuthentication();
        login.put("usuario", authentication.getName());
        login.put("rol",
            authentication.getAuthorities().iterator().next().getAuthority().substring(5));
        return login;
    }
}
```

EvolutivoController.java

```
package com.uah.cofabpm;

import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.io.IOException;
import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.ResourceLoader;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.activiti.engine.HistoryService;
import org.activiti.engine.RepositoryService;
import org.activiti.engine.RuntimeService;
import org.activiti.engine.TaskService;
import org.activiti.engine.history.HistoricActivityInstance;
import org.activiti.engine.history.HistoricVariableInstance;
import org.activiti.engine.repository.ProcessDefinition;
import org.activiti.engine.runtime.ProcessInstance;
import org.activiti.engine.task.Task;
import org.apache.commons.io.IOUtils;

@RestController
public class EvolutivoController {
    @Autowired
    private RepositoryService repositoryService;
    @Autowired
    private RuntimeService runtimeService;
    @Autowired
    private TaskService taskService;
    @Autowired
    private HistoryService historyService;
    @Autowired
    private EvolutivoRepository evolutivoRepository;
    // Cargar los Templates de los Email a enviar
    @Autowired
    private ResourceLoader resourceLoader;
```

```

public String getResources(String nombreFichero) throws IOException {
    String content = IOUtils.toString(resourceLoader.getResource("classpath:" +
    nombreFichero + ".html").getInputStream(),"UTF-8");
    return content;
}

// Información General del Motor
@RequestMapping(method = RequestMethod.GET, value = "/api/informacion",
produces=MediaType.IMAGE_PNG_VALUE) public byte[] getInfo() throws IOException {
    ProcessDefinition processDefinition = repositoryService.createProcessDefinitionQuery()
    .processDefinitionKey("BPM_Evolutivo").singleResult();
    String diagramResourceName = processDefinition.getDiagramResourceName();
    InputStream imageStream =repositoryService
    .getResourceAsStream(processDefinition.getDeploymentId(),
    diagramResourceName);

    System.out.println("Número de procesos definidos : " +
    repositoryService.createProcessDefinitionQuery().count());
    System.out.println("Número de tareas : " + taskService.createTaskQuery().count());
    System.out.println("Número de tareas después de empezar: " +
    taskService.createTaskQuery().count());
    System.out.println("Nombre de tareas: " + taskService.createTaskQuery().list());
    System.out.println("Lista devuelta por el runtimeService: " +
    runtimeService.createExecutionQuery().list());
    return IOUtils.toByteArray(imageStream);
}

@RequestMapping(value = "/api/evolutivos", method = RequestMethod.GET)
public List <Evolutivo> getEvolutivos() {
    List <HistoricActivityInstance> actividades =
    historyService.createHistoricActivityInstanceQuery()
    .activityId("UT1_Alta_Proyecto").list();
    List <Evolutivo> mensajeRespuesta = new ArrayList<Evolutivo>();
    for (HistoricActivityInstance actividad: actividades) {
        String idProceso = actividad.getProcessInstanceId();
        List<HistoricVariableInstance> variablesEvolutivo =
        HistoryService.createHistoricVariableInstanceQuery()
        .processInstanceId(idProceso).list();
        int cont = 0;
        boolean encontrado = false;
        while (cont < variablesEvolutivo.size() && !encontrado) {
            if (variablesEvolutivo.get(cont).getVariableName() == "evolutivo") {
                Evolutivo evoTemp = (Evolutivo) variablesEvolutivo.get(cont).getValue();
                evoTemp.setIdProceso(idProceso);
                mensajeRespuesta.add(evoTemp);
            }
        }
    }
}

```

```
                encontrado = true;
            }
            cont++;
        };
    }
}

return mensajeRespuesta;
}

// Abrir el Evolutivo del Proyecto
@RequestMapping(value = "/api/abrirevolutivo", method = RequestMethod.POST, produces =
    MediaType.APPLICATION_JSON_VALUE)
public Evolutivo inicioEvolutivoProcess(@RequestBody Map<String, String> data) {
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    String usuario = data.get("usuario");
    Date fechalInicioParsed;
    try {
        fechalInicioParsed = df.parse(data.get("fechalInicio"));
    } catch (ParseException e) {
        fechalInicioParsed = new Date();
    }
    Date fechaFinParsed;
    try {
        fechaFinParsed = df.parse(data.get("fechaFin"));
    } catch (ParseException e) {
        fechaFinParsed = null;
    }
    Evolutivo evolutivo = new Evolutivo(data.get("codigo"), data.get("nombre"),
    data.get("descripcion"), fechalInicioParsed,
    fechaFinParsed, "");

    ProcessInstance procesoNuevo = runtimeService.startProcessInstanceByKey("BPM_Evolutivo");
    evolutivo.setIdProceso(procesoNuevo.getId());
    evolutivoRepository.save(evolutivo);

    Task tarea = taskService.createTaskQuery()
        .processInstanceId(procesoNuevo.getId()).singleResult();

    String templateHeader = "";
    String templateFooter = "";

    try {
        templateHeader = getResources("templateHeader");
    } catch (IOException e) {
        e.printStackTrace();
    }
    try {
```

```
        templateFooter = getResources("templateFooter");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
/* Templates del Email */
Map<String, Object> taskVariables = new HashMap<String, Object>();
taskVariables.put("evolutivo", evolutivo);
taskVariables.put("templateHeader", templateHeader);
taskVariables.put("templateFooter", templateFooter);
taskService.setVariablesLocal(tarea.getId(), taskVariables);

taskService.claim(tarea.getId(), usuario);
taskService.complete(tarea.getId(), taskVariables);
return evolutivo;
}

// Devolver toda la información del Estado
@RequestMapping(value = "/api/evolutivoestado", method = RequestMethod.POST, produces =
    MediaType.APPLICATION_JSON_VALUE)
public ArrayList<Tarea> getTareasEvolutivo(@RequestBody Map<String, String> data) {
    String idProceso = data.get("idProceso");
    ArrayList<Tarea> infoTareas = new ArrayList<Tarea>();
    List<HistoricActivityInstance> activities =
        historyService.createHistoricActivityInstanceQuery()
            .processInstanceId(idProceso)
            .orderByHistoricActivityInstanceStartTime().asc().list();

    for (HistoricActivityInstance activity : activities) {
        Tarea tareaAux = new Tarea();
        tareaAux.setIdTarea(activity.getTaskId());
        tareaAux.setNombre(activity.getActivityName());
        tareaAux.setDuracion(activity.getDurationInMillis());
        tareaAux.setTipo(activity.getActivityType());
        tareaAux.setIdActividad(activity.getActivityId());
        tareaAux.setFechaInicio(activity.getStartTime());
        tareaAux.setFechaFin(activity.getEndTime());
        if (activity.getTaskId() != null) {
            Map<String, Object> taskVariables = new HashMap<String, Object>();
            List<HistoricVariableInstance> variablesTarea =
                historyService.createHistoricVariableInstanceQuery()
                    .taskId(activity.getTaskId()).list();
            for (HistoricVariableInstance variableTarea : variablesTarea) {
                // Se descompone el JPA para homogeneizar el JSON
                if (variableTarea.getVariableName() == "evolutivo") {
                    Evolutivo evoTemp = (Evolutivo) variableTarea.getValue();
                    taskVariables.put("nombre", evoTemp.getNombre());
                }
            }
        }
    }
}
```

```
        taskVariables.put("descripcion", evoTemp.getDescripcion());
        taskVariables.put("fechalnicio", evoTemp.getFechaInicio());
        taskVariables.put("fechaFin", evoTemp.getFechaFin());
    } else {
        // Se omiten los Templates de los Email
        if (!variableTarea.getVariableName().contains("template")) {
            taskVariables.put(variableTarea.getVariableName(),
                variableTarea.getValue());
        }
    }
}

tareaAux.setCampos(taskVariables);
}

infoTareas.add(tareaAux);
}

return infoTareas;
}
}

// Devolver toda la información del Estado
@RequestMapping(value = "/api/prueba" , method = RequestMethod.POST, produces =
    MediaType.APPLICATION_JSON_VALUE)
public List<HistoricActivityInstance> evolutivoEstado(@RequestBody Map<String, String> data) {
    String idProceso = data.get("idProceso");
    HistoricActivityInstance endActivity = null;
    List<HistoricActivityInstance> activities =
        historyService.createHistoricActivityInstanceQuery()
            .processInstanceId(idProceso).orderByExecutionId().asc().list();

    for (HistoricActivityInstance activity : activities) {
        if (activity.getActivityType() == "startEvent") {
            System.out.println("BEGIN Nombre"+ activity.getActivityName() + "Empieza" +
                activity.getStartTime());
        }
        if (activity.getActivityType() == "endEvent") {
            endActivity = activity;
        } else {
            System.out.println("-- " + activity.getActivityName() + " [" +
                activity.getActivityId() + "]" + activity.getDurationInMillis() + " ms");
        }
    }

    if (endActivity != null) {
        System.out.println("-- " + endActivity.getActivityName() + " [" +
            endActivity.getActivityId() + "]" + endActivity.getDurationInMillis() + " ms");
        System.out.println("COMPLETE " + endActivity.getEndTime());
    }

    return activities; }
}
```


Evolutivo.java

```
package com.uah.cofabpm;

import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.text.SimpleDateFormat;

public class Tarea {
    private String idTarea;
    private String nombre;
    private String tipo;
    private String idActividad;
    private Date fechaInicio;
    private Date fechaFin;
    private Long duracion;
    private String asignado;
    private Map<String, Object> campos;
    public String getIdActividad() {
        return idActividad;
    }
    public void setIdActividad(String idActividad) {
        this.idActividad = idActividad;
    }

    public String getIdTarea() {
        return idTarea;
    }

    public void setIdTarea(String idTarea) {
        this.idTarea = idTarea;
    }

    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getTipo() {
        return tipo;
    }
    public void setTipo(String tipo) {
        this.tipo = tipo;
    }
}
```

```
public Date getFechaInicio() {
    return fechaInicio;
}

public void setFechaInicio(Date fechaInicio) {
    this.fechaInicio = fechaInicio;
}

public Date getFechaFin() {
    return fechaFin;
}

public void setFechaFin(Date fechaFin) {
    this.fechaFin = fechaFin;
}

public Long getDuracion() {
    return duracion;
}

public void setDuracion(Long duracion) {
    this.duracion = duracion;
}

public String getAsignado() {
    return asignado;
}

public void setAsignado(String asignado) {
    this.asignado = asignado;
}

public Map<String, Object> getCampos() {
    return campos;
}

public void setCampos(Map<String, Object> campos) {
    this.campos = campos;
}

public Tarea() {
```

EvolutivoRepository.java

```
package com.uah.cofabpm;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EvolutivoRepository extends JpaRepository<Evolutivo, Long> {
```

TareaController.java

```
package com.uah.cofabpm;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.io.IOException;
import java.text.SimpleDateFormat;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.ResourceLoader;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.activiti.engine.HistoryService;
import org.activiti.engine.TaskService;
import org.activiti.engine.history.HistoricActivityInstance;
import org.activiti.engine.history.HistoricVariableInstance;
import org.activiti.engine.task.Task;
import org.apache.commons.io.IOUtils;

@RestController
public class TareaController {

    @Autowired
    private TaskService taskService;

    @Autowired
    private HistoryService historyService;

    // Cargar los Templates de los Email a enviar
    @Autowired
    private ResourceLoader resourceLoader;

    public String getResources(String nombreFichero) throws IOException {
        String content = IOUtils.toString(resourceLoader.getResource("classpath:" +
            nombreFichero + ".html").getInputStream(), "UTF-8");
        return content;
    }

    public String getNombreEvolutivo(String idProceso) {
        List<HistoricVariableInstance> variablesEvolutivo =
            historyService.createHistoricVariableInstanceQuery().processInstanceId(idProceso).list();
        int cont = 0;
        boolean encontrado = false;
        Evolutivo evoTemp = new Evolutivo();
        while (cont < variablesEvolutivo.size() && !encontrado) {
```

```
        if (variabalesEvolutivo.get(cont).getVariableName() == "evolutivo") {
            evoTemp = (Evolutivo) variabalesEvolutivo.get(cont).getValue();
            evoTemp.setIdProceso(idProceso);
            encontrado = true;
        }
        cont++;
    };
    return evoTemp.getCodigo() + "-" + evoTemp.getDescripcion();
}

// Buscar las tareas Abiertas de Un Grupo
@RequestMapping(value = "/api/buscartareasabiertas", method = RequestMethod.POST, produces =
    MediaType.APPLICATION_JSON_VALUE)
public ArrayList < Map<String,String> > buscarTareasAbiertas(@RequestBody Map<String, String> data) {
    String usuario = data.get("usuario");
    String grupo = data.get("grupo");
    List<Task> tareas = taskService.createTaskQuery().taskCandidateGroup(grupo).list();
    ArrayList < Map<String,String> > tareasAbiertas = new ArrayList<Map<String,String>>();
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");

    for (Task tarea:tareas) {
        Map<String,String> tareaAbierta = new HashMap<String, String>();
        tareaAbierta.put("id", tarea.getId());
        tareaAbierta.put("idActividad",tarea.getTaskDefinitionKey());
        tareaAbierta.put("evolutivo",getNombreEvolutivo(tarea.getProcessInstanceId()));
        tareaAbierta.put("nombre",tarea.getName());
        tareaAbierta.put("descripcion",tarea.getDescription());
        tareaAbierta.put("fecha", df.format(tarea.getCreateTime()));
        tareasAbiertas.add(tareaAbierta);
    }
    return tareasAbiertas;
}

// Buscar las tareas Realizadas por ese Usuario
@RequestMapping(value = "/api/buscartareascerradas", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE)
public ArrayList < Map<String,String>> buscarTareasCerradas(@RequestBody Map<String, String> data) {
    String usuario = data.get("usuario");
    String grupo = data.get("grupo");
    ArrayList < Map<String,String> > tareasCerradas = new
    ArrayList<Map<String,String>>();
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    List<HistoricActivityInstance> realizadas =
    historyService.createHistoricActivityInstanceQuery().taskAssignee(usuario).orderByHistori
    cActivityInstanceStartTime().asc().list();

    for (HistoricActivityInstance realizada:realizadas) {
```

```
Map<String, String> tareaCerrada = new HashMap<String, String>();
tareaCerrada.put("evolutivo", getNombreEvolutivo(realizada.getProcessInstanceId()));
tareaCerrada.put("nombre", realizada.getActivityName());
tareaCerrada.put("fecha", df.format(realizada.getTime()));
tareasCerradas.add(tareaCerrada);

}

return tareasCerradas;
}

// Completar Tarea
@RequestMapping(value = "/api/completartarea", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_VALUE)
public Boolean completarTarea(@RequestBody Map<String, String> data) {

    String usuario = data.get("usuario");
    String grupo = data.get("grupo");
    String tareald = data.get("id");

    data.remove("usuario");
    data.remove("grupo");
    data.remove("id");

    taskService.claim(tareald, usuario);
    Map<String, Object> taskVariables = new HashMap<String, Object>();

    for (Map.Entry<String, String> entry : data.entrySet())
    {
        System.out.println(entry.getKey() + "/" + entry.getValue());
        taskVariables.put(entry.getKey(), entry.getValue());
    }

    String templateHeader = "";
    String templateFooter = "";

    try {
        templateHeader = getResources("templateHeader");
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        templateFooter = getResources("templateFooter");
    } catch (IOException e) {
        e.printStackTrace();
    }

    /* Templates del Email */
    taskVariables.put("templateHeader", templateHeader);
    taskVariables.put("templateFooter", templateFooter);
    taskService.setVariablesLocal(tareald, taskVariables);
```

```
        taskService.complete(tareald,taskVariables);  
  
        return true;  
    }  
}
```

Tarea.java

```
package com.uah.cofabpm;  
  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
import java.text.SimpleDateFormat;  
  
public class Tarea {  
    private String idTarea;  
    private String nombre;  
    private String tipo;  
    private String idActividad;  
    private Date fechalnicio;  
    private Date fechaFin;  
    private Long duracion;  
    private String asignado;  
    private Map<String, Object> campos;  
    public String getIdActividad() {  
        return idActividad;  
    }  
    public void setIdActividad(String idActividad) {  
        this.idActividad = idActividad;  
    }  
  
    public String getIdTarea() {  
        return idTarea;  
    }  
  
    public void setIdTarea(String idTarea) {  
        this.idTarea = idTarea;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
    public void setNombre(String nombre) {  
        this.nombre = nombre; }  
    public String getTipo() {  
        return tipo;  
    }  
}
```

```
}  
  
public void setTipo(String tipo) {  
    this.tipo = tipo;  
}  
  
public Date getFechaInicio() {  
    return fechaInicio;  
}  
  
public void setFechaInicio(Date fechaInicio) {  
    this.fechaInicio = fechaInicio;  
}  
  
public Date getFechaFin() {  
    return fechaFin;  
}  
  
public void setFechaFin(Date fechaFin) {  
    this.fechaFin = fechaFin;  
}  
  
public Long getDuracion() {  
    return duracion;  
}  
  
public void setDuracion(Long duracion) {  
    this.duracion = duracion;  
}  
  
public String getAsignado() {  
    return asignado;  
}  
  
public void setAsignado(String asignado) {  
    this.asignado = asignado;  
}  
  
public Map<String, Object> getCampos() {  
    return campos;  
}  
  
public void setCampos(Map<String, Object> campos) {  
    this.campos = campos;  
}  
  
public Tarea() {
```

MVConfig.java

```
package com.uah.cofabpm;

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

@Configuration
public class MvcConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers(ViewControllerRegistry registry) {
        registry.addViewController("/").setViewName("index");
        registry.addViewController("/login").setViewName("login");
    }
}
```


9.3 FrontEnd

app.js

```
/*global angular*/

'use strict';

angular.module('cofabpm', ['ui.router', 'cofabpm.controllers', 'cofabpm.directives']);

angular.module('cofabpm').config(function($stateProvider, $locationProvider, $urlRouterProvider) {

    $urlRouterProvider.otherwise('/menu');

    $stateProvider

        // login
        .state('stateLogin', {
            url: '/login',
            controller: 'Controller1',
            templateUrl: 'html/partials/login.html'
        })

        // menu inicial
        .state('stateMenu', {
            url: '/menu',
            controller: 'Controller2',
            templateUrl: 'html/partials/menu.html'
        })

        // buzones
        .state('stateBuzones', {
            url: '/buzones',
            controller: 'Controller3',
            templateUrl: 'html/partials/buzones.html'
        })

        // evolutivos
        .state('stateEvolutivos', {
            url: '/evolutivos',
            controller: 'Controller4',
            templateUrl: 'html/partials/evolutivos.html'
        })

        // sala de maquinas
        .state('stateSalaMaquinas', {
            url: '/salamaquinas',
```

```
controller: 'Controller5',
templateUrl: 'html/partials/salamaquinas.html'
})

// consulta detalle evolutivos
.state('stateDetalleEvolutivo', {
  url: '/detalleevolutivo',
  controller: 'Controller6',
  templateUrl: 'html/partials/evolutivo.html',
  params: {
    registro: null
  }
})

// formulario
.state('stateFormulario', {
  url: '/formulario',
  controller: 'Controller7',
  templateUrl: 'html/partials/formulario.html',
  params: {
    registro: null
  }
})

// cierra la sesion
.state('stateCerrarSesion', {
  url: '/salida',
  template: "",
  controller: ['$rootScope', function($rootScope) {
    $rootScope.cerrarSesion();
  }],
})

});

angular.module('cofabpm').run(function($rootScope, $timeout, $state, $stateParams, cofabpmService) {

  $rootScope.opcionesCalendario = {
    selectMonths: true, // Creates a dropdown to control month
    monthsFull: ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre'],
    monthsShort: ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic'],
    weekdaysFull: ['Domingo', 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado'],
    weekdaysShort: ['L', 'M', 'X', 'J', 'V', 'S', 'D'],
    firstDay: true,
    today: 'Hoy',
    clear: 'Limpiar',
```

```
close: 'Cerrar',
formatSubmit: 'yyyy-mm-dd',
}

$scope.timeStart = new Date();

$scope.componentesInicio = function() {

    $rootScope.iniciarComponentesComunes();

}

/* Componentes a desactivar cada cambio de estado */
$scope.iniciarComponentesComunes = function() {
    /* Para la cámara y los tooltips */
    $rootScope.$on('$stateChangeStart',
        function(event, toState, toParams, fromState, fromParams) {
            // $rootScope.usuario = "Pedro Palazón - Desarrollador"
            $(':tooltipped').tooltip('remove');
        });

    /* Fija la cabecera de la Aplicacion */
    $('#olasArriba').remove();
    $('#olasAbajo').remove();
    $('#cabeceraAplicacion').removeClass('hide');
    $('#pieAplicacion').removeClass('hide');

    $('.modal').modal();
    $(':tooltipped').tooltip({
        delay: 50
    });
    $('.datepicker').pickadate($rootScope.opcionesCalendario);
    $('.select').material_select();
    $('.materialboxed').materialbox();
    $('.collapsible').collapsible();

    moment.locale('es');

};

$scope.altaProyecto = function() {
    $stateParams.registro = angular.toJson({
        'idActividad': 'UT1_Alta_Proyecto',
        'nombre': 'Alta de Proyecto'
    });
};
```

```
//$stateParams.registro = angular.toJson({'idActividad':'UT11_Aprobacion_Despliegue','nombre':'Aprobación Despliegue'});
$state.go('stateFormulario', $stateParams);
}

$scope.mostrarError = function(mensaje) {
    $('#modalError').modal('open');
    $scope.mensajeError = mensaje;
    $('#carga').addClass('hide');
}

/* Componente de verificación de acceso */
$scope.veficarEntradaCorrecta = function() {
    try {
        $scope.usuario = JSON.parse(atob(sessionStorage.void)).usuario;
        $scope.grupo = JSON.parse(atob(sessionStorage.void)).grupo;
    } catch (err) {
        location.replace("html/error405.html");
    }
}

$scope.cerrarSesion = function() {
    sessionStorage.clear();
    location.replace("html/exit.html");
    cofabpmService.cerrarSesion()
}
})
```

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="cofabpm">

<head>
  <meta charset="utf-8"></meta>
  <meta http-equiv="X-UA-Compatible" content="IE=edge"></meta>
  <meta name="viewport" content="width=device-width, initial-scale=1"></meta>

  <title>CofaBPM</title>

  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet"> </link>

  <link rel="shortcut icon" type="img/png" href="/img/Barco.png"> </link>

  <link rel="stylesheet" type="text/css" href="/library/datatables/media/css/jquery.dataTables.css"> </link>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.1/css/materialize.min.css"> </link>
  <link rel="stylesheet" href="/css/hover-min.css"> </link>
  <link rel="stylesheet" href="/css/animate.css"> </link>
  <link rel="stylesheet" href="/css/styles.css"> </link>
</head>

<body>
  <!-- Cabeceras y Vista Dinámica Enrutada -->

  <header class="enfasisbck hide" id="cabeceraAplicacion">
    <div class="row valign-wrapper center">
      <div class="col s6">
        <div class="col s1" style="cursor: pointer;" onclick="window.location=http://cofabpm.com">
          </img>
        </div>
        <div class="col s5 logo" style="cursor: pointer;" onclick="window.location=http://cofabpm.com">
          <h3 class="logo">CofaBPM</h3>
        </div>
      </div>
      <div class="col s3 offset-s3">
        <a id="logo" class="blue lighten-4 grey-text text-darken-4 waves-effect waves-light btn-large"><i class="material-icons
left">perm_identity</i>{{usuario}} - {{grupo}}</a>
      </div>
    </div>
  </header>
```

```
<!-- Contenedor Principal -->
<main ui-view="" class="" id="main"> </main>

<!-- Pie de Página -->
<footer class="page-footer transparent olas-hang" id="olasAbajo">
  </img>
</footer>

<!-- Pie de Página -->
<footer class="page-footer enfasisbck white-text hide" id="pieAplicacion">
  <div class="row">
    <div class="center-align">
      © UAH TFG Ingeniería Informática 2017. Pedro Palazón Martínez de Alegría
    </div>
  </div>
</footer>

<!-- Modal Grabacion -->
<div id="modalError" class="modal modal-fixed-footer red">
  <div class="modal-content">
    <h1 class="white-text center"><i class="material-icons large">error</i></h1>
    <h5 class="white-text center">{{mensajeError}}</h5>
  </div>
  <div class="modal-footer red">
    <a class="modal-action modal-close waves-effect waves-white btn-flat white-text">Aceptar</a>
  </div>
</div>

<script src="/.library/jquery/dist/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.1/js/materialize.min.js"></script>
<script src="/.library/angular/angular.min.js"></script>
<script src="/.library/angular-ui-router/release/angular-ui-router.min.js"></script>
<script src="/.library/datatables/media/js/jquery.dataTables.js"></script>
<script src="/.library/angular-datatables/dist/angular-datatables.js"></script>
<script src="/.library/moment/min/moment-with-locales.min.js"></script>

<script src="/.js/directives.js"></script>
<script src="/.js/filters.js"></script>
<script src="/.js/services.js"></script>
<script src="/.js/controllers/controller0.js"></script>
<script src="/.js/controllers/controller1.js"></script>
<script src="/.js/controllers/controller2.js"></script>
<script src="/.js/controllers/controller3.js"></script>
<script src="/.js/controllers/controller4.js"></script>
<script src="/.js/controllers/controller5.js"></script>
```

```
<script src="/js/controllers/controller6.js"></script>
<script src="/js/controllers/controller7.js"></script>

<script src="/js/app.js"></script>

</body>

</html>
```

controller2.js

```
'use strict';

angular.module('cofabpm.controllers')
.controller('Controller2', ['$scope', '$rootScope', '$state', '$stateParams', '$timeout', 'cofabpmService',
function($scope, $rootScope, $state, $stateParams, $timeout, cofabpmService) {

    $scope.init = function(usuario) {

        cofabpmService.seguridad()
        .then(
            function(res) {
                sessionStorage.void = btoa(JSON.stringify({
                    'usuario': res.data.usuario,
                    'grupo': res.data.rol
                }));
                $rootScope.usuario = res.data.usuario;
                $rootScope.grupo = res.data.rol
            },
            function(reason) {
                location.replace("html/error405.html");
                document.cookie = "JSESSIONID=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path=/;";
            }
        );
        $rootScope.componentesInicio();

        $timeout(function() {
            $('.slider').slider();
            $('.slider').slider('prev');
            $('.slider').slider('next');
        }, 500);
    }
}
]);
```


menú.html

```
<nav class="nav-extended">
  <div class="nav-content blue lighten-4">
    <ul class="tabs tabs-transparent">
      <li class="tab"><a ui-sref='stateBuzones'>Buzón de Tareas</a></li>
      <li class="tab"><a ng-click="altaProyecto()">Nuevo Evolutivo</a></li>
      <li class="tab"><a ui-sref='stateEvolutivos'>Consultar
Evolutivos</a></li>
      <li class="tab"><a ui-sref='stateSalaMaquinas'>Sala de
Máquinas</a></li>
      <li class="tab"><a ui-sref='stateCerrarSesion'>Cerrar Sesión</a></li>
    </ul>
  </div>
</nav>

<div class="" ng-init="init()">

<div class="slider">
  <ul class="slides">
    <li>
      <div class="caption center-align">
        <h3>Bienvenido a CofaBPM</h3>
        <h5 class="light grey-text text-lighten-3">Una herramienta que te ayudará a llevar tu trabajo de Factoría de Software al siguiente nivel</h5>
        <h1 class="light grey-text text-lighten-3">
<i class="material-icons hero">flight_takeoff</i>
      </h1>
    </div>
    </li>
    <li>
      <div class="caption left-align">
        <h3>Buzón de Tareas</h3>
        <h5 class="light grey-text text-lighten-3">Quiero ver mis tareas para hoy... y también mi trabajo terminado</h5>
        <h1 class="light grey-text text-lighten-3">
<i class="material-icons hero">done_all</i>
      </h1>
    </div>
    </li>
    <li>
      <div class="caption right-align">
        <h3>Nuevo Evolutivo</h3>
        <h5 class="light grey-text text-lighten-3">¿Ha llegado un
nuevo evolutivo al Factoría? ¡Abrelo y que surja la magia del BPM!</h5>
        <h1 class="light grey-text text-lighten-3">
<i class="material-icons hero">fiber_new</i>

```

```
</h1>
  </div>
</li>
</li>
  <div class="caption center-align">
    <h3>Consultar evolutivos</h3>
    <h5 class="light grey-text text-lighten-3">Oye... ¿El proyecto
A?, ¿en qué estado se encuentra? ¿Y el B?, ¿lo hemos terminado ya?</h5>
    <h1 class="light grey-text text-lighten-3">
<i class="material-icons hero">map</i>
</h1>
  </div>
</li>
</li>
  <div class="caption left-align">
    <h3>Sala de Máquinas</h3>
    <h5 class="light grey-text text-lighten-3">¿El barco va bien?
¿El motor responde? ¡Más madera en las calderas! Esto navega
rápido.</h5>
    <h1 class="light grey-text text-lighten-3">
<i class="material-icons hero">whatshot</i>
</h1>
  </div>
</li>
</ul>
</div>
</div>
```

controller3.js

```
'use strict';

angular.module('cofabpm.controllers')
  .controller('Controller3', ['$scope', '$rootScope', '$state', '$stateParams', '$timeout', 'cofabpmService',
    function($scope, $rootScope, $state, $stateParams, $timeout, cofabpmService) {

      $scope.init = function() {
        $rootScope.veficarEntradaCorrecta();

        $rootScope.componentesInicio();

        var tareasPendientes_IN = {'usuario': $rootScope.usuario, 'grupo': $rootScope.grupo}
        cofabpmService.tareasPendientes(tareasPendientes_IN)
          .then(
            function(res) {
              $scope.abiertas = res.data;
            },
            function(reason) {
              $rootScope.mostrarError('Error al recuperar las Tareas Pendientes');
            }
          );

        var tareasCerradas_IN = {'usuario': $rootScope.usuario, 'grupo': $rootScope.grupo}
        cofabpmService.tareasCerradas(tareasCerradas_IN)
          .then(
            function(res) {
              $scope.cerradas = res.data;
            },
            function(reason) {
              $scope.motorErroneo = true;
              $rootScope.mostrarError('Error al recuperar las Tareas Realizadas');
            }
          );
      }

      $scope.verPendientes = function() {
        $('#tareasPendientes').removeClass('hide')
        $('#tareasTerminadas').addClass('hide')
        $('#buzonPendiente').addClass('active');
        $('#buzonTerminada').removeClass('active');
      }

      $scope.verTerminadas = function() {
```



```
$("#tareasTerminadas").removeClass('hide')
$("#tareasPendientes").addClass('hide')
$("#buzonPendiente").removeClass('active');
$("#buzonTerminada").addClass('active');
}

$scope.irFormulario = function(tarea) {
    $stateParams.registro = angular.toJson(tarea);
    $state.go('stateFormulario', $stateParams);
}
}
});
```

buzones.html

```
nav class="nav-extended">
  <div class="nav-content blue lighten-4">
    <ul class="tabs tabs-transparent">
      <li class="tab active"><a ui-sref='stateBuzones'>Buzón de Tareas</a></li>
      <li class="tab"><a ng-click="altaProyecto()">Nuevo Evolutivo</a></li>
      <li class="tab"><a ui-sref='stateEvolutivos'>Consultar Evolutivos</a></li>
      <li class="tab"><a ui-sref='stateSalaMaquinas'>Sala de Máquinas</a></li>
      <li class="tab"><a ui-sref='stateCerrarSesion'>Cerrar Sesión</a></li>
    </ul>
  </div>
</nav>

<div class="container aireArriba" ng-init="init()">

  <h3>Consulta de Buzones </h3>

  <div class="row">
    <div class="col m3">
      <div class="collection">
        <a id="buzonPendiente" ng-click="verPendientes()" class="collection-item waves-effect waves-light"><span class="new
badge">{{abiertas.length}}</span>Tareas Pendientes</a>
        <a id="buzonTerminada" ng-click="verTerminadas()" class="collection-item waves-effect waves-light"><span
class="badge">{{cerradas.length}}</span>Terminadas</a>
      </div>
    </div>

    <div class="col m9 hide" id="tareasPendientes">
      <ul class="collection">
        <a ng-repeat = "abierta in abiertas" class="collection-item avatar waves-effect waves-light" ng-click="irFormulario(abierta)">
          <i class="material-icons circle">event_note</i>
          <span class="title">{{abierta.evolutivo}}</span>
          <p>{{abierta.nombre}}</p>
          <p>{{abierta.descripcion}}</p>
          <p class="secondary-content">{{abierta.fecha}}</p>
        </a>
      </ul>
    </div>

    <div class="col m9 hide" id="tareasTerminadas">
      <ul class="collection">
        <a ng-repeat = "cerrada in cerradas" class="collection-item avatar waves-effect waves-light">
          <i class="material-icons circle">event_note</i>
          <span class="title">{{cerrada.evolutivo}}</span>
          <p>{{cerrada.nombre}}</p>
        </a>
      </ul>
    </div>
  </div>
```



```
<p class="secondary-content">{{cerrada.fecha}}</p>  
</a>  
</ul>  
</div>  
</div>  
</div>
```

controller7.js

```
use strict';

angular.module('cofabpm.controllers')
.controller('Controller7', ['$scope', '$rootScope', '$state', '$stateParams', '$timeout', 'cofabpmService',
function($scope, $rootScope, $state, $stateParams, $timeout, cofabpmService) {

    $scope.init = function() {

        $rootScope.veficarEntradaCorrecta();

        $rootScope.componentesInicio();

        if ($stateParams.registro) {
            $scope.tareaEntrada = angular.fromJson($stateParams.registro);

            $scope.nombreTarea = $scope.tareaEntrada.nombre;
            $scope.id = $scope.tareaEntrada.id;

            $stateParams.registro = null;
        }

        $scope.formulario = {};

        $scope.mensajesFormulario = {};
        $scope.errorFormulario = false;

        $scope.inicializarCamposFormulario();

        $scope.pintarFormulario();

    };

    $scope.inicializarCamposFormulario = function() {
        switch ($scope.tareaEntrada.idActividad) {
            case 'UT1_Alta_Proyecto':
                {
                    $scope.formulario.fechaInicio = new Date().toISOString().slice(0, 10);
                    $scope.formulario.fechaFin = "";
                    return;
                    break;
                }
            case 'UT2_Entrega_DF':
```

```
{
    $scope.formulario.fechaEntrega = "";
    return;
    break;
}
case 'UT3_Validacion_DF':
case 'UT5_Validacion_Activos':
case 'UT7_Validacion_Pruebas_Unitarias':
case 'UT9_Validacion_Pruebas_Funcionales':
case 'UT10_Pruebas_Integracion':
{
    $scope.formulario.conforme = false;
    return;
    break;
}
case 'UT11_Aprobacion_Despliegue':
{
    $scope.formulario.aprobadaImplantacion = false;
    $scope.formulario.fechaImplantacion = "";
    return;
    break;
}

case 'UT4_CodificacionActivos':
case 'UT6_Pruebas_Unitarias_Activos':
case 'UT8_Pruebas_Funcionales':
default:
}
};

$scope.pintarFormulario = function() {
    $timeout(function() {
        Materialize.updateTextFields();
    }, 10);

    if ($scope.formulario.fechaInicio) {
        var $input = $('#fecha_inicio_input').pickadate()
        // Use the picker object directly.
        var picker = $input.pickadate('picker')
        picker.set('select', $scope.formulario.fechaInicio, {
            format: 'yyyy-mm-dd'
        })
    }
}

if ($scope.formulario.fechaFin) {
```



```
var $input = $('#fecha_fin_input').pickadate()

// Use the picker object directly.
var picker = $input.pickadate('picker')
picker.set('select', $scope.formulario.fechaFin, {
  format: 'yyyy-mm-dd'
})

}

$('#fecha_inicio_input').removeClass("invalid");
$('#fecha_fin_input').removeClass("invalid");
}
```

```
$scope.soyFormulario = function(quienSoy) {
  if (quienSoy === $scope.tareaEntrada.idActividad) {
    return true;
  } else {
    return false;
  }
};
```

```
$scope.tablaOpciones = [{
  'campo': 'codigo',
  'opciones': {
    'obligatorio': true,
    'minLong': 5,
    'maxLong': 5
  }
},
{
  'campo': 'nombre',
  'opciones': {
    'obligatorio': true,
    'minLong': 5,
    'maxLong': 255
  }
},
{
  'campo': 'descripcion',
  'opciones': {
    'obligatorio': false,
    'minLong': 5,
    'maxLong': 500
  }
},
{
```

```
'campo': 'ruta',
'opciones': {
  'obligatorio': false,
  'minLong': 5,
  'maxLong': 100
},
{
'campo': 'fechalnicio',
'opciones': {
  'obligatorio': false,
  'minLong': 10,
  'maxLong': 10
},
},
{
'campo': 'fechaFin',
'opciones': {
  'obligatorio': false,
  'minLong': 9,
  'maxLong': 5
},
},
{
'campo': 'fechaEntrega',
'opciones': {
  'obligatorio': false,
  'minLong': 9,
  'maxLong': 5
},
},
{
'campo': 'entregables',
'opciones': {
  'obligatorio': true,
  'minLong': 1,
  'maxLong': 3,
  'number': true
},
},
{
'campo': 'activosCodificados',
'opciones': {
  'obligatorio': true,
  'minLong': 1,
  'maxLong': 3,
```

```
'number': true
}
},
{
'campo': 'testRealizados',
'opciones': {
'obligatorio': true,
'minLong': 1,
'maxLong': 3,
'number': true
}
},
{
'campo': 'testErroneos',
'opciones': {
'obligatorio': true,
'minLong': 1,
'maxLong': 3,
'number': true
}
},
{
'campo': 'casosErroneos',
'opciones': {
'obligatorio': true,
'minLong': 1,
'maxLong': 3,
'number': true
}
},
{
'campo': 'casosUso',
'opciones': {
'obligatorio': true,
'minLong': 1,
'maxLong': 3,
'number': true
}
},
{
'campo': 'coberturaCubierta',
'opciones': {
'obligatorio': true,
'minLong': 1,
'maxLong': 3,
'number': true
}
```

```
    }  
  }  
];  
  
$scope.buscarOpciones = function(campo) {  
  var indice = 0;  
  
  while (indice < $scope.tablaOpciones.length) {  
    if ($scope.tablaOpciones[indice].campo === campo) {  
      return $scope.tablaOpciones[indice].opciones;  
    }  
    indice++;  
  }  
  return {};  
};  
  
$scope.validarCampoFormulario = function(input, campo) {  
  
  var opciones = $scope.buscarOpciones(campo);  
  var error = false;  
  var mensaje = "";  
  
  /* Se inicializa la validacion del campo */  
  eval("#" + input.target.id + ").removeClass('invalid');"  
  eval("$scope.mensajesFormulario." + campo + "= null");  
  
  if (input.target.value === "") {  
    if (opciones.obligatorio) {  
      error = true;  
      mensaje = 'Campo Obligatorio';  
    }  
  } else {  
    if (!error && (opciones.minLong) && (input.target.value.length < opciones.minLong)) {  
      error = true;  
      mensaje = 'Longitud inferior a ' + opciones.minLong + ' caracteres';  
    }  
    if (!error && (opciones.maxLong) && (input.target.value.length > opciones.maxLong)) {  
      error = true;  
      mensaje = 'Longitud superior a ' + opciones.maxLong + ' caracteres';  
    }  
    if (!error && (opciones.number) && (isNaN(input.target.value))) {  
      error = true;  
      mensaje = 'No es un número válido';  
    }  
  }  
};  
}
```

```
if (error) {
    eval("#" + input.target.id + ").addClass('invalid');");
    eval("$scope.mensajesFormulario." + campo + "=" + "'" + mensaje + "'");
    eval("#" + campo + " label').addClass('active');");
}

$scope.hayErrorFormulario();
};

$scope.hayErrorFormulario = function() {
    if ($('#input.invalid').length > 0) {
        $scope.errorFormulario = true;
    } else {
        $scope.errorFormulario = false;
    }
}

$scope.validarFechaImplantacion = function() {
    var $input = $('#fechaImplantacion_input').pickadate();
    var picker = $input.pickadate('picker');
    $scope.formulario.fechaImplantacion = picker.get('select', 'yyyy-mm-dd');
    $('#fechaImplantacion_input').removeClass("invalid");
    $('#fechaImplantacion_input').addClass("valid");
};

$scope.validarFechaEntrega = function() {
    var $input = $('#fecha_entrega_input').pickadate();
    var picker = $input.pickadate('picker');
    $scope.formulario.fechaEntrega = picker.get('select', 'yyyy-mm-dd');
    $('#fecha_entrega_input').removeClass("invalid");
    $('#fecha_entrega_input').addClass("valid");
};

$scope.validarFechaInicio = function() {
    var $input = $('#fecha_inicio_input').pickadate();
    var picker = $input.pickadate('picker');
    $scope.formulario.fechaInicio = picker.get('select', 'yyyy-mm-dd');
    $scope.validarRangoFechas();
};

$scope.validarFechaFin = function() {
    var $input = $('#fecha_fin_input').pickadate();
    var picker = $input.pickadate('picker');
    $scope.formulario.fechaFin = picker.get('select', 'yyyy-mm-dd');
```

```
$timeout(function() {
    $scope.validarRangoFechas();
}, 100);
};

$scope.validarRangoFechas = function() {
    $scope.mensajesFormulario.fechaInicio = "";
    $('#fecha_inicio_input').removeClass("invalid");
    $('#fecha_inicio_input').addClass("valid");
    $('#fecha_fin_input').removeClass("invalid");
    $('#fecha_fin_input').addClass("valid");
    if ($scope.formulario.fechaInicio && $scope.formulario.fechaFin) {
        if (($scope.formulario.fechaInicio) > ($scope.formulario.fechaFin)) {
            $('#fecha_inicio_input').addClass("invalid");
            $scope.mensajesFormulario.fechaInicio = "Rango de fechas inválido";
        }
        $scope.hayErrorFormulario();
    }
};

$scope.abrirEvolutivo = function() {

    $scope.formulario.grupo = $scope.grupo;
    $scope.formulario.usuario = $scope.usuario;

    cofabpmService.abrirEvolutivo($scope.formulario)
        .then(
            function(res) {
                $scope.mostrarResultado('Evolutivo dada de alta con éxito', 'check');
            },
            function(reason) {
                $scope.mostrarResultado('ERROR. El evolutivo no se ha podido dar de alta', 'close');
            }
        );
};

$scope.salvarRegistro = function() {

    $scope.formulario.grupo = $rootScope.grupo;
    $scope.formulario.usuario = $rootScope.usuario;
    $scope.formulario.id = $scope.id;

    cofabpmService.completarTarea($scope.formulario)
        .then(
            function(res) {
```

```
    $scope.mostrarResultado('Tarea completada con éxito', 'check');
  },
  function(reason) {
    $scope.mostrarResultado('ERROR. La tarea no se ha podido completar', 'close');
  }
);
};
```

```
$scope.mostrarResultado = function(mensaje, icono) {
  $('#modalGraba').removeClass('green');
  $('#modalGraba').removeClass('red');
  $scope.mensajeResultado = mensaje;
  $scope.icono = icono;
  if (icono === 'check') {
    $('#modalGraba').addClass('green');
  } else {
    $('#modalGraba').addClass('red');
  }

  $('#modalGraba').modal('open');
  $timeout(function() {
    $('#modalGraba').modal('close');
    $scope.volverAtras();
  }, 2000);
};
```

// Volver Atras

```
$scope.volverAtras = function() {
  $state.go("stateEvolutivos");
};
```

// Traducir el código de Tarea

```
$scope.traducirCodigo = function(codigo) {

  switch (codigo) {
    case 'UT1_Alta_Proyecto':
      return 'Alta Proyecto';
      break;

    case 'UT2_Entrega_DF':
      return 'Entrega Diseño Físico';
      break;

    case 'UT3_Validacion_DF':
      return 'Validación Diseño';
      break;
```

```
case 'UT4_CodificacionActivos':  
    return 'Codificación de Activos';  
    break;  
  
case 'UT5_Validacion_Activos':  
    return 'Validación Activos';  
    break;  
  
case 'UT6_Pruebas_Unitarias_Activos':  
    return 'Pruebas Unitarias de Activos';  
    break;  
  
case 'UT7_Validacion_Pruebas_Unitarias':  
    return 'Validación Pruebas Unitarias';  
    break;  
  
case 'UT8_Pruebas_Funcionales':  
    return 'Pruebas Funcionales';  
    break;  
  
case 'UT9_Validacion_Pruebas_Funcionales':  
    return 'Validación Pruebas Funcionales';  
    break;  
  
case 'UT10_Pruebas_Integracion':  
    return 'Pruebas de Integración';  
    break;  
  
case 'UT11_Aprobacion_Despliegue':  
    return 'Aprobación y Planificación Despliegue';  
    break;  
default:  
  
    }  
    }  
  
    }  
);
```


formulario.html

```
<nav class="nav-extended">
  <div class="nav-content blue lighten-4">
    <ul class="tabs tabs-transparent">
      <li class="tab"><a ui-sref='stateBuzones'>Buzón de Tareas</a></li>
      <li class="tab active"><a ng-click="#">Nuevo Evolutivo</a></li>
      <li class="tab"><a ui-sref='stateEvolutivos'>Consultar Evolutivos</a></li>
      <li class="tab"><a ui-sref='stateSalaMaquinas'>Sala de Máquinas</a></li>
      <li class="tab"><a ui-sref='stateCerrarSesion'>Cerrar Sesión</a></li>
    </ul>
  </div>
</nav>

<div class="container aireArriba" ng-init="init()">
  <h3>Formulario Tarea de Usuario: {{nombreTarea}}</h3>

  <div class="row">&nbsp;</div>

  <div class="row" class="aireArriba">
    <div class="col s12 m12 l12 xl10 offset-xl1">
      <!-- FORMULARIO 1 Alta de Nuevo Evolutivo -->
      <form name="form1" id="form1" class="grey lighten-5 z-depth-3" ng-show="soyFormulario('UT1_Alta_Proyecto')">
        <div class="row">
          <div class="col m2 offset-m1 input-field" id="codigo">
            <input id="codigo_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'codigo')" ng-model="formulario.codigo">
            <label for="codigo" data-error="{{mensajesFormulario.codigo}}" data-success="correcto">Código<sup>*</sup></label>
          </div>
          <div class="col m8 input-field" id="nombre">
            <input id="nombre_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'nombre')" ng-model="formulario.nombre">
            <label for="nombre" data-error="{{mensajesFormulario.nombre}}" data-success="correcto">Nombre del Evolutivo<sup>*</sup></label>
          </div>
        </div>
        <div class="row">
          <div class="col m10 offset-m1 input-field" id="descripcion">
            <textarea id="cp_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'descripcion')" ng-
model="formulario.descripcion"></textarea>
            <label for="descripcion" data-error="{{mensajesFormulario.descripcion}}" data-success="correcto">Descripción del Evolutivo</label>
          </div>
        </div>
        <div class="row">
          <div class="col m3 offset-m1 input-field" id="fecha_inicio">
            <input id="fecha_inicio_input" type="date" ng-change="validarFechaInicio()" class="datepicker" ng-value="formulario.fechaInicio" ng-
model="formulario.fechaInicio">

```

```

    <label data-error="{{mensajesFormulario.fechalnicio}}" data-success="correcto">Fecha Inicio</label>
</div>
<div class="col m3 input-field" id="fecha_fin">
    <input id="fecha_fin_input" type="date" ng-change="validarFechaFin()" class="datepicker" ng-value="formulario.fechaFin" ng-
model="formulario.fechaFin">
    <label for="fecha_fin" data-error="{{mensajesFormulario.fechaFin}}" data-error="erróneo" data-success="correcto">Fecha Fin</label>
</div>

<div class="row">
<div class="col m3">
    <label class="red-text">{{mensajesFormulario.fechalnicio}}</label>
</div>
</div>

<div class="row">
<div class="col s2 offset-s5 m2 offset-m10">
<div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom:-10px left aligned;">
    <a class="btn-floating waves-effect waves-light tooltipped btn-large red " data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
        <i class="large material-icons">menu</i>
    </a>
    <ul>
        <li id="btn_save"><button ng-disabled="errorFormulario || !formulario.nombre" class="btn-floating tooltipped waves-effect waves-light green"
type="submit" data-position="bottom" data-delay="100" data-tooltip="Grabar" ng-click="abrirEvolutivo()"><i class="material-icons">save</i></button></li>
        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-
click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>
    </ul>
</div>
</div>
</div>
</form>

<!-- FORMULARIO 2 Entrega de Diseño Físico -->
<form name="form2" id="form2" class="grey lighten-5 z-depth-3" ng-show="soyFormulario(UT2_Entrega_DF)">

<div class="row">
<div class="col m10 offset-m1 input-field" id="descripcion">
    <textarea id="descripcion_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'descripcion)" ng-
model="formulario.descripcion"></textarea>
    <label for="descripcion" data-error="{{mensajesFormulario.descripcion}}" data-success="correcto">Descripción del Diseño Físico</label>
</div>
</div>

<div class="row">
<div class="col m3 offset-m1 input-field" id="entregables">

```

```

<input id="entregables_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'entregables')" ng-
model="formulario.entregables"></input>
<label for="entregables" data-error="{{mensajesFormulario.entregables}}" data-success="correcto">Número de Entregables</label>
</div>

<div class="col m3 offset-m1 input-field" id="fecha_entrega">
  <input id="fecha_entrega_input" type="date" ng-change="validarFechaEntrega()" class="datepicker" ng-value="formulario.fechaEntrega" ng-
model="formulario.fechaEntrega">
  <label data-error="{{mensajesFormulario.fechaEntrega}}" data-success="correcto">Fecha Entrega</label>
</div>
</div>

<div class="row">
  <div class="col m10 offset-m1 input-field" id="ruta">
    <input id="ruta_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'ruta')" ng-model="formulario.ruta"></input>
    <label for="entregables" data-error="{{mensajesFormulario.ruta}}" data-success="correcto">Ruta de Entregables</label>
  </div>
</div>

<div class="row">
  <div class="col s2 offset-s5 m2 offset-m10">
    <div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom:-10px left aligned;">
      <a class="btn-floating waves-effect waves-light tooltipped btn-large red " data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
        <i class="large material-icons">menu</i>
      </a>
      <ul>
        <li id="btn_save"><button ng-disabled="errorFormulario || !formulario.entregables" class="btn-floating tooltipped waves-effect waves-light green"
type="submit" data-position="bottom" data-delay="100" data-tooltip="Grabar" ng-click="salvarRegistro()"><i class="material-icons">save</i></button></li>
        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-
click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>
      </ul>
    </div>
  </div>
</div>
</form>

<!-- FORMULARIO 3 Validacion de Diseño -->
<!-- FORMULARIO 5 Validacion de Codifcacion -->
<!-- FORMULARIO 7 Validacion de Pruebas Unitarias -->
<!-- FORMULARIO 9 Validacion de Pruebas Funcionales -->
<!-- FORMULARIO 10 Pruebas de Integración -->
<form name="form3" id="form3" class="grey lighten-5 z-depth-3" ng-show="soyFormulario(UT3_Validacion_DF) ||
soyFormulario(UT5_Validacion_Activos) || soyFormulario(UT7_Validacion_Pruebas_Unitarias) || soyFormulario(UT9_Validacion_Pruebas_Funcionales) ||
soyFormulario(UT10_Pruebas_Integracion)">

```

```

<div class="row">
  <div class="col m6 offset-m3 input-field" id="conformidad">
    <!-- Switch -->
    <div class="switch">
      <label>
        No Conforme
        <input type="checkbox" ng-model="formulario.conforme" >
        <span class="lever"></span>
      </label>
      <label>
        Si Conforme
      </label>
    </div>
  </div>
</div>
<br/>
<br/>
</div>

<div class="row">
  <div class="col m10 offset-m1 input-field" id="motivo">
    <textarea id="motivo_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'motivo')" ng-
model="formulario.motivo"></textarea>
    <label for="motivo" data-error="{{mensajesFormulario.motivo}}" data-success="correcto">Motivo</label>
  </div>
</div>

<div class="row">
  <div class="col s2 offset-s5 m2 offset-m10">
    <div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom:-10px left aligned;">
      <a class="btn-floating waves-effect waves-light tooltipped btn-large red " data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
        <i class="large material-icons">menu</i>
      </a>
      <ul>
        <li id="btn_save"><button ng-disabled="errorFormulario" class="btn-floating tooltipped waves-effect waves-light green" type="submit" data-
position="bottom" data-delay="100" data-tooltip="Grabar" ng-click="salvarRegistro()"><i class="material-icons">save</i></button></li>
        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-
click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>
      </ul>
    </div>
  </div>
</div>
</form>

<!-- FORMULARIO 4 Codificación de Activos -->
<form name="form4" id="form4" class="grey lighten-5 z-depth-3" ng-show="soyFormulario('UT4_CodificacionActivos')">

```

```

<div class="row">
  <div class="col m5 offset-m1 input-field" id="activosCodificados">
    <input id="activosCodificados_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'activosCodificados')" ng-
model="formulario.activosCodificados"></input>
    <label for="activosCodificados" data-error="{{mensajesFormulario.activosCodificados}}" data-success="correcto">Número de Activos
Codificados</label>
  </div>
</div>

<div class="row">
  <div class="col m10 offset-m1 input-field" id="ruta">
    <input id="ruta_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'ruta')" ng-model="formulario.ruta"></input>
    <label for="entregables" data-error="{{mensajesFormulario.ruta}}" data-success="correcto">Ruta de Entregables</label>
  </div>
</div>

<div class="row">
  <div class="col m10 offset-m1 input-field" id="descripcion">
    <textarea id="descripcion_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'descripcion')" ng-
model="formulario.descripcion"></textarea>
    <label for="descripcion" data-error="{{mensajesFormulario.descripcion}}" data-success="correcto">Observaciones del Evolutivo</label>
  </div>
</div>

<div class="row">
  <div class="col s2 offset-s5 m2 offset-m10">
    <div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom:-10px left aligned;">
      <a class="btn-floating waves-effect waves-light tooltipped btn-large red" data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
        <i class="large material-icons">menu</i>
      </a>
      <ul>
        <li id="btn_save"><button ng-disabled=" errorFormulario || !formulario.activosCodificados" class="btn-floating tooltipped waves-effect waves-light
green" type="submit" data-position="bottom" data-delay="100" data-tooltip="Grabar" ng-click="salvarRegistro()"><i class="material-
icons">save</i></button></li>
        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-
click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>
      </ul>
    </div>
  </div>
</div>
</form>

<!-- FORMULARIO 6 Pruebas Unitarias -->
<form name="form6" id="form6" class="grey lighten-5 z-depth-3" ng-show="soyFormulario('UT6_Pruebas_Unitarias_Activos')>

```

```

<div class="row">
  <div class="col m3 offset-m1 input-field" id="testRealizados">
    <input id="testRealizados_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'testRealizados')" ng-
model="formulario.testRealizados"></input>
    <label for="testRealizados" data-error="{{mensajesFormulario.testRealizados}}" data-success="correcto">Test Realizados</label>
  </div>
  <div class="col m3 input-field" id="erroneos">
    <input id="testErroneos_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'testErroneos')" ng-
model="formulario.testErroneos"></input>
    <label for="testErroneos" data-error="{{mensajesFormulario.testErroneos}}" data-success="correcto">Test Erroneos</label>
  </div>
  <div class="col m3 input-field" id="cobertura">
    <input id="coberturaCubierta_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'coberturaCubierta')" ng-
model="formulario.coberturaCubierta"></input>
    <label for="coberturaCubierta" data-error="{{mensajesFormulario.coberturaCubierta}}" data-success="correcto">Cobertura Cubierta (%)</label>
  </div>
</div>

<div class="row">
  <div class="col m10 offset-m1 input-field" id="ruta">
    <input id="ruta_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'ruta')" ng-model="formulario.ruta"></input>
    <label for="ruta" data-error="{{mensajesFormulario.ruta}}" data-success="correcto">Ruta de Entregables</label>
  </div>
</div>

<div class="row">
  <div class="col m10 offset-m1 input-field" id="descripcion">
    <textarea id="descripcion_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'descripcion')" ng-
model="formulario.descripcion"></textarea>
    <label for="descripcion" data-error="{{mensajesFormulario.descripcion}}" data-success="correcto">Observaciones de las Pruebas</label>
  </div>
</div>

<div class="row">
  <div class="col s2 offset-s5 m2 offset-m10">
    <div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom:-10px left aligned;">
      <a class="btn-floating waves-effect waves-light tooltipped btn-large red " data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
        <i class="large material-icons">menu</i>
      </a>
      <ul>
        <li id="btn_save"><button ng-disabled="errorFormulario || !formulario.testRealizados || !formulario.testErroneos || !formulario.coberturaCubierta"
class="btn-floating tooltipped waves-effect waves-light green" type="submit" data-position="bottom" data-delay="100" data-tooltip="Grabar" ng-
click="salvarRegistro()"><i class="material-icons">save</i></button></li>
        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-
click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>
      </ul>
    </div>
  </div>
</div>

```

```

    </ul>
  </div>
</div>
</div>
</form>

<!-- FORMULARIO 8 Pruebas Funcionales -->
<form name="form8" id="form8" class="grey lighten-5 z-depth-3" ng-show="soyFormulario('UT8_Pruebas_Funcionales')">

  <div class="row">
    <div class="col m4 offset-m1 input-field" id="casosUso">
      <input id="casosUso_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'casosUso')" ng-
model="formulario.casosUso"></input>
      <label for="casosUso" data-error="{{mensajesFormulario.casosUso}}" data-success="correcto">Casos de Uso</label>
    </div>
    <div class="col m4 input-field" id="casosErroneos">
      <input id="casosErroneos_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'casosErroneos')" ng-
model="formulario.casosErroneos"></input>
      <label for="casosErroneos" data-error="{{mensajesFormulario.casosErroneos}}" data-success="correcto">Casos de Uso Erróneos</label>
    </div>
  </div>

  <div class="row">
    <div class="col m10 offset-m1 input-field" id="ruta">
      <input id="ruta_input" type="text" class="validate" ng-blur="validarCampoFormulario($event,'ruta')" ng-model="formulario.ruta"></input>
      <label for="ruta" data-error="{{mensajesFormulario.ruta}}" data-success="correcto">Ruta de Evidencias</label>
    </div>
  </div>

  <div class="row">
    <div class="col m10 offset-m1 input-field" id="descripcion">
      <textarea id="descripcion_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'descripcion')" ng-
model="formulario.descripcion"></textarea>
      <label for="descripcion" data-error="{{mensajesFormulario.descripcion}}" data-success="correcto">Observaciones de las Pruebas
Funcionales</label>
    </div>
  </div>

  <div class="row">
    <div class="col s2 offset-s5 m2 offset-m10">
      <div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom: -10px left aligned;">
        <a class="btn-floating waves-effect waves-light tooltipped btn-large red " data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
          <i class="large material-icons">menu</i>
        </a>
      </div>
    </div>
  </div>

```

```

        <li id="btn_save"><button ng-disabled=" errorFormulario || !formulario.casosErroneos || !formulario.casosUso" class="btn-floating tooltipped waves-effect waves-light green" type="submit" data-position="bottom" data-delay="100" data-tooltip="Grabar" ng-click="salvarRegistro()"><i class="material-icons">save</i></button></li>

        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>

    </ul>
</div>
</div>
</div>
</div>
</form>

<!-- FORMULARIO 11 Aprobación Planificación y Despligue -->
<form name="form3" id="form3" class="grey lighten-5 z-depth-3" ng-show="soyFormulario(UT11_Aprobacion_Despliegue)">

<div class="row">
<div class="col m8 offset-m2 input-field" id="conformidad">
<!-- Switch -->
<div class="switch">
<label>
Aprobada la Implantación
<input type="checkbox" ng-model="formulario.aprobadaImplantacion" >
<span class="lever"></span>
Revocada la Implantación
</label>
</div>
</div>
<br/>
<br/>
</div>

<div class="row" >
<div class="col m10 offset-m1 input-field" id="descripcion">
<textarea id="descripcion_input" type="text" class="materialize-textarea validate" ng-blur="validarCampoFormulario($event,'descripcion')" ng-model="formulario.descripcion"></textarea>
<label for="descripcion" data-error="{{mensajesFormulario.descripcion}}" data-success="correcto">Observaciones</label>
</div>
</div>

<div class="row">
<div class="col m3 offset-m1 input-field" id="fecha_implantacion">
<input id="fechaImplantacion_input" type="date" ng-change="validarFechaImplantacion()" class="datepicker" ng-value="formulario.fechaImplantacion" ng-model="formulario.fechaImplantacion">
<label data-error="{{mensajesFormulario.fechaImplantacion}}" data-success="correcto">Fecha Implantación</label>
</div>
</div>

```



```

<div class="row">
  <div class="col s2 offset-s5 m2 offset-m10">
    <div class="fixed-action-btn horizontal click-to-toggle" style="position: relative; display: inline-block; right: 30px; bottom:-10px left aligned;">
      <a class="btn-floating waves-effect waves-light tooltipped btn-large red " data-position="bottom" data-delay="100" data-tooltip="Opciones de
Menú">
        <i class="large material-icons">menu</i>
      </a>
      <ul>
        <li id="btn_save"><button ng-disabled="errorFormulario" class="btn-floating tooltipped waves-effect waves-light green" type="submit" data-
position="bottom" data-delay="100" data-tooltip="Grabar" ng-click="salvarRegistro()"><i class="material-icons">save</i></button></li>
        <li><a class="btn-floating waves-effect waves-light tooltipped blue" data-position="bottom" data-delay="100" data-tooltip="Salir" ng-
click="volverAtras()"><i class="material-icons">exit_to_app</i></a></li>
      </ul>
    </div>
  </div>
</div>
</div>
</form>
</div>
</div>
</div>
<!-- Modal Grabacion -->
<div id="modalGraba" class="modal green">
  <div class="modal-content ">
    <br>
    <br>
    <h1 class="white-text center"><i class="material-icons large">{{icono}}</i></h1>
    <h5 class="white-text center">{{mensajeResultado}}</h5>
  </div>
</div>
</div>

```

controller4.js

```
'use strict';

angular.module('cofabpm.controllers')
.controller('Controller4', ['$scope', '$rootScope', '$state', '$stateParams', '$timeout', 'cofabpmService', 'DTOptionsBuilder', 'DTColumnDefBuilder',
function($scope, $rootScope, $state, $stateParams, $timeout, cofabpmService, DTOptionsBuilder, DTColumnDefBuilder) {

$scope.init = function() {
    $rootScope.veficarEntradaCorrecta();

    $rootScope.componentesInicio();

    $scope.dtInstance = {};

    // DataTables opciones configurables
    $scope.dtColumnDefs = [DTColumnDefBuilder.newColumnDef([3, 4]).withOption("type", "date-eu")];

    $scope.dtOptions = DTOptionsBuilder.newOptions()
        .withDisplayLength(10)
        .withPaginationType('full_numbers')
        .withLanguage({
            sUrl: "json/Spanish.json"
        })
        .withOption('order', [0, 'asc']);

    cofabpmService.listarEvolutivos()
        .then(
            function(res) {
                $scope.evolutivos = res.data;
                $timeout(function() {
                    $('select').addClass("browser-default");
                    $('select').material_select();
                    $timeout(function() {
                        $('#carga').addClass('hide');
                        $('#tabla').removeClass('hide');
                    }, 500);
                }, 500);
            },
            function(reason) {
                $scope.motorErroneo = true;
                $rootScope.mostrarError('Error al recuperar los Evolutivos');
            }
        )
    )
}
```

```
};

$scope.seleccionarFila = function(evolutivo) {

    if (evolutivo.fechaInicio) {
        evolutivo.fechaInicio = moment(evolutivo.fechaInicio).format("YYYY-MM-DD");
    }
    if (evolutivo.fechaFin) {
        evolutivo.fechaFin = moment(evolutivo.fechaFin).format("YYYY-MM-DD");
    }

    $stateParams.registro = angular.toJson(evolutivo);
    $state.go('stateDetalleEvolutivo', $stateParams);

};

$scope.completarTarea = function(tarea) {
    $stateParams.registro = angular.toJson(tarea);
    $state.go('stateFormulario', $stateParams);
}

}

});
```

evolutivos.html

```
<div ng-init="init()">
  <nav class="nav-extended">
    <div class="nav-content blue lighten-4">
      <ul class="tabs tabs-transparent">
        <li class="tab"><a ui-sref='stateBuzones'>Buzón de Tareas</a></li>
        <li class="tab"><a ng-click="altaProyecto()">Nuevo Evolutivo</a></li>
        <li class="tab active"><a ui-sref='stateEvolutivos'>Consultar Evolutivos</a></li>
        <li class="tab"><a ui-sref='stateSalaMaquinas'>Sala de Máquinas</a></li>
        <li class="tab"><a ui-sref='stateCerrarSesion'>Cerrar Sesión</a></li>
      </ul>
    </div>
  </nav>

  <div class="container aireArrriba" ng-init="init()">
    <h3>Consulta de Evolutivos </h3>
    <!-- Loader -->
    <div id="carga" class="row ">
      <div class="col m1 offset-m5 ">
        <div class="center-align preloader-wrapper big active">
          <div class="spinner-layer spinner-blue-only">
            <div class="circle-clipper left">
              <div class="circle"></div>
            </div>
            <div class="gap-patch">
              <div class="circle"></div>
            </div>
            <div class="circle-clipper right">
              <div class="circle"></div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <!-- Tabla de Resultados -->
    <div id="tabla" class="row hide">
      <div class="col m12">
        <table datatable="ng" dt-options="dtOptions" dt-column-defs="dtColumnDefs" dt-instance="dtInstance" class="striped highlight responsive-table"
id="resultadoBusqueda">
          <thead>
            <tr>
              <th>Código</th>
              <th>Nombre del Evolutivo</th>
              <th>Descripción</th>
              <th>Fecha Inicio</th>
            </tr>
          </thead>
        </table>
      </div>
    </div>
  </div>
```

```
<th>Fecha Fin</th>
</tr>
</thead>
<tbody>
<tr ng-repeat="evolutivo in evolutivos" ng-click="seleccionarFila(evolutivo)" class="tablaresultado">
<td>{{evolutivo.codigo}}</td>
<td>{{evolutivo.nombre}}</td>
<td>{{evolutivo.descripcion}}</td>
<td>{{evolutivo.fechaInicio | date: 'dd/MM/yyyy'}}</td>
<td>{{evolutivo.fechaFin | date: 'dd/MM/yyyy'}}</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>

<!-- Modal Grabacion -->
<div id="modalGraba" class="modal green">
<div class="modal-content ">
<br>
<br>
<h1 class="white-text center"><i class="material-icons large">{{icono}}</i></h1>
<h5 class="white-text center">{{mensajeResultado}}</h5>
</div>
</div>
</div>
```

controller6.js

```
'use strict';

angular.module('cofabpm.controllers')
.controller('Controller6', ['$scope', '$rootScope', '$state', '$stateParams', '$timeout', 'cofabpmService',
function($scope, $rootScope, $state, $stateParams, $timeout, cofabpmService) {

    $scope.init = function() {

        $rootScope.veficarEntradaCorrecta();

        $rootScope.componentesInicio();

        if ($stateParams.registro) {
            $scope.entrada = angular.fromJson($stateParams.registro);

            var estadoEvolutivo_IN = {
                "idProceso": $scope.entrada.idProceso
            };

            cofabpmService.estadoEvolutivo(estadoEvolutivo_IN)
                .then(
                    function(res) {
                        $scope.tareas = res.data;
                    },
                    function(reason) {
                        $rootScope.mostrarError('Error al recuperar los detalles del Evolutivo');
                    }
                );
        }

        $scope.tipoDePaso = function(activityType) {
            switch (activityType) {
                case "startEvent":
                    return "panorama_fish_eye";
                case "userTask":
                    return "more_vert";
                case "serviceTask":
                    return "more_vert";
                case "endEvent":
                    return "lens";
                default:
            }
        };
    }
}]);
```

```
$scope.tipoDeTarea = function(activityType) {
  switch (activityType) {
    case "startEvent":
      return "layers";
    case "userTask":
      return "directions_walk";
    case "serviceTask":
      return "mail";
    case "endEvent":
      return "layers_clear";
    default:
  }
};

$scope.nombreTipodeTarea = function(activityType) {
  switch (activityType) {
    case "startEvent":
      return "Comienzo";
    case "userTask":
      return "Tarea de Usuario";
    case "serviceTask":
      return "Tarea de Servicio de Email";
    case "endEvent":
      return "Final";
    default:
  }
};

$scope.esUserTask = function(activityType) {
  if (activityType == 'userTask') {
    return true;
  } else {
    return false;
  }
};

$scope.convertirFecha = function(fecha, indice) {
  if (fecha) {
    return new Date(fecha).toLocaleString();
  } else {
    if (indice) {
      $("##titulo" + indice).addClass("orange");
      return "Tarea en Ejecución";
    }
  }
}
```



```
}  
};  
  
$scope.convertirTiempo = function(duracion) {  
    return moment.duration(duracion,'ms').humanize({precision: 3})  
}  
  
$scope.estaAbierta = function(fecha) {  
    if (fecha) {  
        return false;  
    } else {  
        return true;  
    }  
};  
}
```


detalleevolutivo.html

```
<nav class="nav-extended">
  <div class="nav-content blue lighten-4">
    <ul class="tabs tabs-transparent">
      <li class="tab"><a ui-sref='stateBuzones'>Buzón de Tareas</a></li>
      <li class="tab"><a ng-click="altaProyecto()">Nuevo Evolutivo</a></li>
      <li class="tab active"><a ui-sref='stateEvolutivos'>Consultar Evolutivos</a></li>
      <li class="tab"><a ui-sref='stateSalaMaquinas'>Sala de Máquinas</a></li>
      <li class="tab"><a ui-sref='stateCerrarSesion'>Cerrar Sesión</a></li>
    </ul>
  </div>
</nav>

<div class="container aireArriba" ng-init="init()">
  <h3>Detalle de Evolutivo</h3>

  <div class="row">
    <div class="col s12 m12 l12 xl10 offset-xl1">
      <ul class="collapsible" data-collapsible="expandable">

        <li ng-repeat="tarea in tareas">
          <div class="collapsible-header" id="{{titulo+'$index'}}" ng-show="{{tarea.tipo!='exclusiveGateway'}}">
            <div class="col s6 m6 l9">
              <i class="material-icons">{{tipoDePaso(tarea.tipo)}}</i>{{tarea.nombre}}
            </div>
            <div class="col s6 m6 l3">
              <span>&nbsp;<i class="material-icons">timer</i> {{convertirTiempo(tarea.duracion)}}</span>
            </div>
          </div>
          <div class="collapsible-body" ng-show="{{tarea.tipo!='exclusiveGateway'}}">
            <div class="row">
              <div class="col m4">
                <div class="row center taskrow">
                  <i class="medium material-icons">{{tipoDeTarea(tarea.tipo)}}</i>
                </div>
                <div class="row center taskrow">
                  <span>&nbsp;<i class="medium material-icons">timer</i> {{convertirTiempo(tarea.duracion)}}</span>
                </div>
                <div class="row center taskrow">
                  <span>&nbsp;<i class="medium material-icons">date_range</i> {{date_range}}</span>
                </div>
              </div>
            </div>
            <div class="col m4">
              <div class="row center taskrow">
                <i class="medium material-icons">date_range</i> {{date_range}}</span>
              </div>
            </div>
          </div>
        </li>
      </ul>
    </div>
  </div>
```



```
</div>
<div class="row center taskrow">
  <span class="taskTop">Comienzo</span>
</div>
<div class="row center taskrow">
  <span class="taskDown">{{convertirFecha(tarea.fechaInicio,null)}}</span>
</div>
</div>

<div class="col m4">
  <div class="row center taskrow">
    <i class="medium material-icons">date_range</i><span></span>
  </div>
  <div class="row center taskrow">
    <span class="taskTop">Finalización</span>
  </div>
  <div class="row center taskrow">
    <span class="taskDown">{{convertirFecha(tarea.fechaFin,$index)}}</span>
  </div>
</div>
</div>

<div class="row" ng-show="esUserTask(tarea.tipo)&&!estaAbierta(tarea.fechaFin)">
  <h6> <b>Campos de Entrada</b></h6>
  <table class="bordered">
    <thead>
      <thead>
        <th>Campo</th>
        <th>Valor</th>
      </thead>
    <tbody>
      <tr ng-repeat="(key,value) in tarea.campos">
        <td>{{key}}</td>
        <td>{{value}}</td>
      </tr>
    </tbody>
  </table>
</div>

<div class="row center" ng-show="esUserTask(tarea.tipo)&&estaAbierta(tarea.fechaFin)&&(grupo===!JefeProyecto)">
  <a id="botonCompletar" type="submit" href="#" ng-click="completarTarea(tarea)" class="btn enfasisbck waves-effect waves-light">Completar
Tarea<span class="glyphicon glyphicon-paperclip"></span></a>
</div>
</li>
</ul>
</div>
</div>
```

```
<div class="row center">  
  <a class="btn-floating tooltipped btn-large waves-effect blue lighten-4" ui-sref='stateEvolutivos' data-position="bottom" data-delay="100" data-  
tooltip="Volver" ui-sref-opts="{reload: true, notify: true}"><i class="material-icons grey-text text-darken-1">navigate_before </i></a>  
</div>  
  
</div>
```

controller5.js

```
'use strict';

angular.module('cofabpm.controllers')
  .controller('Controller5', ['$scope', '$rootScope', '$state', '$stateParams', '$timeout', 'cofabpmService',
    function($scope, $rootScope, $state, $stateParams, $timeout, cofabpmService) {

      $scope.motorErroneo = false;
      $scope.init = function() {
        $rootScope.veficarEntradaCorrecta();

        $rootScope.componentesInicio();

        cofabpmService.infoMotor()
          .then(
            function(res) {
              $scope.motorErroneo = false;
              $scope.infoMotor = {};
              $scope.infoMotor.countBPMs = res.data.processDefinitionCount
              $scope.infoMotor.countEvolutivos = res.data.runningProcessInstanceCount['BPM_Evolutivo (v1)']
              $scope.infoMotor.countEvolutivosTerminados = res.data.completedProcessInstanceCount['BPM_Evolutivo (v1)']
              $scope.infoMotor.tareasPendientes = res.data.openTaskCount;
              $scope.infoMotor.tareasHoy = res.data.completedTaskCountToday;
            },
            function(reason) {
              $scope.motorErroneo = true;
            }
          );
      }

      $scope.detalleBPM = function(nombreArchivo) {
        $scope.nombreBPM = nombreArchivo;
        $('#modalBPM').modal('open');
      }

    }
  ]);
```

salamaquinas.html

```
<nav class="nav-extended">
  <div class="nav-content blue lighten-4">
    <ul class="tabs tabs-transparent">
      <li class="tab"><a ui-sref='stateBuzones'>Buzón de Tareas</a></li>
      <li class="tab"><a ng-click="altaProyecto()">Nuevo Evolutivo</a></li>
      <li class="tab"><a ui-sref='stateEvolutivos'>Consultar Evolutivos</a></li>
      <li class="tab active"><a ui-sref='stateSalaMaquinas'>Sala de Máquinas</a></li>
      <li class="tab"><a ui-sref='stateCerrarSesion'>Cerrar Sesión</a></li>
    </ul>
  </div>
</nav>

<div class="container aireArriba" ng-init="init()">

  <h3>Sala de Máquinas</h3>

  <div ng-if="!motorErroneo">
    <div class="row center">
      <div class="col m8 offset-m2">
        <div class="card light-green lighten-4">
          <div class="card-content">
            <h4>Estado del Motor Activiti BPM:</h4>
            <h4> Correcto y en Ejecución</h4>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="row center">
    <div class="statContainer col m2 offset-m1">
      <div class="row center">
        <h1 class="statTop">{{infoMotor.countBPMs}}</h1>
      </div>
      <div class="row center">
        <h6 class="statDown">Procesos Definidos</h6>
      </div>
    </div>
    <div class="statContainer col m2 ">
      <div class="row center">
        <h1 class="statTop">{{infoMotor.countEvolutivos}}</h1>
      </div>
      <div class="row center">
        <h6 class="statDown">Evolutivos en Ejecucion</h6>
      </div>
    </div>
  </div>
</div>
```

```

</div>
<div class="statContainer col m2">
  <div class="row center">
    <h1 class="statTop">{{infoMotor.tareasPendientes}}</h1>
  </div>
  <div class="row center">
    <h6 class="statDown">Tareas Pendientes</h6>
  </div>
</div>
<div class="statContainer col m2">
  <div class="row center">
    <h1 class="statTop">{{infoMotor.countEvolutivosTerminados}}</h1>
  </div>
  <div class="row center">
    <h6 class="statDown">Evolutivos Completados</h6>
  </div>
</div>
<div class="statContainer col m2">
  <div class="row center">
    <h1 class="statTop">{{infoMotor.tareasHoy}}</h1>
  </div>
  <div class="row center">
    <h6 class="statDown">Tareas Completadas Hoy</h6>
  </div>
</div>
</div>
<div class="row">
  <div class="col s12 m4 offset-m4">
    <div class="card">
      <div class="card-image">
        
        <a class="halfway-fab waves-effect waves-light blue btn-floating modal-trigger" ng-click="detalleBPM('BPMEvolutivo.jpg')" ><i class="material-
icons">zoom_in</i></a>
      </div>
      <div class="card-content">
        <span class="card-title">BPM_Evolutivo (v1)</span>
        <p>BPM Encargado de modelar el servicio del ciclo de vida de un evolutivo desde su entrada en Factoría hasta su entrega a cliente </p>
      </div>
    </div>
  </div>
<!--
<div class="col s12 m3">
  <div class="card">
    <div class="card-image">
      

```

```

    <a class=" halfway-fab waves-effect waves-light blue btn-floating modal-trigger" ng-click="detalleBPM('ksxfR.png)" ><i class="material-
icons">zoom_in</i></a>
    </div>
    <div class="card-content">
        <span class="card-title">BPM_Incidencia (v1)</span>
        <p>I am a very simple card. I am good at containing small bits of information. I am convenient because I require little markup to use
effectively.</p>
    </div>
</div>
-->
</div>
</div>

<div ng-if="motorErroneo">
<div class="row center">
<div class="col m8 offset-m2">
<div class="card red accent-1 white-text">
<div class="card-content">
    <h4>Estado del Motor Activiti BPM:</h4>
    <h4> Parado o con Error de Conexión</h4>
</div>
</div>
</div>
</div>
<div class="row center">
    </img>
</div>
</div>

<!-- Modal Structure -->
<div id="modalBPM" class="modal">
<div class="modal-content">
<div class="row center">
    
</div>
</div>
</div>
</div>
</div>

```



Alcala de Henares Septiembre de 2017