



TESINA DE LICENCIATURA

Título: Despliegue ágil en proyectos de gran escala

Autores: Alejandro Javier Marfil y Mauro Orias

Director: Claudia Pons

Carrera: Licenciatura en Sistemas Plan 2015

Resumen

El despliegue de una aplicación es una parte muy importante en el desarrollo de software, mediante el mismo, se pone en marcha lo construido en un determinado tiempo, ya sea un sistema nuevo de cero o una nueva versión del mismo. Dicho producto de software a desplegar, en organizaciones de gran tamaño, involucra el esfuerzo de las distintas áreas implicadas de la organización (diseño y desarrollo, análisis funcional, testing, capacitación, arquitectura, soporte, seguridad, coordinación, gestión de proyectos, etc.).

En proyectos de gran tamaño, cuyo sistema a desplegar es una pieza fundamental para el funcionamiento de la organización, el proceso de despliegue cobra una relevancia notoria. Por tal motivo, es fundamental que el mismo se lleve a cabo de la manera más rápida y eficiente posible, permitiendo entregar nueva funcionalidad a los usuarios en el menor tiempo posible sin afectar la operatoria normal del sistema.

Palabras Claves

Despliegue de Software, Metodologías ágiles, Scrum, Gestión de Proyecto, Software, BPM, Workflow.

Trabajos Realizados

Se expone un análisis acerca de cómo es tratado por la Ingeniería de software el proceso de despliegue a través de diferentes modelos.

Se describe la metodología ágil scrum y como se utilizaron ideas de la misma para mejorar la organización y planificación en la etapa de despliegue.

Implementación y descripción de la herramienta P.E.P. utilizada en un proyecto de gran escala para mejorar la gestión y ejecución del proceso de despliegue.

Conclusiones

Durante la etapa del despliegue, vemos que se pueden aplicar las ideas que propone la metodología ágil SCRUM, haciendo algunas adaptaciones. Obteniendo muy buenos resultados en cuanto a la organización y planificación de los despliegues.

Además en los proyectos de gran tamaño, es crucial la utilización de una herramienta para la ejecución y gestión de las tareas que conforman un proceso de despliegue. A fin de evitar errores humanos y agilizar la comunicación entre los participantes.

Trabajos Futuros

Con la nueva versión de la herramienta PEP, encontramos que pueden surgir nuevos requerimientos que ayuden y potencien aún más el proceso de despliegue. Como por ejemplo extender la herramienta para que se realicen tareas automáticas o semi automáticas a través de la interacción con sistemas colaborativos.



Universidad Nacional de La Plata

Facultad de Informática

Tesina de Grado

Despliegue ágil en proyectos de gran escala

Directora: Claudia Pons

Autores: Mauro Orias - Alejandro Javier Marfil

Mayo 2018

INDICE

1. Introducción	4
1.1. Motivación	4
1.2. Objetivos	4
1.3. Estructura de la tesis	5
2. Estado del Arte	7
2.1. Proceso de despliegue	7
2.2. El despliegue y la ingeniería de software	7
2.2.1. Modelos	7
2.2.2. Detalle de los Modelos	8
2.2.3. Comparación	12
2.3. Metodología Agile - SCRUM	13
2.3.1. La esencia de Scrum	14
2.3.2. Flujo SCRUM	15
2.3.3. Elementos de SCRUM	15
2.3.4. Roles	16
2.3.5. Poda de requerimientos	17
2.3.6. Product Backlog	18
2.3.7. Sprint Scrum	18
2.3.8. Planificación	18
2.3.9. Rol del Scrum Master durante la planificación:	19
2.3.10. Sprint Backlog	19
2.3.11. Scrum diario (Daily)	20
2.3.12. Rol del Scrum Master durante el Scrum	20
2.3.13. Estimaciones	20
2.3.14. Builds continuos y pruebas básicas	21
2.3.15. Revisión del Sprint	21
2.3.16. Reunión retrospectiva	22
2.4. Guía para despliegues de Sistemas	22
2.4.1. Subprocesos	24
3. Proyecto de referencia eSidif	27
3.1. Introducción	27
3.2. Características del proyecto	28
3.3. Proceso de despliegue en eSidif	37
3.3.1. Evolución histórica	37

3.3.2. Organización del despliegue ágil	39
4. Primer prototipo de la herramienta P.E.P.	46
4.1. Introducción	46
4.2. Objetivos de la primera versión	46
4.3. Descripción funcional	46
4.3.1. Casos de uso	47
4.3.2. Vistas	58
4.4. Diagrama de clases y decisiones de diseño	68
4.5. Arquitectura general y herramientas utilizadas	73
4.5.1. Tecnologías utilizadas	75
4.6. Resultados obtenidos	76
5. Nueva versión de P.E.P	78
5.1. Objetivos de la nueva versión	78
5.2. Cambios funcionales	78
5.2.1. Casos de uso	78
5.2.2. Vistas	86
5.3. Cambios en la arquitectura general y herramientas	97
5.3.1. Cambios en las tecnologías utilizadas	97
5.4. Cambios en el diseño	98
5.5. Simulación de una ejecución de despliegue	104
5.6. Resultados obtenidos	104
6. Trabajos relacionados	106
7. Conclusiones y trabajos futuros	109
7.1. Conclusiones	109
7.2. Trabajos futuros	110
Referencias	112

1. Introducción

1.1. Motivación

El despliegue de una aplicación es una parte muy importante en el desarrollo de software, mediante el mismo, se pone en marcha lo construido en un determinado tiempo, ya sea un sistema nuevo de cero o una nueva versión del mismo. Dicho producto de software a desplegar, involucró el esfuerzo de las distintas áreas implicadas de la organización (diseño y desarrollo, análisis funcional, testing, capacitación, arquitectura, soporte, seguridad, coordinación, gestión de proyectos, etc.).

En proyectos de gran tamaño, cuyo sistema a desplegar es una pieza fundamental para el funcionamiento de la organización, el proceso de despliegue cobra una relevancia notoria. Por tal motivo, es fundamental que el mismo se lleve a cabo de la manera más rápida y eficiente posible.

Para que el equipo de despliegue, pueda planificar, gestionar y llevar adelante la ejecución del mismo, es fundamental que se utilicen las herramientas y metodologías que permitan lograr los objetivos esperados promoviendo la mejora continua.

Desde la ingeniería de software han surgido muchas metodologías relacionadas al proceso de desarrollo, sin embargo el proceso de despliegue no ha sido abordado con la relevancia que merece.

1.2. Objetivos

Los objetivos de la tesis son:

- Hacer una introducción acerca de la importancia del proceso de despliegue, sobre todo en proyectos grandes.
- Investigar acerca de cómo es abordado por la ingeniería de software el proceso de despliegue.
- Explicar cómo se fue agilizando el proceso de despliegue en un proyecto de gran escala, en el cual trabajamos.
- Presentar un prototipo de la herramienta que sirva de soporte para gestionar las tareas que conforman un despliegue.

- Realizar una simulación de un despliegue, mostrando los resultados.

1.3. Estructura de la tesis

La presente tesis se encuentra dividida en 7 capítulos, de los cuales haremos una breve descripción:

En el capítulo 1 “Introducción”, como su nombre lo indica, se hace una introducción a la tesis, en él se describe la motivación que llevo a realizarla como también los objetivos buscados con la misma. Finalmente se describe como está estructurada la tesis.

El capítulo 2 “Estado del Arte”, es un marco teórico sobre los temas relevantes que se desarrollan en la tesis, se describe que es un despliegue como así también su importancia en los proyectos de gran escala. También se indica como es tratado el proceso de despliegue por la ingeniería de software a través de los diferentes modelos, para luego hacer una comparación de los mismos. Además en el capítulo, se describe de forma detallada la metodología ágil SCRUM con sus aspectos más relevantes. Finalmente se presenta una guía para despliegues de sistemas, explicando los diferentes roles y subprocesos que intervienen en los mismos.

En el capítulo 3 “Proyecto de referencia eSidif”, se hace una introducción al proyecto esidif, el cual es utilizado como caso de estudio en el presente trabajo, enumerando tanto sus principales características, como así también la conformación de los grupos de trabajo. También se detalla cómo fue evolucionando a lo largo del tiempo, el proceso de despliegue en el proyecto. Finalmente se muestra cómo se agilizó el proceso de despliegue, utilizando las ideas que propone la metodología ágil SCRUM. Se agilizó el proceso de despliegue, utilizando las ideas que propone la metodología ágil SCRUM.

Ya en el capítulo 4 “Descripción del primer prototipo de la herramienta P.E.P.”, se presenta la primera versión de la herramienta utilizada para mejorar el proceso de despliegue. Se hace una introducción de la misma y se enumeran los objetivos buscados con ella. Además se hace una descripción funcional utilizando casos de uso, como así también se muestran las diferentes vistas de la aplicación. También se detalla el diagrama de clases utilizado para la solución, explicando las decisiones de diseño que se tomaron con las justificaciones correspondientes. Luego se detalla la arquitectura general que se determinó para la solución, como también las diferentes tecnologías utilizadas. Finalmente se explica cuáles fueron los resultados obtenidos con la primera versión de la herramienta.

El capítulo 5 "Nueva versión de P.E.P." comienza con un detalle de los objetivos buscados con la nueva versión. Luego se explican los cambios funcionales, indicando los casos de uso que siguen igual a la versión anterior, los que se agregan, los que se modifican y los que se quitan. También se hace un recorrido por todas las vistas indicando los cambios que se hicieron con respecto a la versión anterior. El capítulo continúa con una descripción de los cambios en la arquitectura general y las tecnologías utilizadas. Luego se describieron los cambios en el diseño, presentando el nuevo modelo y

detallando las decisiones de diseño que se tomaron junto a los motivos. Sigue el capítulo con una explicación de la simulación que se hizo para mostrar la funcionalidad principal y finalmente se muestran los resultados obtenidos con la nueva versión de la herramienta.

En el capítulo 6 "Trabajos relacionados", como su nombre lo indica se describen los trabajos relacionados.

Finalmente, en el capítulo 7 "Conclusiones y trabajos futuros", se enuncian las conclusiones a las que se arribaron con este trabajo y los trabajos futuros.

2. Estado del Arte

2.1. Proceso de despliegue

El proceso de despliegue es un tema relevante en lo que se refiere al desarrollo de software. A pesar de ello, la ingeniería de software continúa centrándose en abordar los problemas del desarrollo desde la mejora de procesos pero sin abordar de manera sistemática el despliegue como un conjunto de temas específicos a ser tratados.

El proceso de despliegue se define como un conjunto de actividades que deben realizarse para una correcta puesta en marcha del software que se desarrolla, adapta y/o mantiene, en un contexto específico para su uso, la cual requiere tareas referidas a la infraestructura tecnológica, a las particularidades propias del despliegue del producto, como así también capacitación a los recursos humanos involucrados en el cambio tecnológico que será desplegado.

Un despliegue en un proyecto de gran escala, como en el que nosotros trabajamos, requiere una planificación, que definirá cómo se realizará este proceso, además de prever cualquier situación que lo pueda afectar y establecer estrategias para mitigar riesgos. Coordinando las tareas y recursos, previendo las dependencias que puedan existir, utilizando los mecanismos y/o herramientas que hagan falta para poder sincronizar y ejecutar el despliegue de manera exitosa y eficientemente.

2.2. El despliegue y la ingeniería de software

Para el éxito de un proyecto de software, una de las etapas relevantes es el proceso de despliegue del mismo. Desde su surgimiento, la ingeniería de software ha evolucionado abordando diferentes áreas para la mejora de la especialidad. Sin embargo el despliegue de sistemas ha sido un tema que si bien se lo considera importante, no ha sido abordado metodológicamente como etapa del desarrollo de software.

2.2.1. Modelos

En la actualidad existen diversos modelos de proceso que detallan las diferentes actividades que deben llevarse a cabo para los proyectos de tecnología de la información, dividiéndolas en procesos y subproceso.

Los modelos más utilizados por la industria del software y servicios informáticos, son: el Estándar IEEE 1074 “Standard for Developing Software Life Cycle Processes” [1] y el ISO 12207 “Standard for Information Technology - Software Life Cycle Processes” [2], así como el modelo CMMI-dev “Capability Maturity Model Integration For Development” [3] cuyo enfoque se centra en el desarrollo de software. Por otro lado el RUP “Proceso unificado de Software” es la metodología más utilizada en análisis e implementación de sistemas orientados a objetos. En cuanto a la estandarización de actividades de gestión de proyectos, la guía de PMBOK “A Guide to the Project Management Body of Knowledge” [4], define el conjunto de actividades desde una perspectiva más genérica, aplicable en diferentes tipos de proyectos. El ITIL foundation Handbook [5] trabaja sobre las actividades para la gestión del servicio y, por último, el SWBOK presenta una guía al conocimiento presente en Ingeniería de Software. En cuanto a las metodologías ágiles, podemos mencionar Scrum [10] que es una de las más utilizadas hoy en día como así también XP [11] extreme programming que propone una fuerte interacción entre el equipo de desarrollo y los usuarios finales del sistema.

2.2.2. Detalle de los Modelos

El estándar de IEEE 1074 [1] el proceso del software se compone de 17 grupos de actividades, cada uno de los cuales contiene actividades que son responsables de satisfacer sus requisitos asociados. El proceso se compone de un total de 65 actividades. IEEE 1074 engloba las siguientes familias de actividades:

- Gestión del Proyecto
- Actividades Orientadas al desarrollo
- Actividades integrales.

En el grupo de actividades orientadas al desarrollo, se diferencian las de pre-desarrollo, desarrollo en sí mismas y post-desarrollo. En este último grupo de actividades de post-desarrollo, el estándar define el proceso de Instalación, definida como el transporte y la instalación de un sistema de software desde el entorno de desarrollo al entorno de destino.

Incluye la carga – si es necesario - de la base de datos, las modificaciones necesarias del software, las comprobaciones en el entorno de destino y la aceptación del cliente. Si durante la instalación surge algún problema, se identifica e informa acerca de él.

En cuanto al proceso software según el estándar ISO/IEC 12207 – 1995 [2], divide las actividades del ciclo de vida del software en tres procesos primarios, ocho procesos de soporte y cuatro procesos organizativos. Los procesos primarios son:

- Procesos principales del ciclo de vida
- Procesos de soporte del ciclo de vida
- Procesos organizativos del ciclo de vida

No obstante esta separación, no incluye explícitamente un proceso que defina la puesta en marcha o el despliegue de software.

El modelo CMMI-DEV [3], define las siguientes áreas:

- Gestión de la Configuración (Configuration Management, CM).
- Gestión de los Acuerdos con Proveedores (Supplier Agreement Management, SAM).
- Gestión de los Requerimientos (Requirements Management, REQM).
- Aseguramiento de la Calidad del Proceso y del Producto (Process and Product Quality Assurance, PPQA).
- Medición y Análisis (Measurement and Analysis, MA).
- Monitoreo y Control del Proyecto (Project Monitoring and Control, PMC).
- Planeamiento del Proyecto (en inglés, Project Planning, PP).

El RUP propone un proceso iterativo e incremental dividido en cuatro fases, inicio, elaboración, construcción y transición. Por otro lado desarrolla flujos de trabajo del proceso los que están asociados a la construcción propiamente dicha del software que son:

- Modelado del negocio
- Requisitos
- Análisis y diseño
- Implementación
- Pruebas y despliegue

Y también los flujos de trabajo de soporte que son:

- Gestión del cambio y configuraciones
- Gestión de proyecto y entorno

La fase de transición tiene como propósito la entrega del producto de software a la comunidad de usuarios [6].

Respecto a la gestión de proyectos en general, la guía PMBOK [4] provee fundamentos para la gestión de proyectos, aplicables a un amplio rango de proyectos, generalmente aceptados como las mejores prácticas para la gestión.

El PMBOK reconoce 5 grupos de procesos básicos y 9 áreas de conocimiento comunes a casi todos los proyectos. Los grupos de procesos básicos incluyen los de:

- Iniciación
- Planificación
- Ejecución
- Seguimiento y Control
- Cierre

En esta guía, los procesos se traslapan e interactúan a través de un proyecto o fase. Se describen en términos de entradas (documentos, planes, diseños, etc.), herramientas y técnicas (mecanismos aplicados a las entradas), y salidas (documentos, productos, etc.).

Dos de sus capítulos presentan una guía básica de prácticas acerca de cómo debe gestionarse los recursos humanos y las comunicaciones dentro de un proyecto.

Al analizar el ITIL, Information Technology Infrastructure Library, [5] se observa que consta de 5 libros basados en el ciclo de vida del servicio. Estos son:

- Estrategia del Servicio
- Diseño del Servicio
- Transición del Servicio
- Operación del Servicio
- Mejora Continua del Servicio.

A su vez, la transición del servicio está formada por 7 subprocesos. Estos son:

- Planificación y soporte a la transición
- Gestión de Cambios
- Gestión de la configuración y activos del servicio
- Gestión de entregas y despliegues
- Validación y pruebas
- Evaluación y Gestión del conocimiento.

En cuanto al SWBOK, está formado por 10 áreas de conocimiento. Estas son:

- Requisitos de Software
- Diseño de Software
- Construcción de Software
- Pruebas de Software
- Mantenimiento de Software
- Gestión de configuración
- Gestión de la Ingeniería de Software
- Proceso de Ingeniería de Software
- Herramientas y métodos de la Ingeniería de Software
- Calidad del Software

El Proceso de Ingeniería del Software se ocupa principalmente de la implementación, evaluación, medición, gestión, cambio y mejora de los procesos del ciclo de vida. [7]

En cuanto a las metodologías ágiles de desarrollo de software, se puede destacar a Scrum [10] que constituye una metodología de desarrollo ágil que se aplica en proyectos en los cuales se trabaja con requisitos inestables y que requieren flexibilidad y rapidez en el desarrollo. Scrum propone tres etapas en las cuales tanto los desarrolladores como los usuarios juegan un papel muy importante, desde la participación en la planificación hasta la toma de decisiones en cada instancia.

Por su parte, XP [11] propone una fuerte interacción entre el equipo de desarrollo y los usuarios finales del sistema, llegando a proponer que un representante del cliente trabaje en forma conjunta con el equipo de desarrollo. También propone la programación de a pares y las reuniones frecuentes para evaluar el estado y avance de las diferentes actividades.

Desde esta perspectiva metodológica, se observa que existen modelos y técnicas que abordan el trabajo en equipo y la necesidad de definir una dinámica ágil en los proyectos de TI, pero desde un enfoque de la administración de los recursos en general y de la reorganización del proceso, pero no abordan los puntos críticos de instancias como el despliegue, que requiere de todos los involucrados en diferentes roles para el logro del objetivo y una definición específica de actividades en esa instancia particular del proceso.

El conjunto de modelos y estándares existentes, ordenan de manera prescriptiva al conjunto de actividades esenciales, no ordenadas en el tiempo, que deben llevarse a cabo para un correcto desarrollo de proyectos de construcción, adaptación y/o mantenimiento de software.

Sin embargo, estos modelos, no dan cuenta de manera explícita a la definición de actividades esenciales que deben realizarse para una correcta puesta en marcha de los software que se desarrollan, adaptan y/o mantienen, entendiendo el despliegue de un sistema en el contexto específico de su uso, que requiere de un conjunto de actividades

que aborden las tareas específicas referidas a la infraestructura tecnológica, a las particularidades de del despliegue del producto, así como a los recursos humanos involucrados en el cambio tecnológico que será desplegado.

2.2.3. Comparación

La siguiente tabla, presenta una síntesis comparativa entre los modelos estudiados, haciendo foco en los procesos o subprocesos que de alguna manera hacen referencia a la etapa de despliegue.

Modelo	Proceso	Características
RUP	Fase de transición Proceso de despliegue	Define como tareas principales: probar el software en su entorno operativo, empaquetar el software para la distribución y distribuirlo, instalarlo, capacitar a los usuarios finales y la fuerza de ventas, y la migración del software existente o la conversión de la base de datos
Scrum	Etapa de cierre	Dentro de esta etapa se encuentra la integración, pruebas del sistema, documentación de usuario, preparación del material de formación y marketing.
CMMI	Integración de producto	Preparar para la integración de producto. Asegurar la compatibilidad de la interfaz. Ensamblar los componentes de producto y entregar el producto
ISO 12207	Instalación, aceptación y operación de software	Los objetivos de estos procesos son: 1. Instalar el producto de software que cumpla con los requisitos acordados en el entorno de producción. 2. Ayudar al usuario a lograr la confianza de que el producto cumple con los requisitos. 3. Utilizar el producto de software en su ambiente de producción y prestar apoyo a los usuarios.
IEEE 1074	Instalación	Compuesto por las actividades: Distribución de software, Instalación del software, Aceptación

		de software en el entorno operativo
ITIL	Transición	Compuesto por los subprocesos: Gestión de Cambios, Planificación y Soporte de Transición, Gestión de Ediciones e Implementación, Validación y Pruebas de Servicios, Desarrollo y Personalización de Aplicaciones, Activos de Servicio y Gestión de la Configuración y Gestión del Conocimiento
PMBOK	Grupo procesos de cierre	Procedimiento de cierre administrativo y Procedimiento de cierre de contrato.
SWEBOK	Proceso de implementación y cambios	Se basa en lo definido en las normas IEEE 1074 y 12207. Propone un proceso de implantación para el proceso de ingeniería del software a utilizar en la organización, no del software.

Tabla 1 – Comparación modelos vigentes

Del estudio detallado de los modelos vigentes, se observa como resultado que en todos los casos asumen un conjunto de actividades que deben ser ejecutadas para la puesta en marcha de un sistema, no obstante, las propuestas contienen diferentes niveles de detalle y desagregados que no atienden en forma completa y sistemática la problemática del despliegue.

2.3. Metodología Agile - SCRUM

Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Pone su atención en los valores y prácticas de gestión, en lugar de los requerimientos, prácticas de desarrollo, implementación y otras cuestiones técnicas. Ésta metodología delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles, por ende, es muy flexible y los integrantes del equipo pueden optar por organizar la forma de interactuar entre ellos.

La terminología “Scrum” procede del deporte llamado rugby, donde se designa al acto de preparar el avance del equipo en unidad pasando la pelota a uno y otro jugador. Scrum es adaptable, ágil, auto-organizado y con pocos tiempos muertos. Esta metodología ágil fue desarrollada por Jeff Sutherland y elaborada más formalmente por Ken Schwaber. Poco tiempo después Sutherland y Schwaber se unieron para refinar y extender Scrum. Se la ha llegado a conocer como una herramienta de hiper-productividad. Schwaber se dio cuenta entonces de que un proceso necesita aceptar el cambio, en lugar de esperar predictibilidad. Se enfoca en el hecho de que procesos definidos y repetibles sólo funcionan para atacar problemas definidos y repetibles con gente definida y repetible en ambientes definidos y repetibles. Toma el cambio como una forma de entregar al final del desarrollo algo más cercano a la verdadera necesidad del Cliente. Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.

Se basa en los principios ágiles:

- Privilegiar el valor de la gente sobre el valor de los procesos.
- Entregar software funcional lo más pronto posible.
- Predisposición y respuesta al cambio.
- Fortalecer la comunicación y la colaboración.
- Comunicación verbal directa entre los implicados en el proyecto.
- Simplicidad; eliminación de artefactos innecesarios en la gestión del proyecto.

2.3.1. La esencia de Scrum

- Más que una metodología de desarrollo es una herramienta para gestionar proyectos.
- No contiene definiciones en áreas de ingeniería.
- Con visión de que el trabajo es efectuado por equipos auto-organizados y auto dirigidos, logrando motivación, responsabilidad y compromiso.
- Está basada en un proceso constructivo iterativo e incremental donde las iteraciones tienen duración fija.

- Contiene definición de roles, prácticas y productos de trabajo escritas de forma simple.
- Está basada en un conjunto de valores y principios.

2.3.2. Flujo SCRUM

En la siguiente figura, se puede visualizar el flujo de trabajo de la metodología Scrum. Es decir, desde el product backlog, la duración del sprint, las iteraciones, etc.

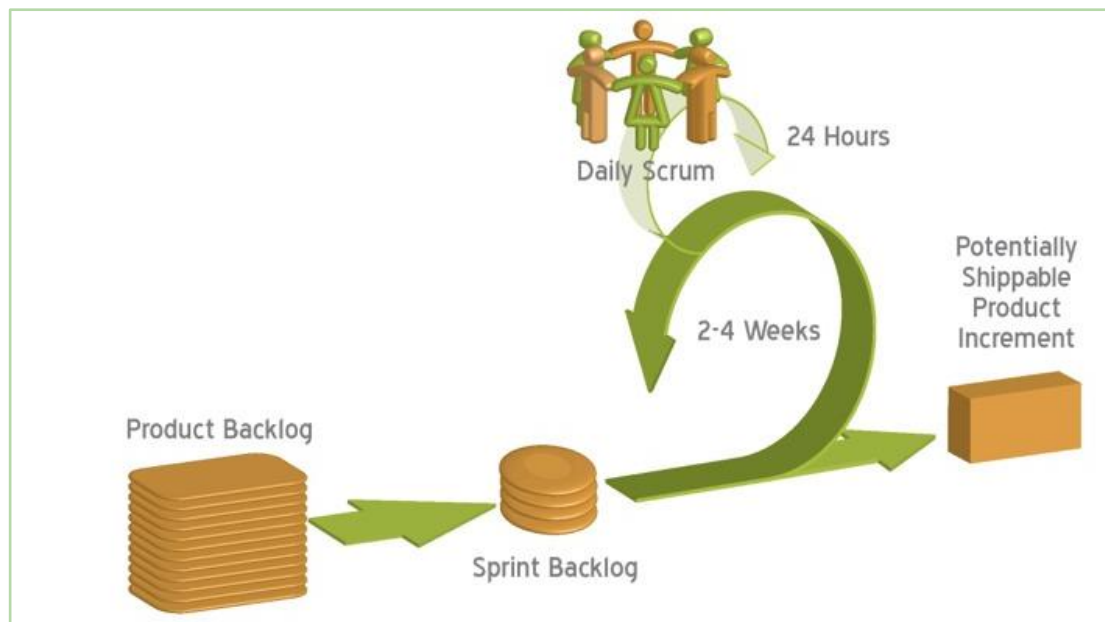


Figura 1 – Flujo de trabajo en SCRUM

2.3.3. Elementos de SCRUM

A continuación se mencionan los elementos que componen scrum y luego se realiza una descripción de los mismos.

- Roles
 - ProductOwner
 - Scrum Master
 - Team (Equipo)
- Poda de requerimientos
- Product Backlog
- Sprint
 - Planificación (Planning)
 - Sprint Backlog
 - Scrum
 - Estimaciones
 - Builds continuos
 - Revisión del Sprint
 - Reunión retrospectiva
- Valores
 - Foco, comunicación, respeto y coraje.

2.3.4. Roles

La dimensión del equipo total de Scrum no debería ser superior a veinte personas. Si hay más, lo más recomendable es formar varios equipos. No hay una técnica oficial para coordinar equipos múltiples, pero se han documentado experiencias de hasta 800 miembros, divididos en Scrum de Scrum definiendo un equipo central que se encarga de la coordinación, las pruebas cruzadas y la rotación de los miembros. Scrum tiene una estructura muy simple. Todas las responsabilidades del proyecto se reparten en 3 roles:

- **Productowner (Dueño del producto)** Representa a todos los interesados en el producto final. Es el responsable oficial del proyecto, gestión, control y visibilidad de la lista de acumulación o lista de retraso del producto (Product Backlog). Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar.

Sus áreas de responsabilidad son:

- Financiación del proyecto.
 - Requisitos del sistema.
 - Retorno de la inversión del proyecto.
 - Lanzamiento del proyecto.
- **Scrum Master (Líder del proyecto)** Responsable del proceso Scrum, de cumplir la meta y resolver los problemas. Así como también, de asegurarse que el proyecto se lleve a cabo de acuerdo con las prácticas, valores y reglas de Scrum y que progrese según lo previsto. Interactúa con el cliente y el equipo. Coordina los encuentros diarios, y se encarga de eliminar eventuales obstáculos. Debe ser miembro del equipo y trabajar a la par.
 - **Team (Equipo)** Responsables de transformar el Backlog de la iteración en un incremento de la funcionalidad del software, es decir, de convertir el product backlog en un software entregable. El equipo tiene la autoridad para reorganizarse y definir las acciones necesarias o sugerir eliminación de impedimentos.
Características del equipo:
 - Auto-gestionado
 - Auto-organizado
 - Multi-funcional

El número ideal para la conformación de un equipo es entre 8 y 10 personas.

2.3.5. Poda de requerimientos

La primera actividad es armar una lista exhaustiva de los requerimientos originales del sistema. Luego se procede a ver qué requerimientos son realmente necesarios, cuáles pueden posponerse y cuáles eliminarse. Para ello debe identificarse un representante con capacidad de decisión, priorizar los requerimientos en base a su importancia y acordar cuáles son los prioritarios para la fecha de entrega. La poda de requerimientos es una

buena práctica en los modelos ágiles, se hace lo que el cliente realmente desea, no más. Es decir, se efectúa una priorización de requerimientos.

2.3.6. Product Backlog

Con los requerimientos priorizados y podados armamos el Backlog de Producto. Este es una forma de registrar y organizar el trabajo pendiente para el producto (actividades y requerimientos). Es un documento dinámico que incorpora constantemente las necesidades del sistema. Por lo tanto, nunca llega a ser una lista completa y definitiva. Se mantiene durante todo el ciclo de vida (hasta finalizar el sistema) y es responsabilidad del ProductOwner.

2.3.7. Sprint Scrum

Un Sprint es el período de tiempo durante el que se desarrolla un incremento de funcionalidad. Constituye el núcleo de Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas “carreras”.

Características del Sprint:

- Duración máxima de 30 días.
- Durante el Sprint no se puede modificar el trabajo que se ha acordado en el Backlog.
- Sólo es posible cambiar el curso de un Sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por alguna de las razones siguientes:
 - - La tecnología acordada no funciona.
 - - Las circunstancias del negocio han cambiado.
 - - El equipo ha tenido interferencias.

2.3.8. Planificación

Se planifica en detalle el trabajo al inicio de cada Sprint asumiendo que los objetivos no van a cambiar durante el mismo. De esta manera se atenúa el riesgo.

Aspectos a tener en cuenta sobre la planificación de un Sprint:

- Una determinación general de alcance, frecuentemente basada en una EDT (Estructura de División del Trabajo).

- Estimaciones de esfuerzo de alto nivel realizadas durante la etapa de concepción del proyecto.
- Esfuerzo dedicado a labores de soporte o de preparación de los ambientes requeridos por el proyecto.
- Esfuerzo asociados a las reuniones diarias, de planificación y de revisión.
- Requerimientos de recursos de infraestructura o logísticos (máquinas, redes, licencias, papel, pizarras, etc.).
- Habilidades presentes y necesarias en el equipo.
- Restricciones asociadas al conocimiento del negocio, la tecnología o externas (legales, reglamentarias, estándares, etc.).

2.3.9. Rol del Scrum Master durante la planificación:

- Dirige la planificación.
- Es vínculo entre el equipo y el ProductOwner del proyecto.
- Registra problemas y riesgos detectados durante la planificación.
- Registra las tareas, asignaciones y estimaciones.
- Inicia el Backlog del Sprint.

2.3.10. Sprint Backlog

Se refiere al trabajo o tareas determinadas por el equipo para realizar en un Sprint. Es decir, la conversión de tareas a un producto funcional.

Características:

- Las tareas se estiman en una duración entre 1 a 20 horas de trabajo. Las de mayor duración deben intentar descomponerse en sub-tareas de ese rango de tiempo.
- La estimación se actualiza día a día.

2.3.11. Scrum diario (Daily)

Scrum asume que el proceso es complejo y que es necesario revisarlo frecuentemente, por eso se realiza una reunión diaria de seguimiento. El encuentro diario impide caer en el dilema señalado por Fred Brooks: “¿Cómo es que un proyecto puede retrasarse un año? Un día a la vez”.

El foco de la reunión es determinar el avance en las tareas y detectar problemas o “bloqueos” que están haciendo lento el progreso del equipo o que eventualmente impidan a un equipo cumplir con la meta del Sprint. La idea es que ningún problema quede sin resolver o, por lo menos, sin iniciar alguna acción de respuesta dentro de las 24 horas después de su detección. La reunión es además un espacio definido para que cada miembro del equipo comunique a los demás el estado de su trabajo y por lo tanto reafirme el compromiso.

2.3.12. Rol del Scrum Master durante el Scrum

Dentro de los roles del Scrum Master durante el Scrum encontramos los siguientes:

- Dirigir la reunión y mantener el foco de atención.
- Realizar preguntas para aclarar dudas.
- Registrar, escribir y/o documentar los problemas para su resolución después de la reunión.
- Asegurarse que los miembros cuenten con el ambiente adecuado para la reunión.

2.3.13. Estimaciones

Las estimaciones se realizan por primera vez en la reunión de planificación al inicio del Sprint. Posteriormente, las tareas se re-estiman todos los días y se registran sus cambios en el Backlog del Sprint.

Esta actividad puede ser realizada inmediatamente antes o después del Scrum diario.

Algunas claves para la estimación son las siguientes:

- Siempre se realizan estimaciones de esfuerzo, no de duración.
- Siempre se estima el esfuerzo total pendiente para terminar la tarea, no se estima el esfuerzo consumido.
- Se buscan unidades manejables, lo usual es que estén en un mínimo de 2 horas y un máximo de 20. Si la tarea es muy corta se trata de juntarla con otras relacionadas. Si la tarea es muy grande se trata de descomponerla.

2.3.14. Builds continuos y pruebas básicas

La estrategia que generalmente se utiliza es la de Builds continuos y “smoke test” (prueba básica para la funcionalidad del sistema). El procedimiento de Builds continuos es el siguiente:

- Los programadores desarrollan según el Backlog del Sprint, y al finalizar, notifican al integrador.
- El integrador toma el código y lo integra con el resto del producto.
- Se compila el software y se prueba “por arriba” el producto, para verificar que no se haya roto.
- Si se encuentran problemas se devuelve al desarrollador.
- Se notifica al equipo que hay una nueva versión “estable” del código para usar como base.

2.3.15. Revisión del Sprint

El objetivo de la reunión de revisión es presentar el producto o porción del producto desarrollada por el equipo a los usuarios. La reunión se utiliza para detectar inconformidades mayores que se vuelven elementos del Backlog de Producto y que eventualmente se resuelven en el siguiente Sprint. A la reunión asisten el equipo, el Scrum Master, el Product Owner y todas las personas implicadas en el proyecto.

2.3.16. Reunión retrospectiva

Scrum involucra el concepto de mejora continua a través de las reuniones de retrospectión. Las reuniones buscan detectar los puntos positivos y negativos del Sprint para generar propuestas de mejora para futuros Sprints. Las reuniones de retrospectión son el concentrador del aprendizaje organizacional sobre el Scrum. Los puntos positivos y negativos se registran y se definen ítems de acción para cada uno. Los ítems de acción definidos se toman en cuenta en los siguientes Sprints.

A este tipo de reuniones asisten el equipo y el Scrum Master, y opcionalmente el Product Owner del Producto.

2.4. Guía para despliegues de Sistemas

La guía para despliegues que aquí se describe, fue desarrollada por el Grupo GIS (Grupo de Ingeniería de Software), conformado como un grupo de investigadores interuniversitario desde el año 2004. Sus trabajos se han focalizado en temas de calidad de software, modelos de proceso, experimentación en Ingeniería de Software e implementación de Sistemas.

Ante la carencia desde la Ingeniería de Software en sistematizar el proceso de despliegue en un contexto real, los investigadores Alicia Mon y Fernando López Gil, de la Universidad Nacional de La Matanza (UNLaM) y Universidad del Noroeste de la Provincia de Buenos Aires (UNNOBA) respectivamente, desarrollaron una guía de aplicación que contiene un conjunto de subprocesos, actividades, productos y roles involucrados en cada uno de ellos.

La guía, se ha estructurado sobre un conjunto de 10 sub-procesos a los cuales se les ha asignado un código de identificación y una tipificación según se refiera a los procesos del tipo de Gestión, Operación o Apoyo. Estos subprocesos se han tomado en base al análisis de los diferentes modelos estudiados, extrayendo de cada uno de ellos las actividades o grupos de actividades referidas a la implantación. En aquellos casos que las actividades eran similares en diversos modelos, se definió el objetivo de la misma y la denominación más usual.

Para la ejecución del conjunto de actividades definidas en el interior de cada subproceso, se define un conjunto de Roles responsables con asignación de funciones. Se utilizaron roles existentes y definidos por otros modelos vigentes, asociando a las actividades y responsabilidades específicas.

La siguiente Tabla, expone los roles definidos para la aplicación de la guía y la descripción de las funciones:

Nombre del Rol	Descripción
Cliente	Persona -física o jurídica- que mediante un pago solicita el producto software, discute las cláusulas del contrato y su modo de cierre. Habitualmente también define los requisitos.
Usuario referente	Usuario final que por sus conocimientos de la organización o del proceso que debe realizar se lo puede tomar como idóneo para definir los requisitos.
Usuario final	Persona a la cual va destinado el producto y que trabaja directamente con este.
Usuario área de soporte	Persona del área de IT interno a la organización que se ocupa de dar el soporte del software. Interactúa con el usuario final.
Analista funcional	Persona que hace de nexo entre los usuarios y el grupo de desarrollo. Tiene conocimiento del negocio y del uso de la aplicación.
Líder de proyecto	Persona responsable del planeamiento del proyecto, del control de su ejecución y de la gestión de los recursos económicos, materiales y humanos asignados al mismo.
Responsable de pruebas	Persona responsable de la planificación y ejecución de las pruebas del sistema. Particularmente pruebas de sistema y aceptación.
Capacitador	Persona responsable de transferir a los diferentes usuarios -y/o involucrados- el conocimiento para el uso del software.
Afectados	Todas las personas que son afectadas por el proyecto, con mayor o menor grado de involucramiento que no son usuarios directos del software.
Personal técnico	Personal que pertenece al grupo de desarrollo de la aplicación y que se ocupa de llevar a cabo diferentes procesos de tipo técnicos.
Responsable de procesos	Persona encargada de recopilar y documentar el funcionamiento de los diferentes procesos de la organización.
R. gestión de cambios	Persona responsable de la administración y registro de los cambios en los proyectos de la organización
Proveedor	Persona -física o jurídica- que mediante un cobro entregará el producto software que se requiere para la instalación.
Comprador	Persona responsable de las adquisiciones dentro de la organización.

Tabla 2 – Definiciones de roles

La ejecución del conjunto de subprocesos definidos generan como resultado uno o varios productos como salida. Por lo tanto, la guía contiene un conjunto de productos que deben ser elaborados por cada uno de los subprocesos y que a su vez constituyen el producto de entrada para otro subproceso.

La definición del Modelo de Proceso de Despliegue, incluye el conjunto de actividades, no ordenadas en el tiempo, pero vinculadas entre sí por las relaciones de entrada y salida, así como por los productos que cada subprocesos genera y los roles involucrados.

Para consultar la guía completa, con los subprocesos definidos, sus actividades, roles y la lista de todos los productos involucrados ver [12].

2.4.1. Subprocesos

Cada uno de los subprocesos incluye un conjunto de actividades, que en la guía propuesta se referencian de la siguiente manera:

DIST: Distribución del software ensamblado

Es el proceso que se encarga de ensamblar los componentes de acuerdo a la tecnología utilizada, su preparación para ser instalados en un nuevo entorno y la posterior distribución de los mismos a los diferentes puntos de instalación. Su objetivo reside en generar los componentes instalables del sistema. Lograr que estos componentes se encuentren disponibles en cada punto en que se deben instalar.

INST: Instalación de software

Es el proceso que se ocupa de transferir el nuevo software al entorno productivo. Se encarga de adaptar las condiciones de dicho entorno, de modo que el nuevo producto software pueda ejecutarse correctamente. Su objetivo es dejar el nuevo sistema funcionando en el entorno productivo.

CONF: Configuración de software

La configuración del software es el proceso en el cual se definen los parámetros del sistema de modo que éste responda a los diferentes casos de acuerdo a lo esperado por el usuario. Se basará para determinar este comportamiento en la información provista por la especificación, los casos de prueba y los usuarios referentes. Su objetivo es

personalizar el nuevo sistema de modo que cumpla con los requisitos definidos y pueda ser operado por el usuario en el ambiente real.

ACEP: Aceptación de software

La aceptación del software consiste en un análisis de la información de evaluación entregada, comparado con la información de aceptación del usuario prevista, de modo de garantizar que el software instalado funciona como se esperaba. Cuando los resultados del análisis satisfacen los requisitos de aceptación del usuario, el sistema de software instalado se acepta. Su objetivo es garantizar que el cliente Acepta el nuevo sistema, con su actual configuración y satisface los requisitos definidos.

CONV: Conversión de sistema

La conversión de sistema consistirá en la definición de la estrategia de puesta en marcha del nuevo sistema, ya sea reemplazando a un sistema anterior o automatizando procesos manuales preexistentes. Su objetivo es realizar el cambio operativo desde el sistema actual al nuevo.

CAPA: Capacitación de usuarios

El proceso de capacitación proporciona a todas las personas involucradas en el uso del nuevo sistema -operadores, personal de soporte, personal gerencial, etc.-, los conocimientos necesarios para llevar a cabo las actividades que les corresponden, familiarizándose con la aplicación y los procesos automatizados, a fin de disminuir los problemas de operación. Su objetivo reside en que todos los usuarios del sistema conozcan su operación y que todos los afectados por el cambio conozcan el impacto del nuevo software.

OPER: Operación de software

El proceso de operación del software es el que se encarga del uso habitual del mismo.

Su objetivo es desarrollar las actividades normales operativas que afectan o son afectadas por el software una vez instalado en el ambiente definitivo.

ACTP: Actualización de los Procesos de la Organización

La actualización de los procesos de la organización consiste en actualizar toda la documentación y definición de procesos, en lo referente o que son afectados por el sistema implantado. Su objetivo es dejar consistente la definición de procesos y documentación de la organización.

CIER: Cierre de proyecto

El proceso cerrar proyecto es el necesario para finalizar todas las actividades de los diferentes grupos de procesos a fin de cerrar formalmente el proyecto o una fase del mismo. Su objetivo es dar por concluidas las obligaciones entre las partes.

GEST: Gestión de la implantación

El proceso de gestión es el responsable de administrar y coordinar las diferentes actividades que se deben llevar a cabo a lo largo de la implantación. Su objetivo es gestionar los procesos que son necesarios para llevar a cabo la implantación, utilizando un modelo de referencia para la coordinación de las actividades, productos y los roles involucrados en un proyecto específico.

3. Proyecto de referencia eSidif

3.1. Introducción

Sancionada la Ley N° 24.156 de Administración Financiera y de los Sistemas de Control del Sector Público Nacional el 30 de septiembre de 1992 se desarrolla un sistema integrado de información financiera (SIDIF) cuyo principal propósito es la formulación del presupuesto nacional y registro de la ejecución presupuestaria.

El SIDIF fue concebido como un sistema integrado, compuesto por diversos subsistemas y módulos dentro de una visión funcional, que contempla la distribución de la base de datos lógica, en una base de datos central y tantas bases institucionales como SAF existentes (Servicios de Administración Financiera)

Bajo este esquema operativo, la Secretaría de Hacienda se comunicaba, a través del SIDIF Central con los sistemas periféricos instalados en los organismos (sistemas locales), con el sistema de gestión para la Unidades Ejecutoras de Préstamos Externos y demás aplicaciones que interactúan con la base de datos central. A través de esta comunicación la base central concentra el registro de la ejecución presupuestaria. La información de gestión permanece en las bases locales.

Dentro del SIDIF convivían una variedad de sistemas locales en las distintas entidades con características diferentes. Esta diversidad de aplicativos incrementó el costo de mantenimiento y de replicación de las adecuaciones en los sistemas locales, lo que implicaba en algunas oportunidades demoras para su disponibilidad.

Ante esta situación se da comienzo a una etapa de transformación que consiste en un proceso de homogeneización de los sistemas locales mediante la implementación de un nuevo sistema SIDIF Local Unificado.

Este proceso, que aún se está llevando a cabo, implica reemplazar los sistemas locales existentes en distintos organismos por una versión unificada con mejoras en cuanto a la provisión de información confiable para la alta gerencia de las entidades, el ajuste de procedimientos de compras, presupuesto, contabilidad y tesorería y la reducción de aplicativos complementarios requeridos para soportar la gestión.

Habiendo cumplido esta primera etapa de transformación que ha introducido importantes mejoras a través del SLU, se plantea un nuevo desafío, la denominada segunda transformación que se materializa con la extensión del alcance y renovación tecnológica con el desarrollo del sistema **e-Sidif**.

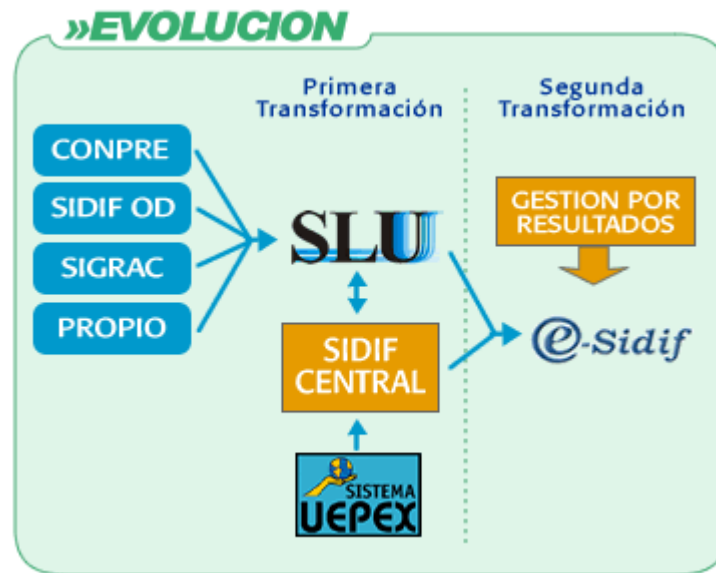


Figura 2 – Evolución de eSidif

Es este sistema el que busca mejorar la calidad del SIDIF incorporando las innovaciones y posibilidades que en materia de tecnología de comunicaciones se han producido en la última década, aprovechando la flexibilidad y reducción de costos que posibilita la modernización, introduciendo cambios que sirvan como base para una mayor orientación de la gestión pública a resultados y una ampliación del alcance y vinculación del sistema.

ESidif es un sistema Enterprise de gran tamaño, como consecuencia posee las características inherentes a este tipo de sistemas. Además de los requerimientos funcionales y procesos que se deben cubrir, adquieren mucha relevancia los requerimientos no funcionales que también deben ser considerados. Por tal motivo, entre muchos de los requerimientos no funcionales, se debe cumplir con un despliegue ágil, seguro y eficiente que permita entregar nueva funcionalidad a los usuarios en el menor tiempo posible y sin afectar la operatoria normal del sistema. Por esta razón eSidif es utilizado como caso de estudio de esta tesis.

3.2. Características del proyecto

A continuación, se mencionan algunas características del proyecto e-Sidif:

- Posee una base de datos única.
- Provee de gestión, registro y control simultáneos.

- Presenta mejoras en las buenas prácticas de administración financiera.
- Beneficio de ser el único sistema de administración financiera.
- Es de fácil uso.
- Incorpora seguridad y auditoría.
- Se tiene la posibilidad de acceder a distintas visiones de la información.
- Autonomía del usuario.
- Provee descentralización operativa.
- Presenta facilidad para los usuarios a la hora de la toma de decisiones.

Por último, otra característica importante a tener en cuenta del sistema eSidif es la forma o estrategia de despliegue del mismo. Esta estrategia se basa en la puesta en producción de manera incremental. Esto es, a medida que se desarrollan los módulos.

Cabe mencionar que se eligió esta estrategia debido a las fuertes demandas de las autoridades para minimizar el impacto de las nuevas tecnologías en los usuarios.

Pero también es necesario definir un proceso repetible para la instalación de la aplicación. Este proceso permitirá automatizar, con herramientas, la instalación del sistema en los distintos ambientes en etapas de desarrollo y producción.

Durante la etapa de desarrollo, se requieren múltiples ambientes, a saber:

- Ambiente de Desarrollo
- Ambiente de Aceptación de los usuarios
- Ambiente de Capacitadores
- Ambiente de Prueba de carga

A su vez en la etapa de producción, se requieren dos ambientes:

- Ambiente de Preproducción
- Ambiente de Producción

El proceso no sólo deberá definir las actividades relacionadas con el deploy de la aplicación sino también pautas que deben ser cumplidas durante el desarrollo.

Este proceso contempla la automatización de tareas sobre la base datos y tareas de configuración de datos de instalación. Para que la automatización sea posible, el equipo de desarrollo debe cumplir con estas pautas preestablecidas. Para automatizar el proceso se debe disponer de herramientas que faciliten la tarea de deploy y registro de logs para su posterior verificación. Adicionalmente, para atender las emergencias de la instalación, se debe desarrollar un proceso que permita realizar un hot deploy manteniendo la disponibilidad del sistema en producción.

Conformación de los grupos de trabajo

Este apartado tiene como fin explicar cómo está conformado el proyecto en cuanto a los grupos de trabajo, desde los niveles más bajos como Diseño y Desarrollo, hasta los más altos como la Coordinación General.

El proyecto para el cual trabajamos eSidif depende de la Dirección General de Sistemas Informáticos de la Administración Financiera (DGSIAF). Cumpliendo con las siguientes responsabilidades y funciones:

Responsabilidad Primaria

- Entender en todo lo relacionado con la planificación, integración, seguridad y administración de la plataforma informática y de comunicaciones, el desarrollo de sistemas, la capacitación, apoyo a usuarios y asesoría informática para la SECRETARÍA DE HACIENDA y unidades organizacionales, en todo lo relacionado con el Sistema de Administración Financiera, con vistas a lograr eficacia y eficiencia en el procesamiento de la información.
- Entender en la optimización y evolución de la plataforma informática y los sistemas vinculados.

Acciones

1. Formular la estrategia y planes informáticos a corto y mediano plazo, relacionados con el Sistema de Administración Financiera.
2. Asistir y asesorar en cuestiones de índole informática a las autoridades, a responsables del sistema de los distintos sectores y usuarios finales del Sistema de Administración Financiera.

3. Coordinar el diseño y desarrollo de las aplicaciones de administración financiera y sistemas relacionados, así como la definición de metodologías y normas de ingeniería de sistemas y documentación, la selección de la plataforma informática y el ambiente de desarrollo y ejecución de los Sistemas de Administración Financiera.
4. Coordinar la ejecución de acciones destinadas a asegurar el correcto funcionamiento del equipamiento informático central y su vinculación con los de las unidades del Sector Público Nacional relacionados con el Sistema de Administración Financiera. Coordinar el tratamiento de los requerimientos de desarrollo y modificación de los sistemas originados en los organismos rectores de la SECRETARÍA DE HACIENDA.
5. Coordinar las acciones necesarias para asegurar la calidad de los Sistemas de Administración Financiera, validando que los requerimientos de los usuarios, tanto de los órganos rectores como de los Servicios Administrativo-Financieros (SAF), se hayan implementado correctamente.
6. Intervenir en los distintos aspectos relacionados con la plataforma de comunicaciones que vincula los sistemas centrales con los Servicios Administrativo-Financieros del Sector Público Nacional.
7. Formular y coordinar el plan de trabajo “Desarrollo Informático de la Administración Financiera” de cada período anual, teniendo en cuenta las prioridades definidas por la SUBSECRETARÍA DE PRESUPUESTO.
8. Intervenir en la definición y control de las tareas desarrolladas por proveedores externos de sistemas informáticos y servicios relacionados con el Sistema de Administración Financiera.
9. Coordinar la definición del equipamiento y servicios informáticos a incorporar, para dar soporte al Sistema de Administración Financiera, e intervenir en los procesos de contratación.
10. Formular el presupuesto anual de la Dirección General de Sistemas Informáticos de Administración Financiera, para su elevación a la SUBSECRETARÍA DE PRESUPUESTO.
11. Coordinar las acciones pertinentes para garantizar el acceso a los recursos de la plataforma informática y de comunicaciones según los procedimientos de seguridad definidos, en el ámbito del MINISTERIO DE HACIENDA Y FINANZAS PÚBLICAS.

- 12. Asegurar el cumplimiento de las políticas y estándares que se determinen para la Administración Pública Nacional.
- 13. Promover la interoperabilidad entre los distintos componentes y organismos usuarios del sistema.
- 14. Impulsar la incorporación de herramientas web.
- 15. Promover la transparencia de la información.

La siguiente imagen muestra el organigrama general de la Dirección General de Sistemas Informáticos de la Administración Financiera, con las distintas áreas de direcciones y coordinación, los cuales realizan diferentes tareas. Aproximadamente participan en las distintas tareas y actividades 232 personas.

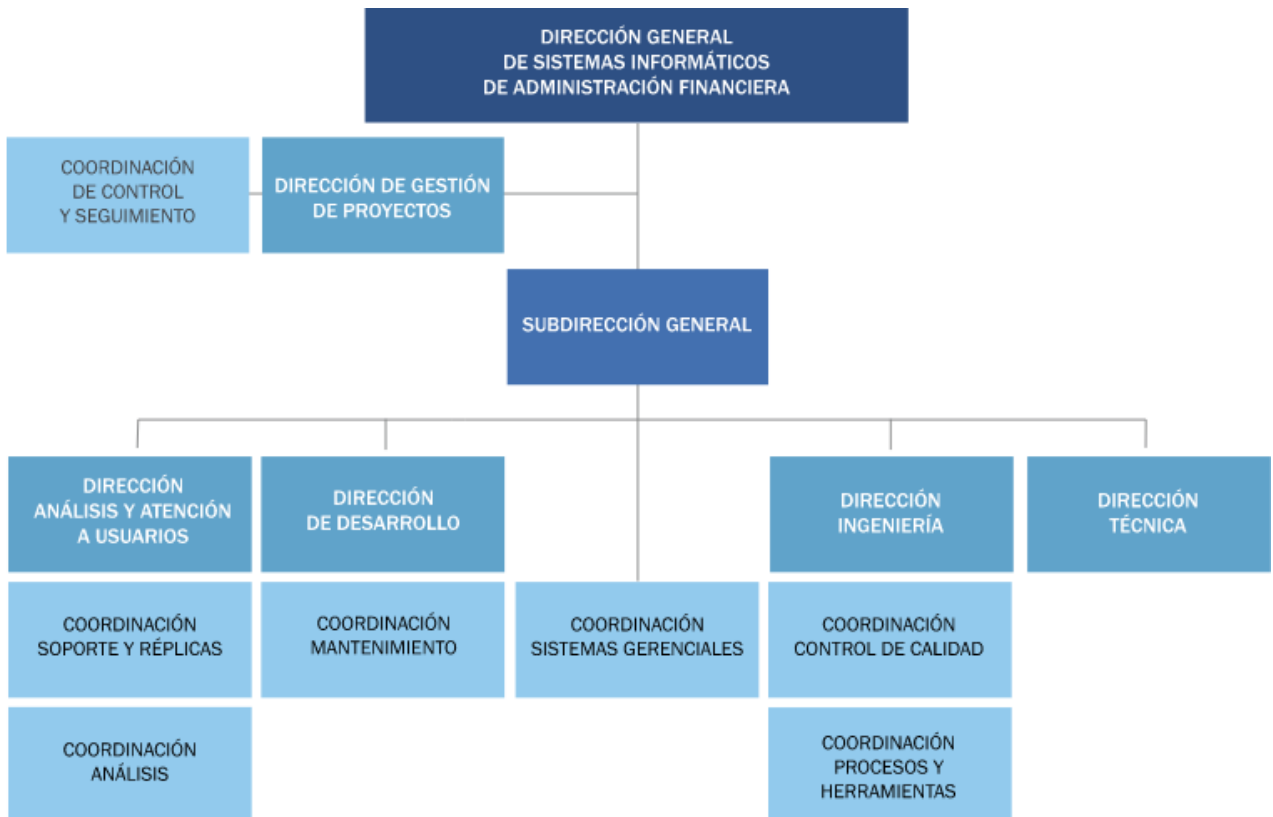


Figura 3 – Organigrama DGSIAF

A continuación, se explicará brevemente qué tipo de tareas realiza cada uno, así como también se detallará cada área de acuerdo a sus funciones y conformación.

Comenzando por la parte inferior izquierda, se encuentra la Dirección Análisis y Atención a Usuarios. Dependiendo dos áreas de coordinación, dentro del área de Coordinación de Análisis se encuentra el grupo de **Análisis**, su misión es la de interpretar los requerimientos del usuario y realizar la concepción de la solución plasmándola en los modelos definidos en el proceso mediante el lenguaje de modelado UML.

Entre sus actividades principales figuran:

- Interpretación de requerimientos del usuario.
- Modelar requerimientos en los diagramas de Casos de Uso correspondientes.
- Estimación del tamaño y esfuerzo de cada negocio.
- Implementación de los modelos UML definidos en la metodología, ya sean, MDR, MCU, MRN, MCA.
- Definición de la estrategia y documentación del análisis de convivencia y migración.
- Generación de la documentación del despliegue.
- Revisación y control de los casos de prueba, y modelos de bases de datos desarrollados.

En el área de Coordinación Soporte y Replicas, se encuentra el grupo de **Réplicas**, que tiene distintas funciones de acuerdo a las etapas en que se encuentre el proyecto.

Cuando el proyecto está en pasaje a producción, se encarga de la realización de demos y capacitaciones de los usuarios hasta que alcancen autonomía en el uso de la aplicación.

En cambio, cuando el proyecto se encuentra durante las etapas de desarrollo, realiza la participación en talleres de definición funcional y reuniones de avance y evaluación de negocios implementados y a implementar.

Los grupos de **Diseño y Desarrollo + Convivencia** (en conjunción, es decir, DyD más Convivencia) es una de las más numerosas, dependiendo de la Dirección de Desarrollo entre sus actividades se destacan:

- Diseño y Construcción de artefactos.
- Realización de ejecutables para los usuarios.

En particular, el área de Diseño y Desarrollo, tiene como misión materializar los requerimientos del usuario en artefactos visibles y ejecutables por el usuario.

Entre sus funciones se destacan:

- Modelado y construcción de componentes para ejecutar la funcionalidad del sistema.
- Construcción de los algoritmos de convivencia entre sistemas legados desarrollados por el equipo de convivencia.

El grupo de **Arquitectura Aplicativa** su misión es definir la arquitectura de software del eSidif y la plataforma de desarrollo sobre la cual se construye el producto. Además, se encarga de proveer de actualización tecnológica de la plataforma, y de mantenimiento permanente de la misma, así como también de brindar soporte a los demás equipos.

Entre sus funciones principales se encuentran:

- Definición de la arquitectura de software.
- Gestión de la plataforma de desarrollo, que incumbe desde la definición, diseño, implementación y mantenimiento de la misma.
- Evaluación de las nuevas tecnologías que pueden ser de utilidad para la plataforma.
- Realización de manuales de uso para cada uno de los componentes de la plataforma.

Existe el grupo de **GOD (Grupos de Orientación del diseño)**. Es un grupo pequeño, pero también tiene actividades interesantes. Entre ellas se destacan:

Optimización de los productos, tanto intermedio como final.

- Promover la reutilización de código normalizado.
- Fomentar la reutilización de componentes gráficos comunes.

- Participación en los procesos de aseguramiento de calidad de modelado de Análisis y Diseño.
- Colaboración con el área de Arquitectura para implementar componentes generales.

El grupo de **mantenimiento**, dependiendo del área de Coordinación Mantenimiento. Ya sea, mantenimiento del eSidif como mantenimiento del SLU/UI (SIDIF Local Unificado/Unidad Informática). Tiene como objetivo, tomar los requerimientos de los usuarios de las aplicaciones que se encuentran en producción, los cuales pueden ser una nueva funcionalidad o de corrección. Entre sus funciones principales se encuentran:

- Interpretación y modelado de modificaciones de los requerimientos del usuario.
- Modificación de los ejecutables.
- Participación en las reuniones de priorización de los requerimientos.
- Participación en los emprendimientos de mejoras de los procesos.

Otro grupo de trabajo es el de BI (**Business Intelligence**), que depende del área de Coordinación de Sistemas Gerenciales. Este equipo se encarga de interpretar requerimientos de niveles ejecutivos e implementar soluciones brindando información para acceder, analizar y reportar contra los datos que más frecuentemente residen en Datawarehouse, Datamarts y almacenes de datos operacionales.

Dentro de la Dirección Ingeniería, está área de Coordinación de procesos y herramientas que se encuentra el grupo de **Ingeniería**, entre los objetivos de esta área podemos nombrar el de definir el proceso de desarrollo del eSidif, así como también el de proveer al equipo del proyecto un ambiente estándar de herramientas que soporten el proceso de desarrollo definido.

Entre sus actividades se destacan:

- Definición de procesos de desarrollo.
- Definición de guías de trabajo.
- Proveimiento de un ambiente estándar.
- Aseguramiento del cumplimiento de Calidad.
- Investigación de las disciplinas del proceso Rup, adaptándolas al proyecto y definiendo las actividades, roles y artefactos.

- Estandarización de cada tipo de artefacto: templates, Ejemplos, Guías de buenas prácticas, Material de capacitación, etc.
- Gestión de Control de calidad de los artefactos generados.
- Elaboración de guías de administración, instalación y uso de herramientas.
- Capacitación y soporte al equipo de desarrollo, a través de sus disciplinas, artefactos y herramientas de soporte.
- Investigación y adaptación de prácticas de procesos ágiles al proyecto.

Dentro del área de Coordinación Control de Calidad, se encuentra el grupo de **Testing**, es un área que cuenta con un gran cantidad de integrantes, (incluidos el coordinador y los dos subcoordinadores). Entre sus actividades se destacan:

- Control de la calidad de los ejecutables.
- Verificación del cumplimiento de los requerimientos del usuario, esto es, a través de casos de prueba.

Dentro de la Dirección Técnica, existe varios grupos que como su palabra lo indica realizan tareas técnicas, como el grupo de **Soporte, Redes, Unix** entre otros, entre sus funciones incluye monitorear la capa de aplicación para garantizar el buen funcionamiento aplicativo.

En el área de Dirección de Gestión de Proyectos, encontramos el **equipo de despliegue** que es responsable de la planificación de los despliegues, de la coordinación de los distintos equipos involucrados, como así también de la ejecución y control de los despliegues.

Dentro de los niveles más alto se encuentra el área de Coordinación de Control y Seguimiento, encontramos el equipo de PMO (Project Management Office).El objetivo de **PMO** es ejecutar con los distintos coordinadores del proyecto, los aspectos específicos relacionados con la gestión, haciendo hincapié en la planificación y seguimiento del proyecto. Entre sus actividades principales encontramos:

- Gestión de la planificación.
- Evaluación y proposiciones de mitigación.
- Gestión del Seguimiento.
- Gestión de la contratación de recursos.

- Gestión y seguimiento de riesgos.
- Gestión del despliegue.

El área de Coordinación general que depende de la Subdirección General, se encarga (entre otras tantas actividades) de elaborar planes para el cumplimiento de los objetivos de las SSP (Sub Secretaria de Presupuesto).

3.3. Proceso de despliegue en eSidif

3.3.1. Evolución histórica

En el inicio del proyecto, los despliegues se plantearon siguiendo los lineamientos generales de la metodología RUP. La labor principal consistía en realizar un relevamiento y un posterior análisis de todas las actividades que debían llevarse a cabo, en el marco de la puesta en producción de un determinado producto. Lo relevado se volcaba detalladamente en forma de tareas en un cronograma (archivo en MS Project). El cronograma hacía referencia a los diferentes artefactos (planillas de cálculo, documentos, listados, etc.) utilizados para referenciar o detallar las diferentes tareas a realizar.

La ejecución del despliegue consistía en ir enviando mails a las áreas involucradas con las tareas que debían realizar. A su vez, vía mail o telefónicamente, se recibían los avisos de finalización de las tareas, y de ese modo se analizaba y controlaba el avance del despliegue.

La complejidad y el tamaño de los despliegues era grande y por consiguiente los cronogramas también. Esto hacía complejo el seguimiento de las tareas y por ende se invertía mucho tiempo en la gestión del despliegue. La gestión era llevada adelante por una sola persona y las tareas eran ejecutadas por las personas asignadas de cada área involucrada. En la figura x se puede ver a modo de ejemplo un fragmento de un cronograma, si bien en la imagen se visualizan 36 registros el mismo posee 167.

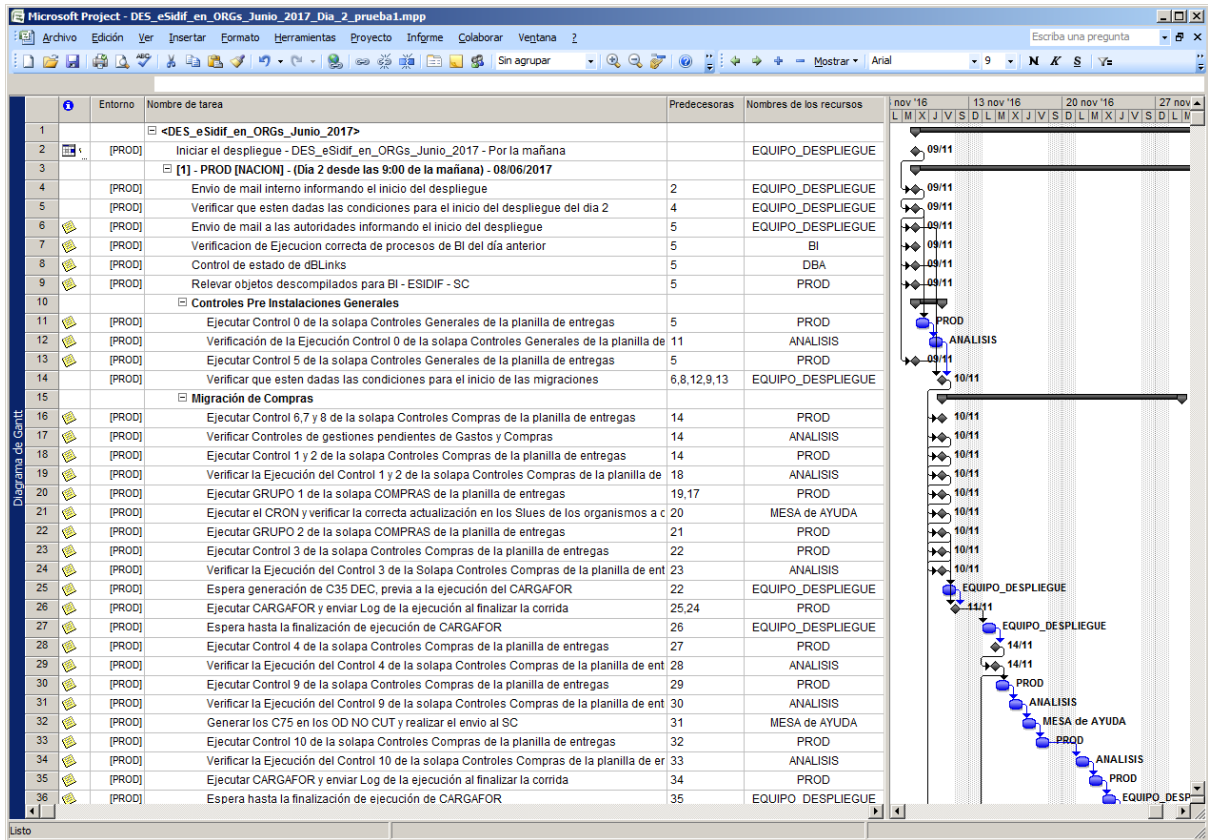


Figura 4 – Cronograma MS-Project

Más adelante se generó, a partir del cronograma, un diagrama en papel para poder visualizar de manera más amigable las dependencias y el paralelismo entre las tareas. Esto aportó una mejora significativa, pero como contrapartida, los diagramas eran muy grandes y difíciles de mantener.

Continuando con la revisión y mejora de la actividad, por un lado, se conformó un equipo de despliegue para abarcar toda la disciplina y, por otro lado, se decidió llevar adelante el proceso de despliegue, implementando la metodología ágil scrum.

Tomando como punto de partida la convivencia de las metodologías mencionada antes (RUP y Agiles), se trabajó en la adaptación de la disciplina de despliegue realizando los ajustes necesarios, sin perder la rigurosidad que detalla RUP, pero adaptándose a los fundamentos ágiles.

A partir de esto último, se trabajó en un marco teórico para la actividad y se llegó a la división en sprints, en donde se detallaron las tareas necesarias para la construcción y ejecución de los diferentes despliegues planificados en el año, logrando de esa manera facilitar la división y ejecución de las tareas generales.

Paralelamente a esto, fue surgiendo en la organización, la necesidad de agregar nuevos despliegues de menor tamaño, incrementando la cantidad total de los mismos. El cambio ágil permitió hacer frente a los nuevos requerimientos. En el punto 3.3.2 del presente capítulo se detalla cómo se decidió aplicar las ideas de scrum en el proceso de despliegue.

A la mejora en términos teóricos mencionada antes, se sumó una mejora en términos de la ejecución del despliegue. Se realizó la implementación de una herramienta que permitiera agilizar la gestión de las tareas, notificando a las áreas involucradas de manera automática, que pueden ejecutar sus tareas y a medida que se fueren realizando “disparar” las tareas que cumplen todas las dependencias de manera automática.

Aparece de esta manera la primera versión de PEP (Plataforma de Ejecución de Procesos), una aplicación cliente servidor, en la cual se utiliza un motor de BPM (business process management) concretamente JBPM [29].

Con el uso de P.E.P. se lograron importantes mejoras en cuanto a la gestión del proceso de despliegue. En el Capítulo 4, se detallarán los logros de la primera versión de P.E.P., como así también los problemas y limitaciones que sumado a nuevos requerimientos, condujeron a la creación de PEP2, siendo esta una de las motivaciones más importante del presente trabajo.

3.3.2. Organización del despliegue ágil

En el presente punto detallaremos el fundamento que permitió aplicar ideas de la metodología SCRUM en el proceso de despliegue, con el objetivo de agilizar el proceso.

Contexto

Dentro de la metodología RUP, la disciplina de despliegue se encarga de que el producto construido sea transferido al ambiente de producción para ser utilizado por el usuario. Para ello, se debe elaborar un plan detallando con una serie de tareas que permitan, mediante su ejecución, dejar disponible el sistema al usuario final.

Si bien resulta difícil relacionar una disciplina surgida dentro de una metodología estructurada como RUP, con los principios que promueven las metodologías ágiles. En el proyecto se combinan ambas haciendo posible que convivan.

Por un lado, la definición de los Artefactos se realiza siguiendo los principios que describe RUP. Por otro lado, para la gestión de procesos, en la mayoría de los equipos (diseño y desarrollo, testing, etc.) se utiliza scrum.

Tomando como punto de partida la convivencia de las metodologías mencionada antes, se trabajó en la adaptación de la disciplina de despliegue realizando los ajustes necesarios, sin perder la rigurosidad que la misma detalla, pero adaptándose a los fundamentos ágiles.

Despliegue

En el desarrollo de un producto siguiendo los lineamientos que define scrum, al final de cada sprint se realiza el despliegue de la funcionalidad implementada al usuario final. De este modo, al finalizar el último sprint se contará con toda la funcionalidad implementada en producción. En este marco, tiene sentido plantear el despliegue como un conjunto de actividades que permita realizar este pasaje al entorno productivo.

Considerando que cada proyecto tendrá diferentes plazos y limitaciones para efectuar los despliegues, es lógico elegir quizás no realizar esta actividad al fin de cada sprint, sino dejarla para cuando se hayan acumulado las suficientes partes del producto que tengan relevancia para desplegarse en un entorno productivo.

Los despliegues en el proyecto esidif, corresponden a la acumulación de porciones de producto que constituyen un hito a desplegar. Dichas porciones del producto, pueden corresponder a diferentes negocios (funcionales) en los cuales diferentes equipos realizan el análisis, el desarrollo y el testing dentro del marco del desarrollo ágil con scrum.

Es decir, varios equipos de diferentes disciplinas interactúan para construir un producto y cada uno de ellos usa scrum. Si a esto le agregamos varios productos desarrollándose en paralelo; nos encontramos con varios equipos de diferentes disciplinas interactuando para construir varios productos en donde además el nivel de acoplamiento puede ir variando. (Podríamos en este punto relacionarlo con el concepto de scrum de scrum, para que nos permita tomar dimensión de la magnitud del producto que se está implementando. Este sería el caso más complejo que deberíamos considerar al momento de realizar las actividades de un despliegue.

Fases del despliegue ágil

Se divide la gestión de un despliegue en 4 sprints, que no necesariamente tienen la misma duración. El objetivo fue justamente poder cerrar cada uno de estos sprints con un producto entregable. A cada sprint se le asignó un nombre: Relevamiento, Construcción, Ejecución y Cierre.

Sprint de relevamiento

Durante este sprint, se debe tomar conocimiento de las características propias que tendrá el despliegue a realizar. Identificar las particularidades, conocer las fechas relevantes y realizar la comunicación a todas las áreas de estas novedades.

Paralelamente se comienza a trabajar en el armado de los artefactos que utilizarán al momento de ejecutar el despliegue y en el acompañamiento de la elaboración de los documentos que se enviarán al usuario.

Al finalizar este sprint se presenta a los involucrados e interesados un documento que detalla las características del despliegue.

Sprint de construcción

Este sprint eventualmente podría iniciarse aún sin que haya terminado el anterior, siendo esta una licencia que se toma para poder adaptar la metodología a las necesidades. Durante esta etapa, se gestiona el envío de avisos a todos los usuarios, indicando que se realizarán tareas de puesta en producción que requerirán realizar la baja del sistema durante un determinado tiempo. Del mismo modo y dado que en algunas ocasiones hay un número considerable de personas involucradas, se revisan y evalúan las acciones para asegurar la disponibilidad de los recursos técnicos necesarios para que se puedan realizar las tareas.

Paralelamente se gestiona una serie de documentos asociados que sirven de presentación del producto que se está desplegando.

Por otro lado y entrando en la construcción propia del despliegue, se realiza la elaboración del cronograma detallado de tareas que se deben llevar a cabo previamente a la fecha de despliegue, las tareas que se realizarán el día del despliegue y las tareas posteriores. También se trabaja en la generación y seguimiento de lo que se debe instalar, mediante la elaboración de una lista de entregas a instalar en el entorno productivo. Esta lista de tareas se referencia desde el cronograma, que es en donde se representa el flujo de control del despliegue.

Sprint de ejecución

En este punto, se realiza la ejecución de las tareas previamente definidas y planificadas, ya sea que estén a cargo del equipo de despliegue o a cargo de otros equipos asignados al despliegue.

Se indica a cada responsable o a cada área lo que debe realizar, en qué momento y bajo qué condiciones. Idealmente antes de la implementación en el entorno productivo se

realiza una simulación de todo el proceso en un entorno similar al productivo. Esto permite repasar y depurar el proceso, realizando los ajustes necesarios para su ejecución.

Sprint de cierre

En este sprint, el equipo de despliegue realiza una retrospectiva donde se repasa lo ocurrido durante todo el proceso de despliegue, tomando estas conclusiones para futuras implementaciones. A su vez también se realizan reuniones con las áreas intervinientes a fin de repasar y sacar conclusiones que sirvan para la mejora continua.

Finalmente se realiza un breve informe de situación sobre el despliegue realizado (o implementación del producto).

Actividades del despliegue ágil

Las principales actividades que deben realizarse en el proyecto, considerando un despliegue de gran funcionalidad son:

- Relevar y consensuar las fechas en que se realizará el despliegue.
- Relevar el alcance que tendrá el despliegue (en cuanto a funcionalidad que incluye, tareas técnicas particulares y cualquier otra consideración especial a tener en cuenta).
- Gestionar la generación de los documentos que permitan al usuario conocer la funcionalidad entregada.
- Planificar la ejecución en los entorno de preproducción
- Colaborar con la planificación del mantenimiento de los entornos de preproducción.
- Planificar los momentos del despliegue en cada una de sus instancias mediante la elaboración de un cronograma con las tareas a realizarse para la puesta en producción. Adicionalmente, participar en la elaboración de los artefactos que acompañan a la ejecución (planilla de entregas a instalar).
- Mantener informado a los integrantes del proyecto sobre las características propias y las novedades del despliegue y fomentar la interacción con ellos.
- Coordinar y Ejecutar el despliegue.

- Acompañar el post despliegue, generando un informe de situación e incidentes derivados de la implementación del producto.
- Realizar reuniones de mejora continua que permitan tener una abierta y clara comunicación con cada una con los integrantes de las áreas involucradas.
- Revisar y mejorar la automatización de los procesos que acompañan al despliegue, tanto en su planificación y ejecución como en las tareas anexas.
- Realizar el acompañamiento en los despliegues no funcionales.

Ejemplo en Agilefant ^[15]

En el proyecto se utiliza la herramienta Agilefant para organizar las tareas aplicando la metodología scrum. Esta herramienta permite definir las tareas que componen el backlog y distribuir las en los diferentes sprints planificados. En las siguientes 3 figuras, se ve un ejemplo de cómo se organiza en agilefant los sprints de un proceso de despliegue.

En la figura 5, se puede visualizar la portada inicial de agilefant, ahí se puede ver a la izquierda, todos los proyectos de despliegue existentes. Al estar posicionados en el clasificador de despliegue, se pueden ver en el centro de la pantalla más información de los despliegues, como por ejemplo las fechas estimadas de inicio y fin.

Luego en la figura 6, se puede ver a la izquierda de la pantalla, los sprint de un despliegue específico. De manera que al estar posicionados en el sprint llamado “relevamiento” podemos ver en el centro de la pantalla, los storys que lo componen. Los storys en agilefant, son agrupamientos de tareas relacionadas por algún motivo.

Finalmente, en la figura 7 se ve el contenido, es decir las tareas del story que está seleccionado (Relevar el alcance del despliegue y planificar).

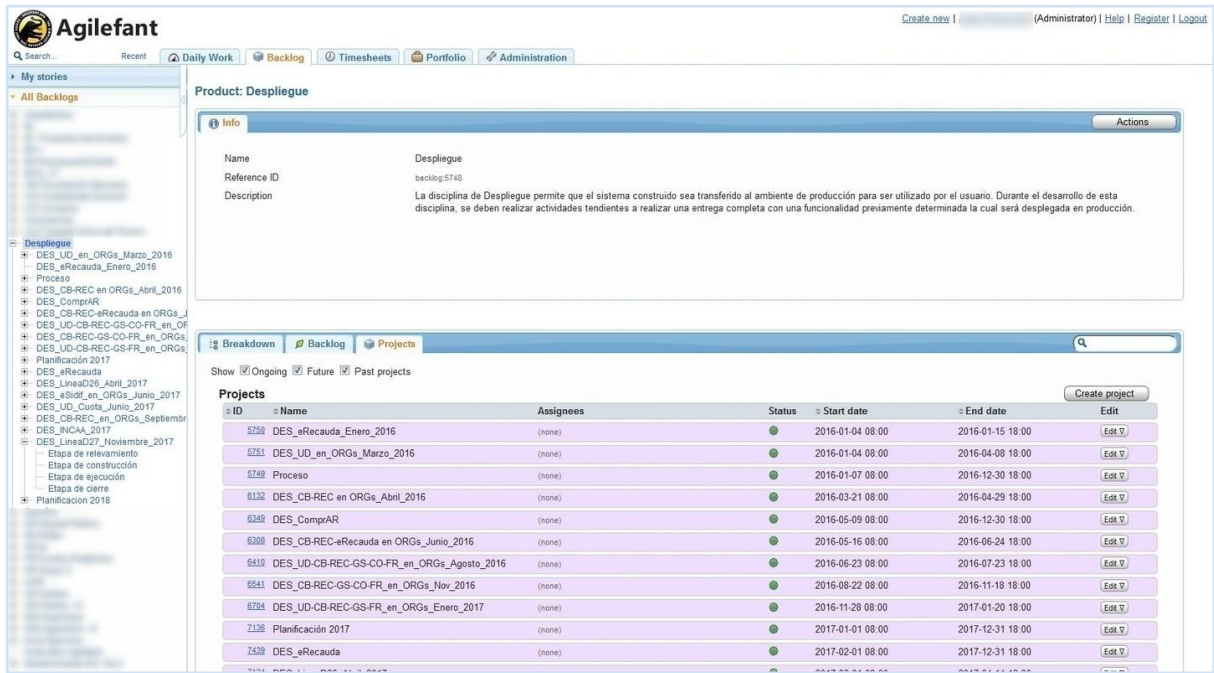
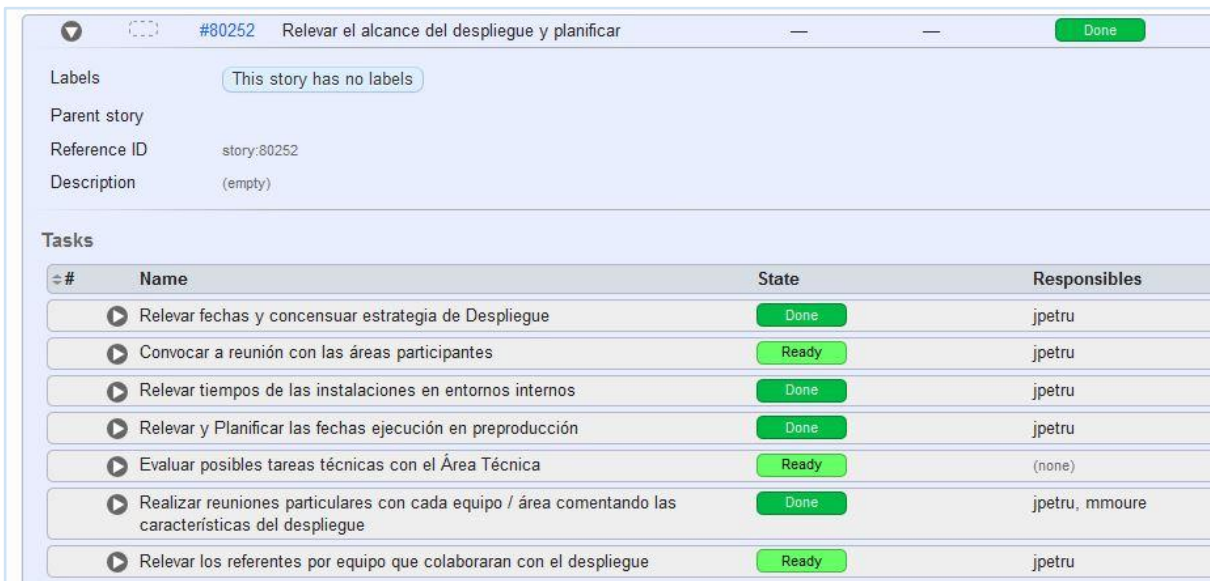


Figura 5 – Vista inicial de agilefant



Figura 6 – Vista de los sprint de un despliegue y los storys de un sprint.



The screenshot shows a Jira story view for the story #80252, titled "Relevar el alcance del despliegue y planificar". The story is marked as "Done". It has no labels, no parent story, and a reference ID of "story:80252". The description is empty. Below the story details, there is a "Tasks" section with a table listing seven tasks. Each task has a play button icon, a name, a state (Done or Ready), and responsible parties.

#	Name	State	Responsibles
▶	Relevar fechas y concensuar estrategia de Despliegue	Done	jpetru
▶	Convocar a reunión con las áreas participantes	Ready	jpetru
▶	Relevar tiempos de las instalaciones en entornos internos	Done	jpetru
▶	Relevar y Planificar las fechas ejecución en preproducción	Done	jpetru
▶	Evaluar posibles tareas técnicas con el Área Técnica	Ready	(none)
▶	Realizar reuniones particulares con cada equipo / área comentando las características del despliegue	Done	jpetru, mmoure
▶	Relevar los referentes por equipo que colaboraran con el despliegue	Ready	jpetru

Figura 7 – Vista de las tareas de un story.

4. Primer prototipo de la herramienta P.E.P.

4.1. Introducción

Los despliegues en forma general responden a una serie de pasos conocidos que se repiten en forma regular y que se van configurando dependiendo de lo que se esté desplegando. Además, los procesos de despliegue se ejecutan por lo menos dos veces; primero en el ambiente de pre-producción a modo de prueba para poder ajustarlo, si es necesario, y finalmente se ejecuta en el ambiente productivo.

Con el objetivo de automatizar las actividades relacionadas con los despliegues y facilitar en general la tarea de seguimiento de los mismos, se encaró el desarrollo de una herramienta que contemplara todos los requerimientos ligados a estas actividades. De esa manera se elaboró la herramienta nombrada con el acrónimo P.E.P. que significa "Plataforma de Ejecución de Procesos".

4.2. Objetivos de la primera versión

Cuando se inició el desarrollo de la primera versión se plantearon como objetivos principales los que se enuncian a continuación:

- Agilizar la gestión y administración del despliegue.
- Mejorar los tiempos de ejecución y coordinación del despliegue.
- Obtener fácilmente información para estimar los tiempos de los futuros despliegues.
- Automatizar el envío de tareas a los usuarios.
- Informar tanto la complejidad de un despliegue como las dependencias de tareas.
- Brindar información on-line del avance del despliegue a todos los participantes del proyecto que requieran tener feedback de la ejecución.

4.3. Descripción funcional

P.E.P. es una herramienta web que funciona como interface a partir de la cual los usuarios pueden interactuar con las definiciones de los procesos y sus correspondientes

instancias. Actúa como una bandeja de actividades en la cual van apareciendo las tareas asignadas a cada usuario para que éste pueda llevar adelante una determinada tarea y de esta manera continúe el proceso en cuestión.

4.3.1. Casos de uso

A los fines de mostrar las principales funcionalidades de P.E.P. especificamos los casos de uso mediante el diagrama correspondiente, indicando también los actores que intervienen y el desarrollo de cada uno de los casos de uso.

Actores

Administrador: Requiere autenticación. Puede generar un proceso nuevo y administrarlo.

Usuario ejecutor de tareas: Requiere autenticación. Recibe una tarea por mail y puede iniciarla, darla por cumplida (ok), darla por cumplida con error (indicando un motivo) o marcarla como errónea (deteniendo allí el proceso).

Usuario iniciador de procesos: Requiere autenticación. Puede iniciar cualquier proceso para el cual fue habilitado por el administrador.

Público: Usuario que no requiere autenticación. Tiene acceso a la vista de solo consulta del avance de las tareas del despliegue (dashboard).

Diagrama de casos de uso

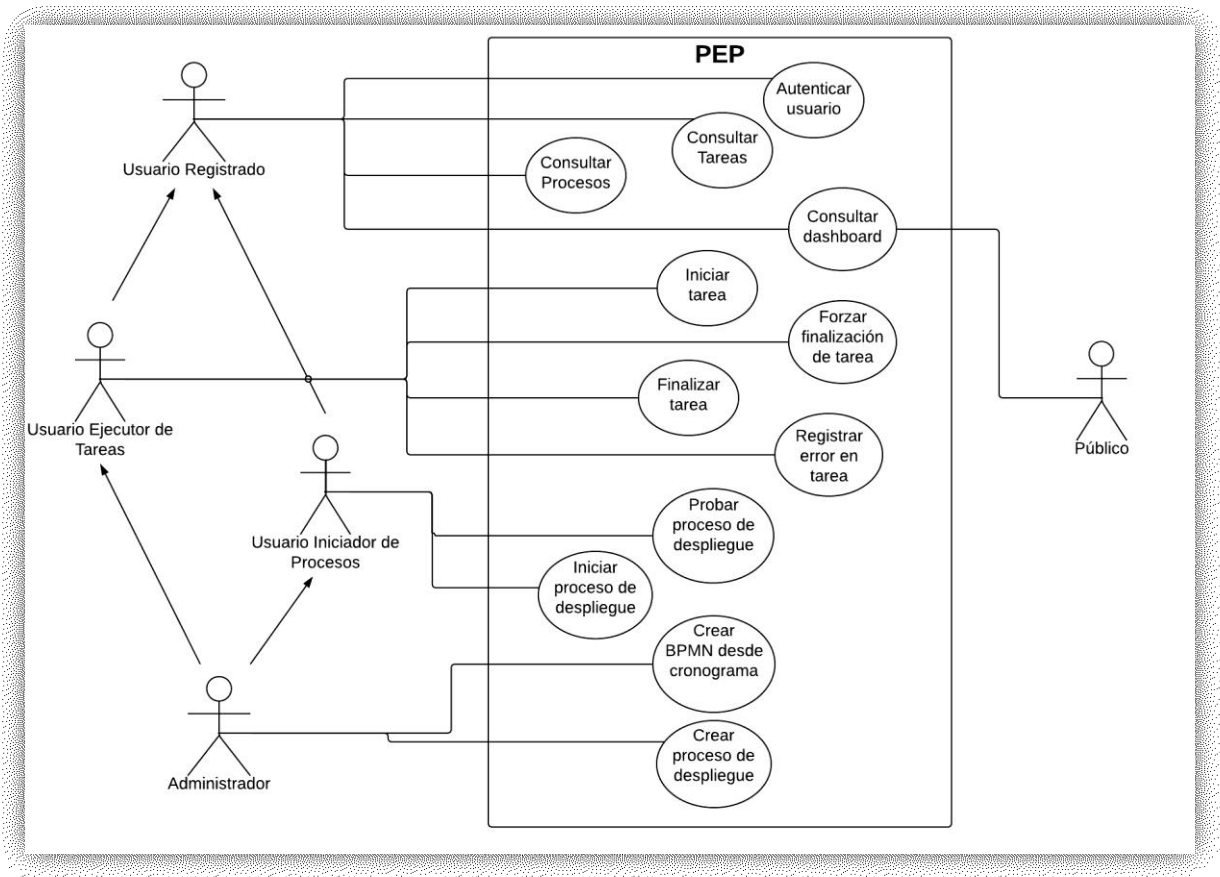


Figura 8 – Diagrama de casos de uso.

Especificación de los casos de uso

Para simplificar la lectura se nombrará como “usuario” a todos los actores que intervienen en el CU. Además, cuando hablamos de procesos, nos estamos refiriendo a procesos de despliegue.

1 - caso de uso *Autenticar usuario*.

Datos Generales	
<i>Descripción Contextual</i>	Permite que un usuario se autentique para poder iniciar una sesión en el sistema.
<i>Actores Involucrados</i>	▪ Usuario Administrador, usuario ejecutor de tareas y usuario iniciador de procesos.
Relaciones	

<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	▪ El usuario debe existir en el Sistema.
<i>Post-condiciones</i>	▪ El usuario, inicia una sesión en el sistema.
<i>Observaciones</i>	▪ Ninguna

Tabla 3 – CU autenticar usuario, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
	<ol style="list-style-type: none"> 1. El usuario ingresa el nombre de usuario y contraseña. 2. El <i>Sistema</i> verifica la existencia del usuario. 3. El <i>Sistema</i> verifica que la contraseña sea correcta. 4. El <i>Sistema</i> inicia una sesión y muestra la pantalla principal.
Curso alternativo – No existe el usuario ingresado.	
	<ol style="list-style-type: none"> 2.1 El <i>Sistema</i> no puede verificar la existencia del usuario. 2.2 El <i>Sistema</i> informa al usuario ingresado es incorrecto.
Curso alternativo – Contraseña incorrecta.	
	<ol style="list-style-type: none"> 3.1 El <i>Sistema</i> verifica que la contraseña ingresada es incorrecta. 3.2 El <i>Sistema</i> informa al usuario que la contraseña es incorrecta.

Tabla 4 – CU autenticar usuario, desarrollo.

2 - Caso de uso **Crear BPMN desde cronograma.**

Datos Generales	
<i>Descripción Contextual</i>	Este caso de uso permite a un usuario administrador, generar un BPMN [18] en formato xml, a partir de la información de un cronograma MSP [16].

<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario Administrador
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> • El cronograma debe ser un archivo MSP[16] con un formato preestablecido. • Es usuario inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> • El usuario, obtiene un BPMN [18] en formato xml.
<i>Observaciones</i>	

Tabla 5 – CU Crear BPMN desde cronograma, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. Un usuario selecciona un archivo MSP[16] desde el file system. 2. El Sistema verifica la validez del archivo. 3. El Sistema imprime en pantalla el BPMN [18] en formato xml.
Curso alternativo – El archivo no es válido.
<ol style="list-style-type: none"> 2.1. El Sistema determina que el archivo no es válido. 2.2. El Sistema informa al usuario que debe elegir un archivo MSP[16] válido.

Tabla 6 – CU Crear BPMN desde cronograma, desarrollo.

3 - caso de uso Crear proceso de despliegue.

Datos Generales	
<i>Descripción Contextual</i>	Permite crear un proceso.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario Administrador
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	

Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> El usuario inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> El sistema crea un nuevo proceso.
<i>Observaciones</i>	<ul style="list-style-type: none"> El JAR generado a partir de Kie [17], contiene la representación de un proceso previamente importado desde kie[17] mediante el archivo xml que se creó en el CU Crear BPMN desde cronograma.

Tabla 7 – CU Crear proceso de despliegue, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
<ol style="list-style-type: none"> Un usuario selecciona la opción de crear un nuevo proceso. El Sistema muestra la pantalla donde debe ingresar los datos necesarios. El usuario ingresa los datos para poder crear el proceso: <ol style="list-style-type: none"> Ingresar los datos generales. Selecciona un archivo MSP [16]. Selecciona un archivo JAR generado a partir del Kie [17]. Selecciona los usuarios habilitados para iniciar el proceso. El Sistema informa que se ha creado el proceso correspondiente con éxito. 	
Curso alternativo – No están cargados los archivos.	
<ol style="list-style-type: none"> El Sistema verifica que no se han cargado los archivos mencionados en los incisos b y c. El Sistema informa que los archivos son obligatorios para crear el proceso. 	

Tabla 8 – CU Crear proceso de despliegue, desarrollo.

4 - caso de uso Iniciar proceso de despliegue.

Datos Generales	
<i>Descripción Contextual</i>	Mediante este caso de uso, un usuario administrador puede iniciar un proceso de despliegue en modo producción.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> Usuario iniciador de procesos
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> El proceso debe existir en el Sistema. El usuario inició una sesión.

<i>Post-condiciones</i>	▪ El usuario, inicia un proceso de despliegue.
<i>Observaciones</i>	▪ Ninguna

Tabla 9 – CU Iniciar proceso de despliegue, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. Un usuario selecciona el proceso de despliegue que va a iniciar. 2. El usuario inicia el proceso. 3. El sistema envía a todos los participantes un mail con el link para acceder al dashboard correspondiente. 4. El sistema, cambia el estado de la primera tarea a "lista para tomar" y envía mail a los recursos asignados avisando que la pueden iniciar.

Tabla 10 – CU Iniciar proceso de despliegue, desarrollo.

5 - Caso de uso **Probar proceso de despliegue.**

Datos Generales	
<i>Descripción Contextual</i>	Mediante este caso de uso, un usuario administrador puede iniciar un proceso de despliegue en modo de prueba.
<i>Actores Involucrados</i>	▪ Usuario iniciador de procesos
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> ▪ El proceso debe existir en el Sistema. ▪ El usuario inició una sesión.
<i>Post-condiciones</i>	▪ El usuario, inicia un proceso de despliegue en modo de prueba.
<i>Observaciones</i>	▪ Ninguna

Tabla 11 – CU Probar proceso de despliegue, datos generales.

Desarrollo del Caso de Uso
Curso Normal

<ol style="list-style-type: none"> 1. Un usuario selecciona el proceso de despliegue que va a iniciar. 2. El usuario inicia el proceso en modo de prueba. 3. El sistema envía a un mail de prueba con el link para acceder al dashboard correspondiente. 4. El sistema, cambia el estado de la primera tarea a "lista para tomar" y envía mail a un mail de prueba para simular la ejecución del proceso.

Tabla 12 – CU Probar proceso de despliegue, desarrollo.

6 - Caso de uso **Iniciar tarea.**

Datos Generales	
<i>Descripción Contextual</i>	Mediante este caso de uso, se inicia una tarea de un proceso de despliegue iniciado.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> ▪ Usuario ejecutor de tareas.
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> ▪ La tarea debe existir en el Sistema y encontrarse en estado "lista para tomar". ▪ El usuario inició una sesión. ▪ El usuario es uno de los recursos asignados a la tarea. ▪ El proceso al cual pertenece la tarea se encuentra en estado iniciado.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> ▪ El usuario inicia la tarea.
<i>Observaciones</i>	<ul style="list-style-type: none"> ▪ Ninguna

Tabla 13 – CU Iniciar tarea, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. El usuario inicia una tarea. 2. El sistema cambia el estado de la tarea a iniciada. 3. El sistema informa que la tarea ha sido iniciada con éxito y muestra la información de la tarea.

Tabla 14 – CU Iniciar tarea, desarrollo.

7 - Caso de uso **Registrar error en la tarea.**

Datos Generales	
<i>Descripción Contextual</i>	Permite registrar que una tarea tuvo un error que impidió completarse.
<i>Actores Involucrados</i>	▪ Usuario ejecutor de tareas.
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> ▪ La Tarea debe existir en el sistema y encontrarse en estado "iniciada". ▪ El usuario inició una sesión. ▪ El usuario es uno de los recursos asignados a la tarea. ▪ El proceso al cual pertenece la tarea se encuentra en estado iniciado.
<i>Post-condiciones</i>	▪ El usuario notifica que la tarea se encuentra con error.
<i>Observaciones</i>	▪ Ninguna

Tabla 15 – CU Registrar error en la tarea, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
	<ol style="list-style-type: none"> 1. El usuario indica que una tarea está con error. 2. El <i>Sistema</i> solicita una descripción del error. 3. El usuario ingresa el detalle y confirma la operación. 4. El <i>Sistema actualiza el estado de la tarea a "con Error"</i>.

Tabla 16 – CU Registrar error en la tarea, desarrollo.

8 - Caso de uso **Finalizar tarea.**

Datos Generales	
<i>Descripción Contextual</i>	Este caso de uso permite indicar que una tarea ha finalizado.
<i>Actores Involucrados</i>	▪ Usuario ejecutor de tareas
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	

Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> ▪ La Tarea debe existir en el sistema y encontrarse en estado "iniciada". ▪ El usuario inició una sesión. ▪ El usuario es uno de los recursos asignados a la tarea. ▪ El proceso al cual pertenece la tarea se encuentra en estado iniciado.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> ▪ El usuario indica que la tarea ha finalizado.
<i>Observaciones</i>	<ul style="list-style-type: none"> ▪ Ninguna

Tabla 17 – CU Finalizar tarea, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. El usuario indica que la tarea ha finalizado. 2. El Sistema actualiza el estado de la tarea a "finalizada". 3. El sistema verifica si corresponde cambiar el estado de las tareas sucesoras. <ol style="list-style-type: none"> 3.1. El Sistema cambia el estado de las tareas sucesoras a "lista para tomar". 3.2. El sistema notifica a los usuarios que pueden tomar dicha tarea.

Tabla 18 – CU Finalizar tarea, desarrollo.

9 - Caso de uso **Forzar finalización de tarea.**

Datos Generales	
<i>Descripción Contextual</i>	Este caso de uso permite forzar la finalización de una tarea.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> ▪ Usuario ejecutor de tareas
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> ▪ La Tarea debe existir en el sistema y encontrarse en estado "iniciada". ▪ El usuario inició una sesión. ▪ El usuario es uno de los recursos asignados a la tarea. ▪ El proceso al cual pertenece la tarea se encuentra en estado iniciado.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> ▪ El usuario fuerza la finalización de una tarea.
<i>Observaciones</i>	<ul style="list-style-type: none"> ▪ Ninguna

Tabla 19 – CU Forzar finalización de tarea, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
<ol style="list-style-type: none"> 1. El usuario indica que la tarea ha finalizado de manera forzada. 2. El sistema solicita un motivo por el cual se forzó la finalización. 3. El usuario ingresa el detalle del motivo. 4. El sistema actualiza el estado de la tarea a "finalizada". 5. El sistema verifica si corresponde cambiar el estado de las tareas sucesoras. <ol style="list-style-type: none"> 5.1. El sistema cambia el estado de las tareas sucesoras a "lista para tomar". 5.2. El sistema notifica a los usuarios que pueden tomar dicha tarea. 	

Tabla 20 – CU Forzar finalización de tarea, desarrollo.

10 - Caso de uso **Consultar dashboard.**

Datos Generales	
<i>Descripción Contextual</i>	Permite a cualquier usuario consultar el dashboard de un proceso que se esté ejecutando.
<i>Actores Involucrados</i>	▪ Usuario público
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	
<i>Post-condiciones</i>	▪ El usuario visualiza el dashboard del proceso.
<i>Observaciones</i>	▪ Ninguna

Tabla 21 – CU Consultar dashboard, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
<ol style="list-style-type: none"> 1. El usuario solicita visualizar el dashboard. 2. El sistema muestra el dashboard solicitado. 	

Tabla 22 – CU Consultar dashboard, desarrollo.

11 - Caso de uso **Consultar procesos.**

Datos Generales	
<i>Descripción Contextual</i>	Permite consultar los procesos.
<i>Actores Involucrados</i>	▪ Usuario iniciador de procesos y usuario ejecutor de tareas
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	▪ El usuario inició una sesión.
<i>Post-condiciones</i>	▪ El usuario visualiza los procesos.
<i>Observaciones</i>	▪ Ninguna

Tabla 23 – CU Consultar procesos, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. El usuario solicita visualizar los procesos. 2. El sistema verifica el tipo de usuario. <ol style="list-style-type: none"> 2.1. El usuario es iniciador de procesos. <ol style="list-style-type: none"> 2.1.1. El sistema muestra los procesos que puede iniciar. 2.2. El usuario es ejecutor de tareas. <ol style="list-style-type: none"> 2.2.1. El sistema muestra los procesos que contienen tareas que puede ejecutar el usuario.

Tabla 24 – CU Consultar procesos, desarrollo.

12 - Caso de uso **Consultar tareas.**

Datos Generales	
<i>Descripción Contextual</i>	Permite consultar las tareas.
<i>Actores Involucrados</i>	▪ Usuario ejecutor de tareas
Relaciones	
<i>Extiende</i>	

<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	▪ El usuario inició una sesión.
<i>Post-condiciones</i>	▪ El usuario visualiza las tareas.
<i>Observaciones</i>	▪ Ninguna

Tabla 25 – CU Consultar tareas, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. El usuario solicita consultar las tareas. 2. El Sistema muestra las tareas que el usuario puede gestionar.

Tabla 26 – CU Consultar tareas, desarrollo.

4.3.2. Vistas

En las siguientes imágenes, se puede ver la aplicación P.E.P. y los puntos de entrada a los casos de uso mencionados en el punto anterior.

CU autenticar usuario

Al hacer click en el botón enviar se efectúa la autenticación correspondiente.

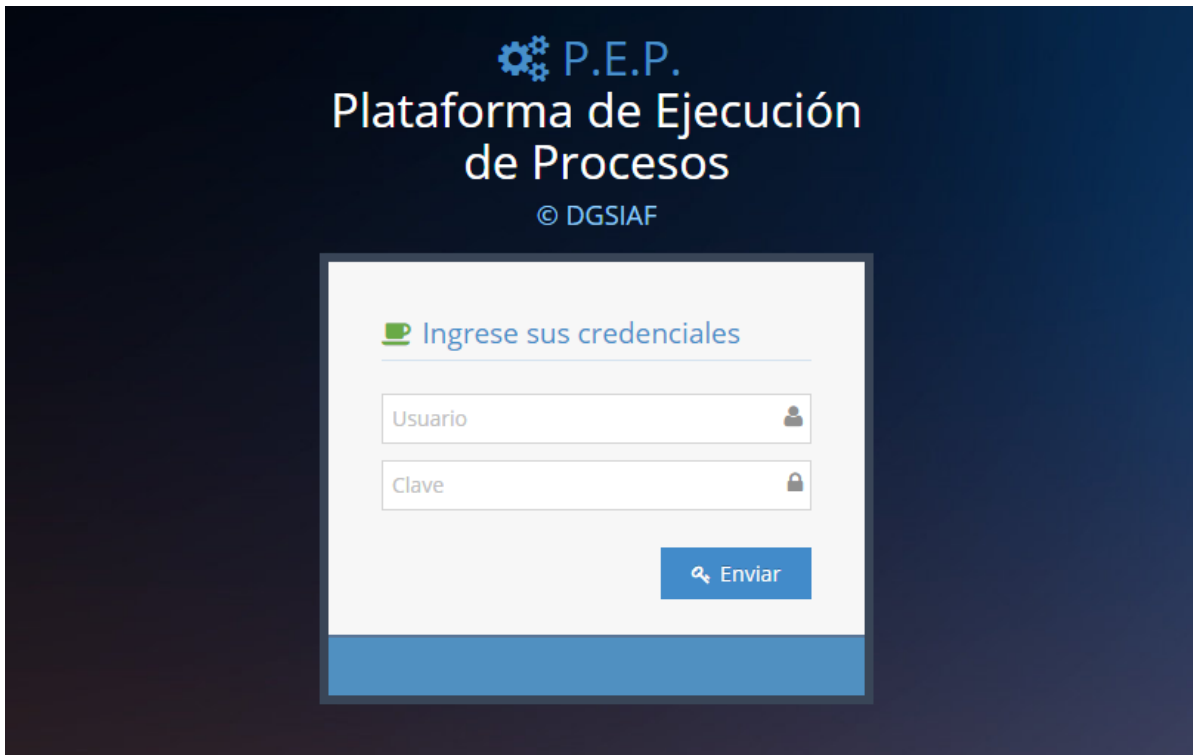


Figura 9 – Vista de login.

CU crear BPMN desde cronograma

Al hacer click en el botón enviar, se confirma la creación del BPMN [18]. Como resultado, el xml se muestra en el cuadro "contenido de archivo". El botón restablecer, sirve para limpiar la información cargada, tanto el archivo como el actor principal.

The screenshot shows the P.E.P. web application interface. The top navigation bar includes 'Dashboard' and 'Procesos > Crear BPMN a partir de un cronograma'. A left sidebar contains a menu with options: 'Iniciar despliegue', 'Alta proceso de despliegue', 'Crear BPMN desde cronograma', 'Ver mis procesos', 'Tareas', and 'Usuarios'. The main content area is titled 'Archivo del cronograma (extensión ".mpp")' and features a file selection interface with a 'Seleccionar archivo' button and the text 'No se eligió archivo'. Below this is a text input field for 'Actor principal' containing the value 'username'. A horizontal bar contains two buttons: 'Restablecer' and 'Enviar'. The lower section is titled 'Archivo BPMN (copie el contenido en un archivo con extensión ".bpmn2")' and includes a label 'Contenido del archivo' next to a large, empty text area for pasting content.

Figura 10 – Vista de creación de BPMN desde cronograma.

CU crear proceso de despliegue

Una vez cargada la información se confirma con click en el botón "enviar". El botón restablecer, sirve para limpiar toda la información cargada.

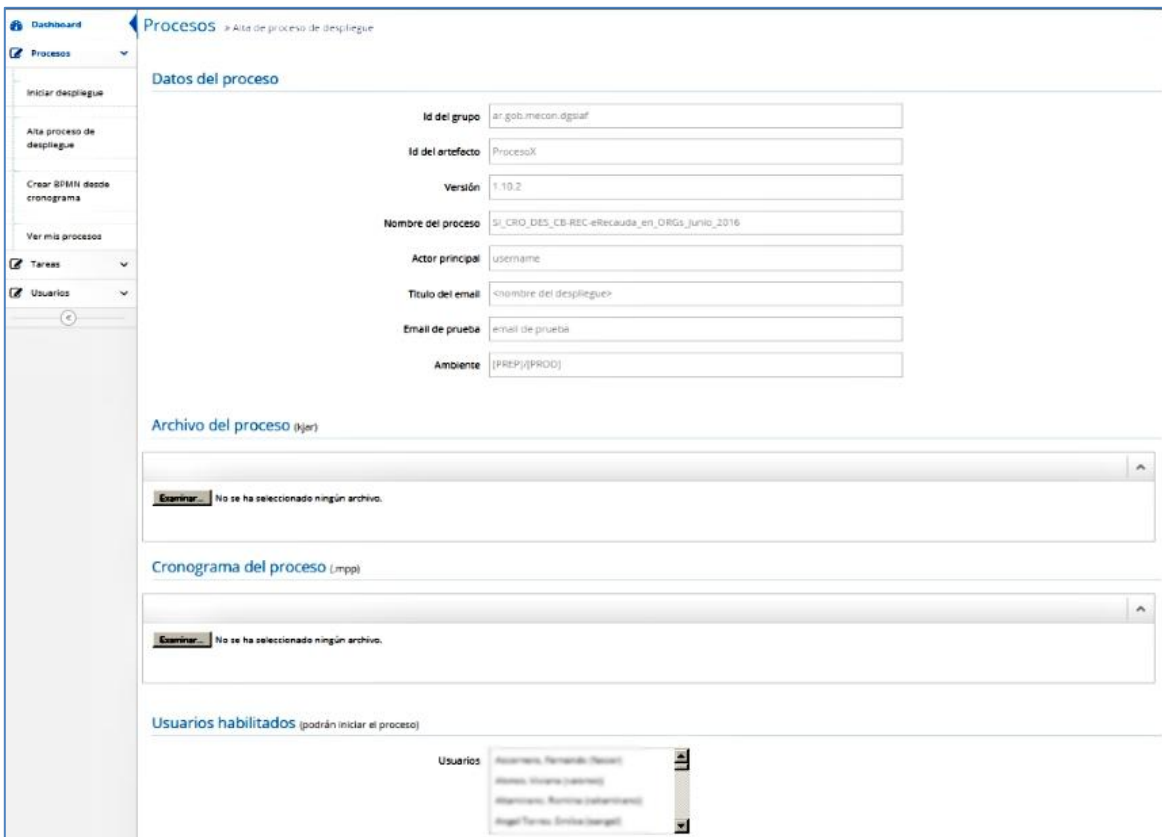


Figura 11 – Vista de creación de procesos.

CU iniciar proceso de despliegue y CU probar proceso de despliegue

En la siguiente vista, se ve el llamado a los dos casos de uso botón "Iniciar" y botón "Probar" previa selección del proceso correspondiente. Al igual que en las vistas anteriores con el botón restablecer, se limpia la información cargada en el combo.



Figura 12 – Vista de Inicio y prueba de proceso.

En ambos casos de usos, se envía un mail indicando el link donde se puede acceder al dashboard público, con el formato que se en la siguiente imagen.

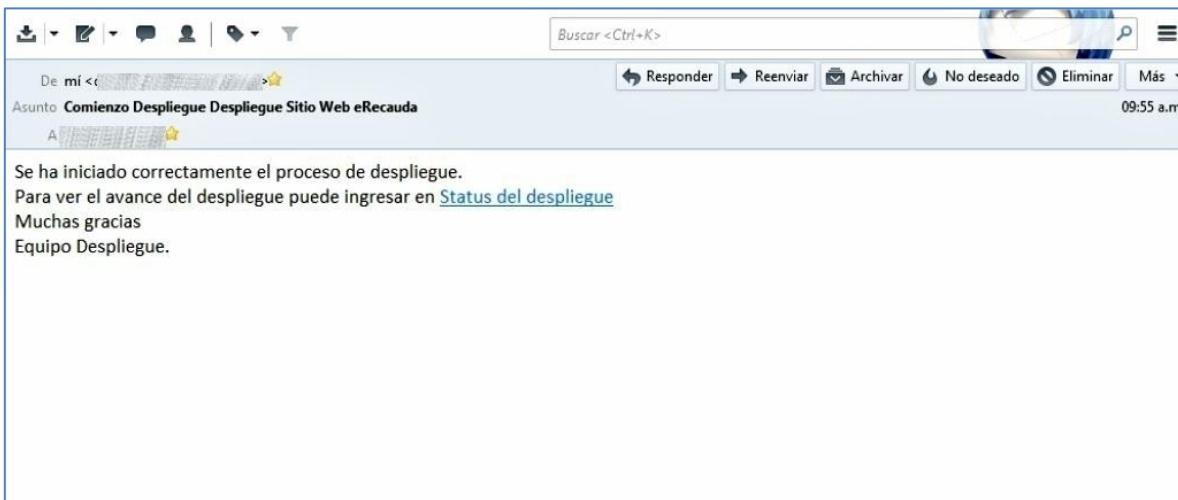


Figura 13 – Ejemplo de mail que se envía cuando se inicia o prueba un proceso.

CU iniciar proceso

Este caso de uso se puede llamar de dos formas diferentes. Una manera es haciendo click en el link que les llega a los usuarios responsables de las tareas indicando que pueden tomarla, como consecuencia de dicha acción pasa ha estado iniciada, solicitando autenticación en caso de que el usuario todavía no haya iniciado una sesión en el sistema. En la siguiente imagen se muestra un ejemplo del mail que se envía.

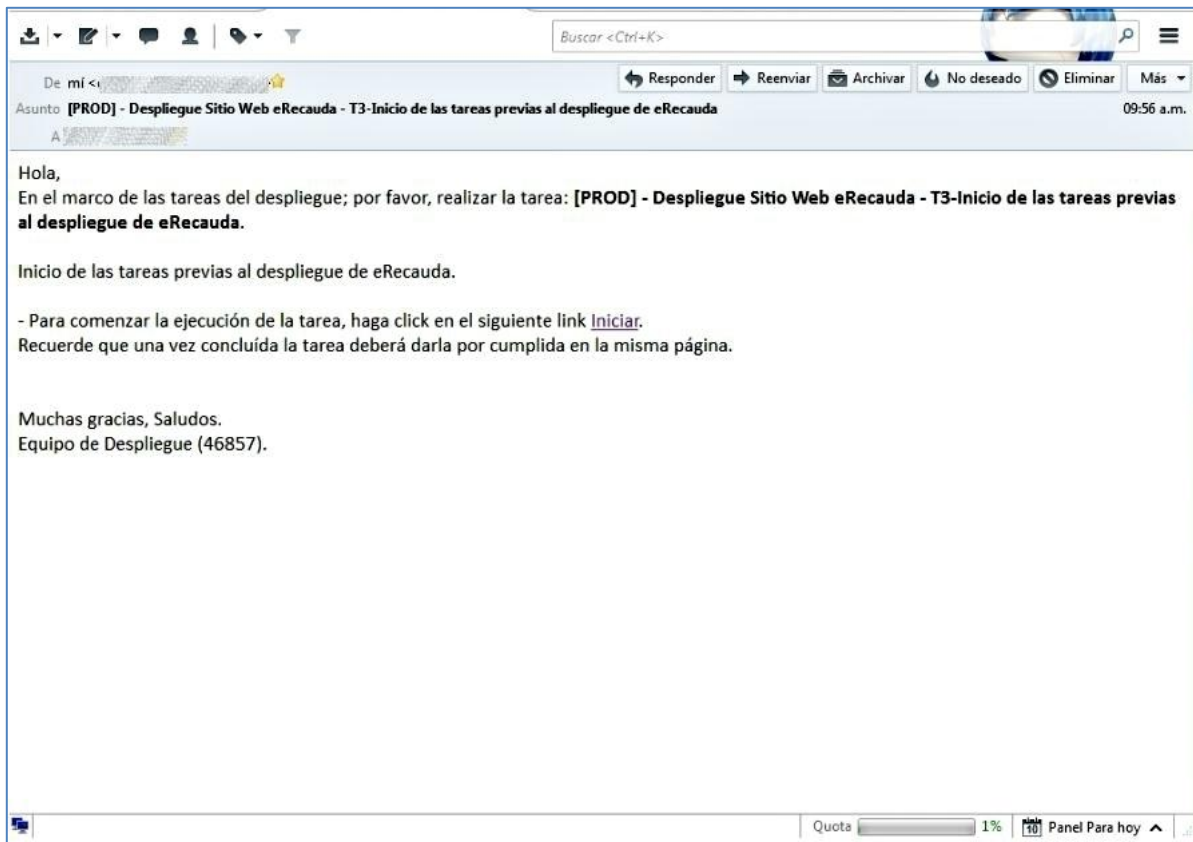


Figura 14 – Ejemplo de mail que se envía cuando una tarea se encuentra disponible para tomarla.

La segunda forma de llamar al CU es con una sesión iniciada en el P.E.P., el usuario consulta las tareas y puede tomar una, para eso se debe hacer click en la columna "**Nombre**" de la tarea. Como consecuencia de dicha acción el sistema cambiará la tarea al estado "iniciada". En la siguiente imagen se ve lo descrito previamente.

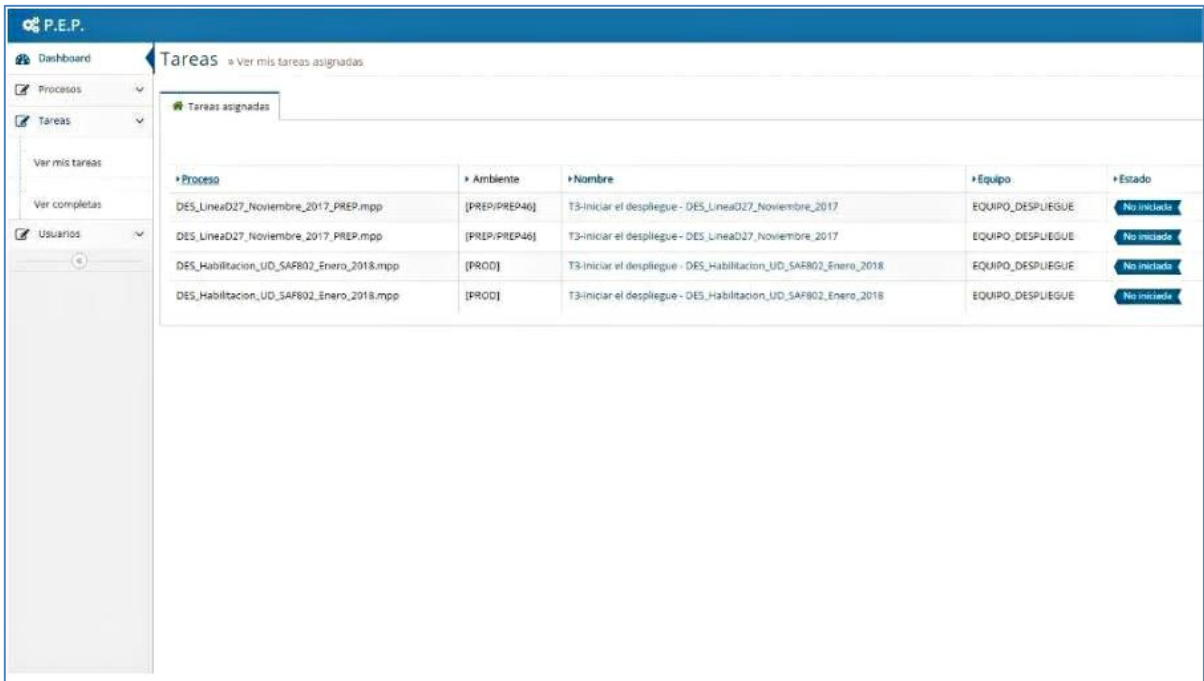


Figura 15 – Vista de tareas para poder iniciar una de ellas desde la aplicación.

Cuando finaliza el CU iniciar proceso, se puede ver como se informa que se ha iniciado de manera exitosa y también se muestra la información de la tarea junto con las funcionalidades que se puede efectuar sobre ella.



Figura 16 – Vista de gestión de tareas.

CU "Registrar error en la tarea", CU Finalizar tarea y CU "Forzar finalización de tarea"

En la vista anterior se puede ver cómo llamar a los casos de usos relacionados con las *tareas*, el CU *"Registrar error en la tarea"* se llama al hacer click en el botón "Error". Este es un estado temporal hasta que se resuelva el error. Al presionar dicho botón, se abrirá una ventana donde el usuario deberá ingresar un texto detallando el motivo del error. Este estado impide que se pueda continuar con las tareas posteriores que dependan de la tarea que originó el error.

Haciendo click en el botón "Hecho!" el usuario podrá ejecutar el CU "Finalizar tarea". Como consecuencia de esta acción, la ejecución del despliegue continua normalmente.

Y finalmente al hacer click en el botón "Forzar Hecho!", se llama al CU "Forzar finalización de tarea" de esta manera la tarea quedará cumplida, pero con alguna observación. Al presionar dicho botón, se abrirá una ventana donde el usuario podrá ingresar un texto detallando alguna eventualidad por el cual se decidió forzar hecho. Como consecuencia de esta acción, la ejecución del despliegue continua normalmente.

CU "Consultar dashboard"

El CU "Consultar dashboard" solo puede ser invocado desde el link del mail que se envía a todos los participantes del despliegue. De esta manera, los usuarios sin necesidad de haber iniciado sesión podrán visualizar la información relacionada con la ejecución de un despliegue. Es decir, el conjunto de las tareas que lo componen, como el estado en que se encuentran las tareas. En la siguiente imagen se puede ver como se visualiza el dashboard, se puede ver que las tareas están divididas en diferentes solapas según su estado actual y en la última solapa llamada "Grafo" se pueden ver la totalidad de las tareas del despliegue.

Dashboard
Despliegue ComprAR

Tareas del proceso [13]

Tareas en curso | **Tareas completas** | Tareas sin iniciar | Tareas listas para tomar | Grafo

★ Tareas completas [13]

Ambiente	Nombre	Equipo	Estado	Hora Inicio	Responsable
[PROD]	T4-Aviso de comienzo de tareas a grupo ComprAR	EQUIPO_DESPLIEGUE	Completa	2017-12-04 08:59:03.0	mmoure
[PROD]	T5-Aviso de deshabilitación del SLU a los usuarios del SAF	MESA de AYUDA	Completa	2017-12-04 09:15:04.0	mmoure
[PROD]	T6-Deshabilitación de SLU a los SAF que despliegan	WINDOWS	Completa	2017-12-04 09:16:17.0	mmoure
[PROD]	T8-Ejecutar Control 1 de la solapa "Controles COMPRAS" de la planilla de entregas	PROD	Completa	2017-12-04 09:21:16.0	mmoure
[PROD]	T9-Verificar Control 1 de la solapa "Controles COMPRAS" de la planilla de entregas	ANALISIS	Completa	2017-12-04 09:23:48.0	mmoure
[PROD]	T10-Ejecutar GRUPO 1 de la solapa "COMPRAS" de la planilla de entregas	PROD	Completa	2017-12-04 09:25:15.0	mmoure
[PROD]	T11-Ejecutar Control 2 de la solapa "Controles COMPRAS" de la planilla de entregas	PROD	Completa	2017-12-04 09:28:14.0	mmoure
[PROD]	T12-Verificar Control 2 de la solapa "Controles COMPRAS" de la planilla de entregas	ANALISIS	Completa	2017-12-04 09:30:05.0	mmoure
[PROD]	T14-Verificar en SLU desplegado no exista la opción "IR A INGRESO" en SCO y en parametro de contratación	MESA de AYUDA	Completa	2017-12-04 09:28:45.0	mmoure
[PROD]	T15-Habilitación de SLU a los SAF que despliegan	WINDOWS	Completa	2017-12-04 09:33:50.0	mmoure
[PROD]	T16-Aviso de la habilitación del SLU a los usuarios del SAF	MESA de AYUDA	Completa	2017-12-04 09:36:39.0	mmoure
[PROD]	T17-Informar a ComprAR finalización de tareas DGSIAF	EQUIPO_DESPLIEGUE	Completa	2017-12-04 09:38:02.0	mmoure
[PROD]	T18-Conformidad de ComprAR en relación a la finalización de tareas	EQUIPO_DESPLIEGUE	Completa	2017-12-04 12:39:28.0	mmoure

Figura 17 – Vista del dashboard.

CU consultar procesos

Este CU se llama desde el panel izquierdo, haciendo click en el menú "ver mis procesos". En la imagen siguiente se puede ver como muestra el resultado.

The screenshot shows the P.E.P. dashboard with a sidebar on the left and a main content area. The sidebar includes options like 'Dashboard', 'Procesos', 'Iniciar despliegue', 'Alta proceso de despliegue', 'Crear BPMN desde cronograma', 'Ver mis procesos', 'Tareas', and 'Usuarios'. The main content area is titled 'Procesos' and shows a list of processes under the heading 'Procesos hallados'. The list is presented as a table with the following data:

Nombre	Estado	Iniciador	Fecha Inicio
DES_Habilitacion_UD_SAF802_Enero_2018.mpp	En ejecución	jpetru	19/01/2018 9:46:49
SI_CRO_DES_ComprarAR_template_PEP.mpp	En ejecución	jpetru	24/10/2016 12:49:35
DES_UD-CB-REC-GS-CO-FR_en_ORGs_Enero_2017_PREP.mpp	En ejecución		22/12/2016 17:30:45
DES_UD_Cuota_Junio_2017_PREP	En ejecución		21/06/2017 16:09:23
DES_CB-REC-GS-CO-FR_en_ORGs_Nov_2016_PREP.mpp	En ejecución	jpetru	31/10/2016 15:24:13
SI_CRO_DES_ComprarAR_template_PEP.mpp	En ejecución	jpetru	24/10/2016 13:42:32
DES_UD-CB-REC-GS-CO-FR_en_ORGs_Enero_2017.mpp		jpetru	16/01/2017 9:21:43
DES_CB-REC-GS-CO-FR_en_ORGs_Nov_2016_PREP.mpp	En ejecución	jpetru	31/10/2016 15:59:08
DES_LineaD27_Noviembre_2017_PREP.mpp	En ejecución		30/10/2017 16:25:22
DES_UD-CB-REC-GS-CO-FR_en_ORGs_Enero_2017.mpp	En ejecución		03/01/2017 17:05:17

Figura 18 – Vista de procesos.

CU consultar tareas

Este CU se llama desde el panel izquierdo, haciendo click en el menú "ver mis tareas". En la imagen siguiente se visualiza tanto el panel lateral como el resultado de la consulta.

The screenshot shows the P.E.P. interface with the following data in the 'Tareas asignadas' table:

Proceso	Ambiente	Nombre	Equipo	Estado	Fecha Inicio
SI_CRO_DES_eRecauda_SitioWeb_template_PEP.mpp	[PROD]	T4-Envío de la Pauta de Capacidades a Seguridad	REPLICAS	Iniciada	2017-08-10 10:17:08.0
SI_CRO_DES_eRecauda_SitioWeb_template_PEP.mpp	[PROD]	T7-Carga de Concepto Recaudación Internet CRI por parte del Organismo	REPLICAS	Iniciada	2017-08-10 10:18:04.0

P.E.P. Plataforma de Ejecución de Procesos © 2016

Figura 19 – Vista de tareas.

4.4. Diagrama de clases y decisiones de diseño

A continuación, se expone el modelo del P.E.P. mediante el diagrama de clases, representando la estructura para luego profundizar en el comportamiento de los objetos principales del sistema con las decisiones y patrones de diseño [22] aplicados.

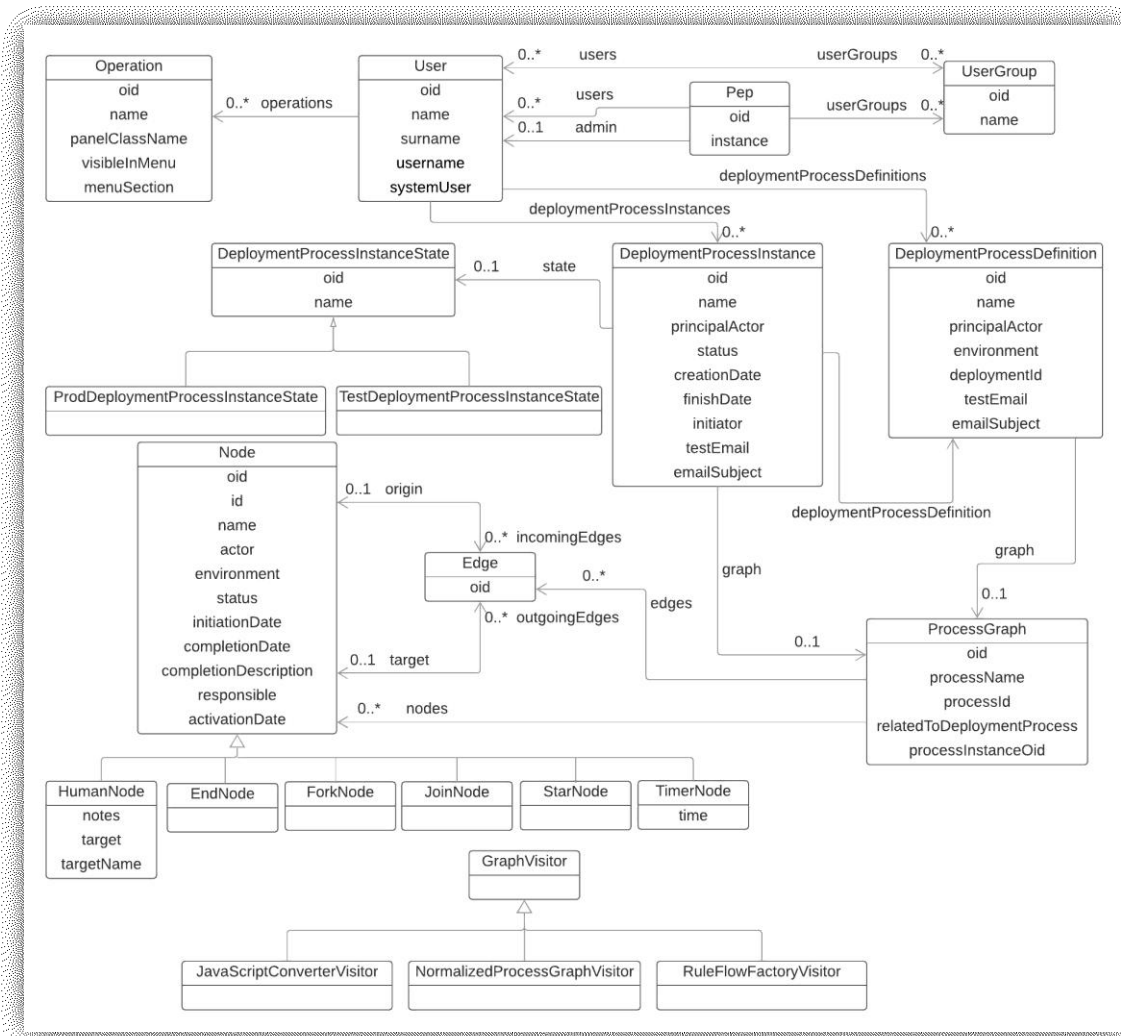


Figura 20 – Diagrama de clases.

- La clase **Pep** representa la aplicación y proporciona un punto de acceso global a ella. Expone una interfaz general, delegando las diversas peticiones de los clientes en los objetos del sistema. Esta clase implementa el patrón **facade** logrando reducir al mínimo la comunicación y las dependencias. Además, con el objetivo de garantizar una instancia única, se implementó el patrón de diseño **singleton**.

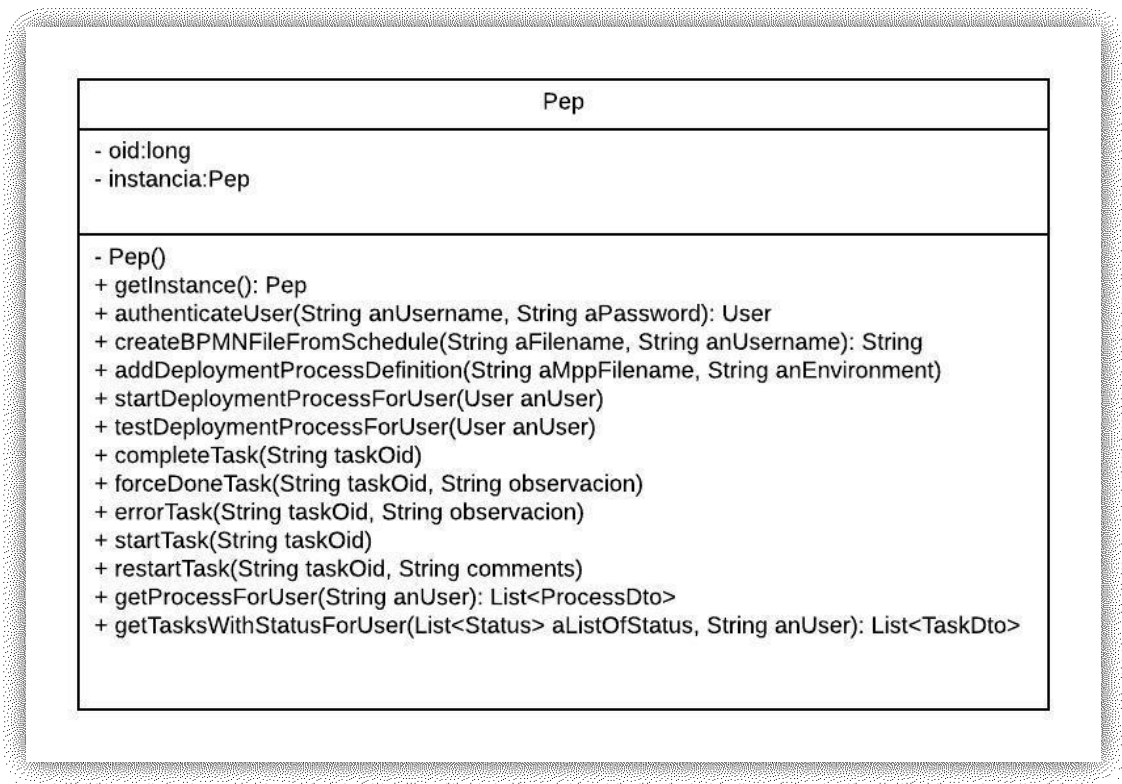


Figura 21 – Detalle de la clase Pep.

- Para la persistencia de los objetos, se decidió la utilización del concepto de **Persistencia por alcance** [20]. Este concepto establece que "todo objeto al cual se pueda navegar a partir de un objeto persistente, debe ser necesariamente persistente a su vez". Al utilizar dicho concepto, se ha mejorado la claridad del código por el hecho de no introducir en los objetos de dominio, código que está relacionado con la capa de persistencia, por ejemplo, SQL. También se logró mayor independencia con respecto al motor de base de datos, haciendo que sea transparente el cambio de motor si fuese necesario.
- Para la generación de OID de los objetos, se decidió utilizar una fórmula de **(UUID)"Identificador único universal"**, que es un número entero de 128 bits. Gracias a esta decisión, se lograron menos accesos a la base de datos al momento de la creación de los objetos. Al evitar el guardado y posterior consulta a la base con el único motivo de conocer el ID de dichos objetos.
- Para modelar los procesos de despliegue y las ejecuciones de los mismos, se aplicó el patrón de diseño **TypeObject** [21]. Con la aplicación del mismo, se logran crear "Clases" de manera dinámica, y está entre comillas porque exactamente son instancias de la clase **DeploymentProcessDefinition**. De esta manera las instancias de **DeploymentProcessDefinition** son los typeObjects. Y las instancias de la clase **DeploymentProcessInstance**, representan instancias de los

typeObject.

En el siguiente fragmento del diagrama de clases se puede ver el conocimiento del rol <<Class>>del patrón hacia su <<TypeClass>>.

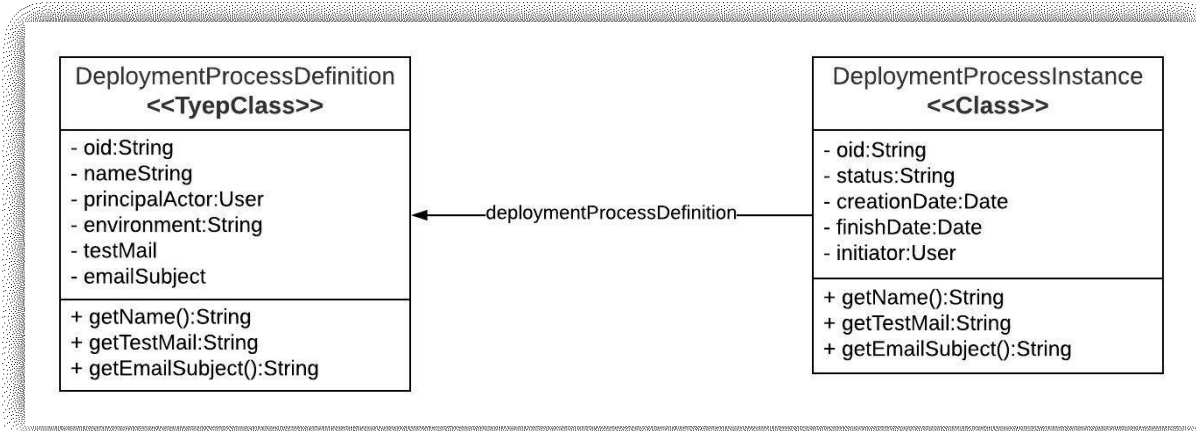


Figura 22 – Patrón TypeObject en las clases DeploymentProcessDefinition y DeploymentProcessInstance.

Y en el siguiente diagrama de instancias se puede apreciar el funcionamiento del patrón en el modelo. En dicho diagrama, vemos tres instancias de un proceso de despliegue <<Object>> con el conocimiento hacia su definición <<TypeObject>> correspondiente. Funcionalmente las 3 instancias representan ejecuciones del proceso de despliegue. De esta manera, las instancias de **DeploymentProcessInstance** delegan en su **DeploymentProcessDefinition**, todo el comportamiento que depende de dicha definición.

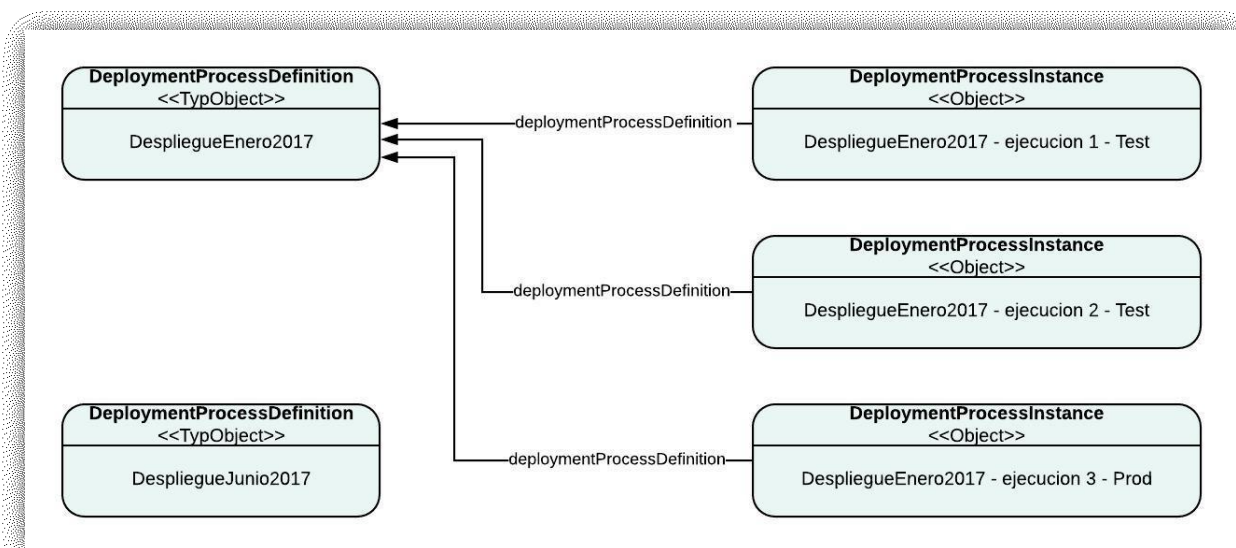


Figura 23 – Diagrama de instancias, aplicación del patrón TypeObject.

- La clase **Node**, representa las tareas de los procesos. Las subclases de **Node**, se hicieron para implementar el patrón **visitor** y poder aplicar la técnica de "**double dispatching**". Los objetivos de visitar dichos objetos eran dos, por un lado la generación de un archivo xml que representa el grafo de tareas, en notación BPMN [18] y por otra parte, para el armado de un archivo Javascript, necesario para la visualización del grafo de tareas en el cliente.

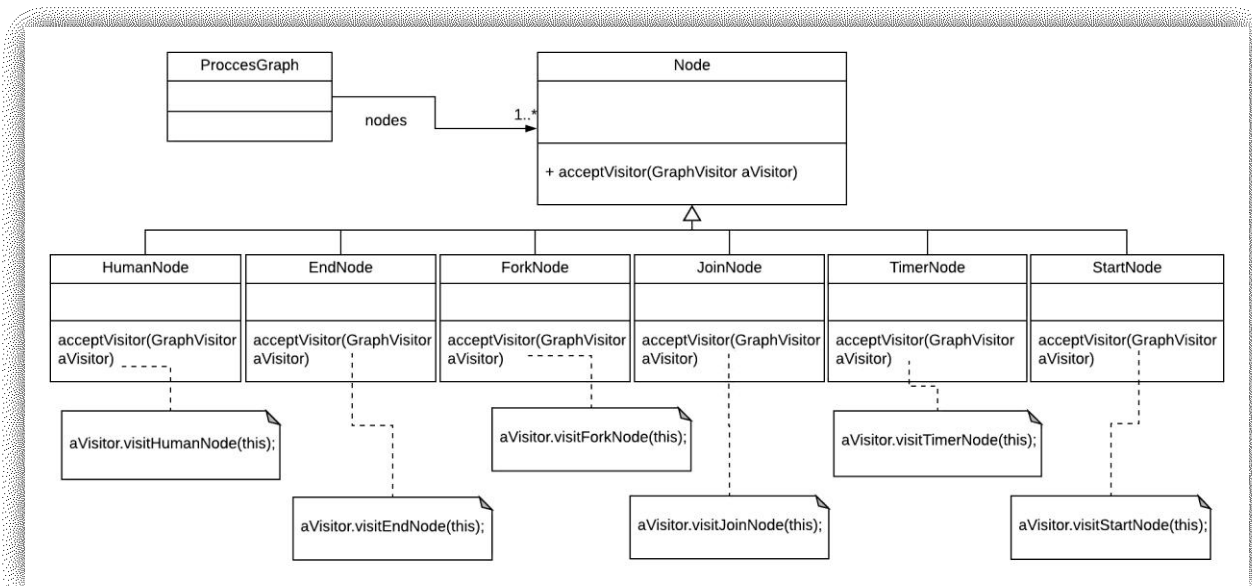


Figura 24 – Jerarquía de la clase Node, aplicación de patrón visitor.

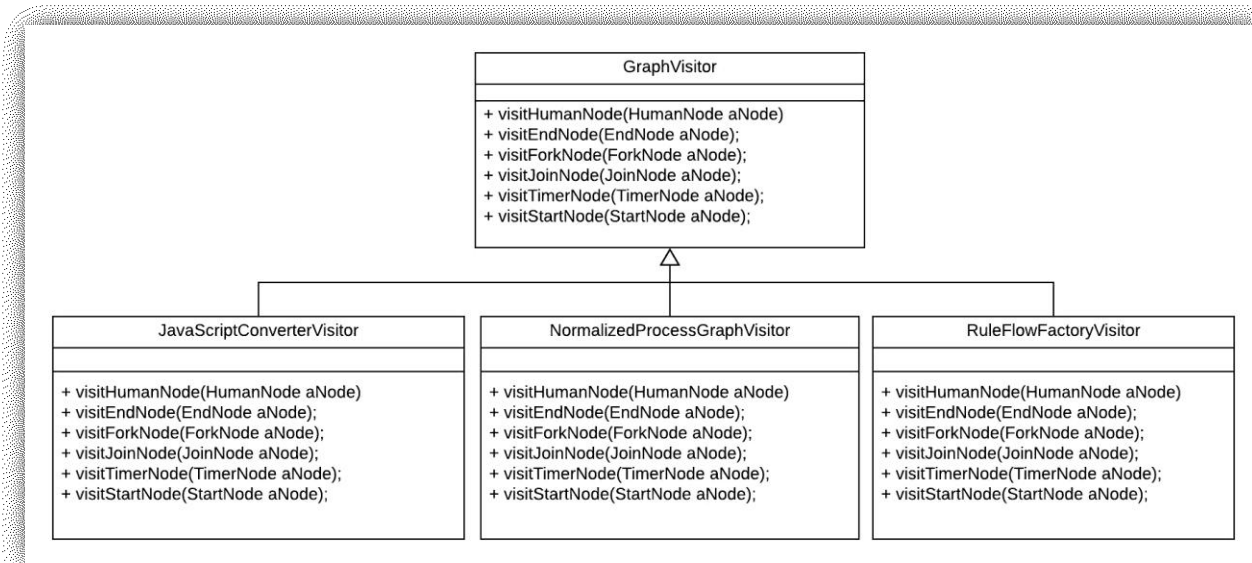


Figura 25 – Jerarquía de GraphVisitor, aplicación de patrón visitor.

- La jerarquía de **DeploymentProcessInstanceState**, representa la implementación del patrón **State** logrando de esta manera manejar la funcionalidad que depende del modo de ejecución de los procesos. Al aplicar este patrón se facilita la escalabilidad ante la necesidad de nuevos estados en el futuro.

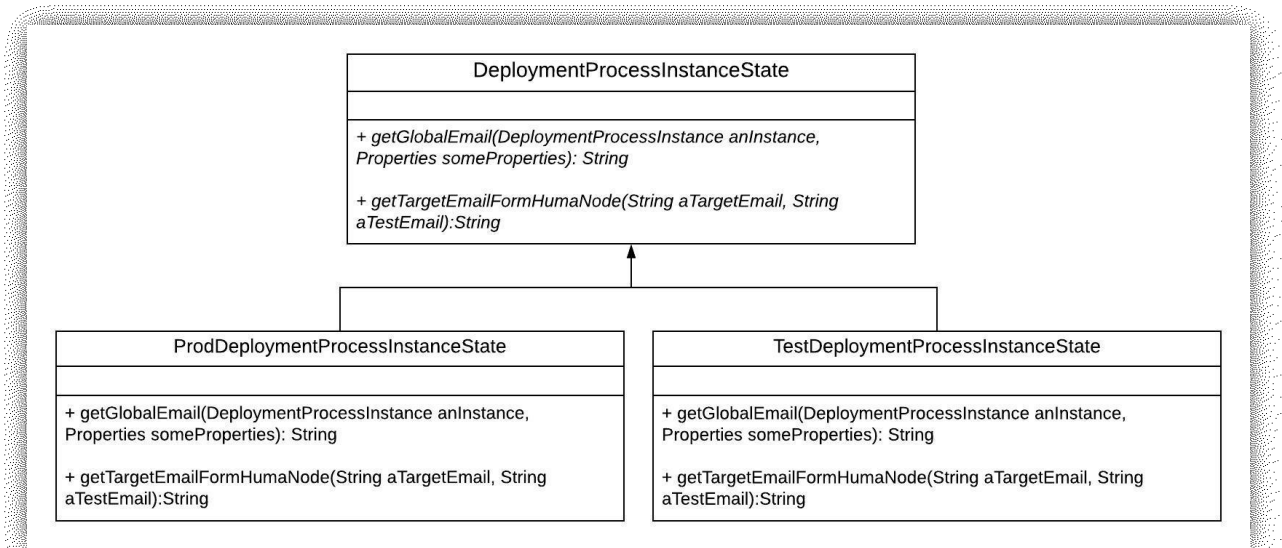


Figura 26 – Jerarquía de DeploymentProcessState, aplicación de patrón State.

- Las clases **ProcessGraph**, **Edges** y **Node** se utilizaron para representar el grafo de tareas.
- La clase **Operation**, se utilizó para determinar las operaciones que pueden efectuar los usuarios.
- La clase **User** representa a los usuarios del sistema.
- Y finalmente los **UserGroup** que como su nombre lo indica representa los diferentes grupos a los que puede pertenecer un usuario.

4.5. Arquitectura general y herramientas utilizadas

A partir de tareas de investigación de los productos disponibles en el mercado, se decidió encarar el desarrollo basando la arquitectura en una herramienta de código abierto denominada JBPM, sobre la cual se incorporaron desarrollos propios a través de la aplicación web llamada P.E.P. (plataforma de ejecución de proceso).

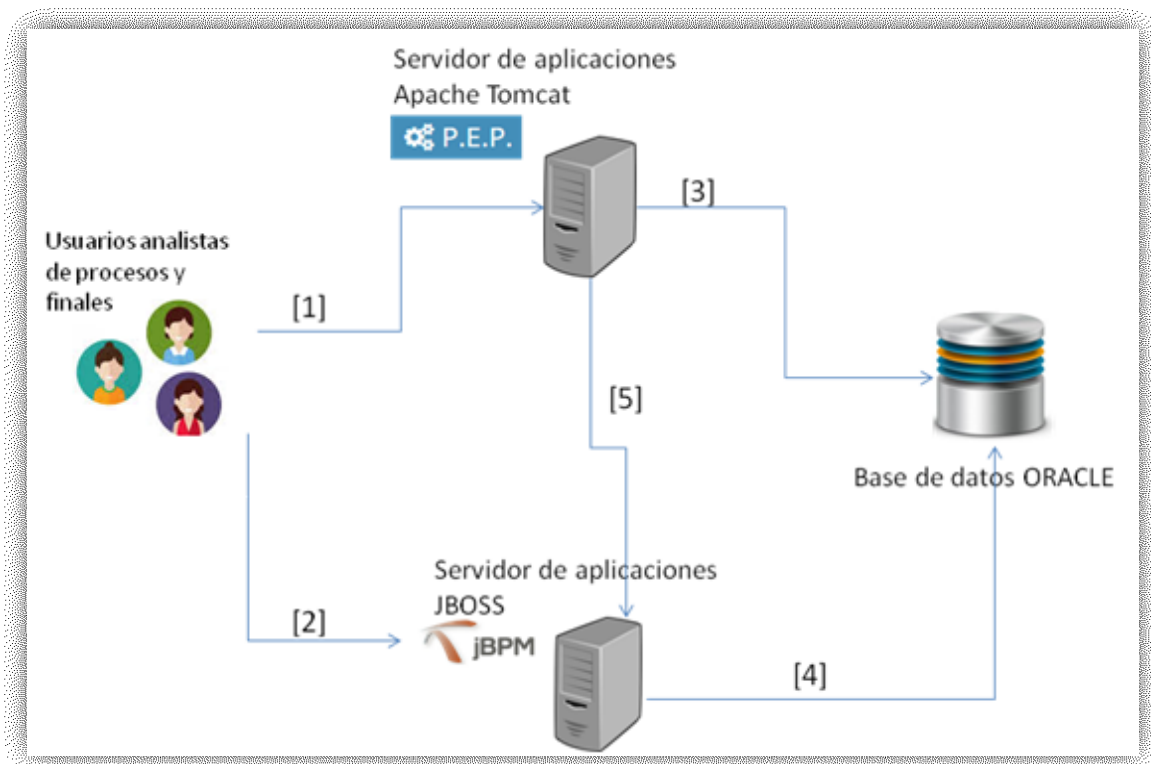


Figura 27 – Esquema de la arquitectura general.

- **P.E.P.:** herramienta web que funciona como interface a partir de la cual los usuarios pueden interactuar con las definiciones de los procesos y sus correspondientes instancias. Actúa como una bandeja de actividades en la cual van apareciendo las tareas asignadas a cada usuario para que éste pueda llevar adelante una determinada tarea y de esta manera continúe el proceso en cuestión.
- **JBPM:** esta herramienta es la responsable de ejecutar los procesos cuyas definiciones han sido desarrolladas por los usuarios analistas. Permite modelar los procesos de manera gráfica, incluso registrando los nuevos desarrollos para luego ser utilizados como parte de una paleta de opciones disponibles al momento de crear un nuevo proceso.
- **Base de datos ORACLE:** se utiliza para almacenar la información tanto de P.E.P. como también la necesaria para JBPM.

Interacciones

1. Los usuarios se conectan a P.E.P. a través de HTTPS, por lo que deben tener el certificado de seguridad apropiado instalado y configurado en su navegador web.
2. Los usuarios que se conecten a JBPM deben ser usuarios de P.E.P. y pertenecer al menos a un grupo de usuarios para poder ingresar en forma correcta a esta herramienta.
3. P.E.P. almacena en un esquema toda la información relacionada con los procesos que administra, así como también los usuarios y grupos.
4. JBPM utiliza otro esquema para almacenar la información de los procesos y sus instancias.
5. P.E.P. se comunica con JBPM a través de HTTP (vía API REST) para iniciar procesos y consultar el status de las tareas de cada uno de los usuarios. JBPM utiliza este mismo canal para ir notificando a P.E.P. sobre el avance de las tareas y ejecución de ciertas actividades como por ejemplo envío de mails.

4.5.1. Tecnologías utilizadas

Para el desarrollo de P.E.P. se utilizaron las siguientes tecnologías:

- El lenguaje seleccionado para desarrollar fue Java, siendo que es un lenguaje muy robusto, multiplataforma y además la organización cuenta con muchos recursos que poseen amplia experiencia en la utilización del mismo.
- El IDE de desarrollo elegido para el proyecto fue Eclipse.
- Para el desarrollo en el servidor se ha utilizado el framework Spring.
- En cuanto al mapeo objeto-relacional, el framework elegido fue Hibernate.
- El servidor de aplicaciones utilizado fue el Apache Tomcat 8.0.36.
- En cuanto al desarrollo del cliente, se decidió utilizar Wicket [19] dado que es un framework muy maduro, se desarrolla en java, tiene soporte para Ajax y además en la organización hay mucho conocimiento en la utilización del mismo.
- Como motor de BPM, se utilizó es JBPM 6.4, para lo cual se utilizó la herramienta Kie[17] tanto el cliente como el servidor.
- El servidor para que ejecute JBPM, es un JBoss dedicado.

- También se utilizó Apache Maven 3.3.9 dado que P.E.P. requiere Maven configurado para poder realizar los despliegues de los procesos al servidor JBPM.
- Como se mencionó anteriormente, como motor de base de datos se utilizó Oracle y el acceso a la base de datos se hizo a través de la herramienta Toad.
- El control de versiones se hizo con la herramienta CVS.

4.6. Resultados obtenidos

Con la primera versión de P.E.P. se lograron avances muy notorios. Como se explicó en el capítulo 3 (punto 3.4.1) la forma manual en que se hacían anteriormente los despliegues, incrementaba la posibilidad de cometer errores humanos. Además, los tiempos necesarios para analizar las tareas que se podían ejecutar según las dependencias de las mismas eran mayores.

Teniendo en cuenta los antecedentes, no solo se mejoró la ejecución del proceso de despliegue en general, sino que también, se facilitó tanto a los participantes como a cualquier otra persona interesada, la posibilidad de conocer la magnitud de cada despliegue, el seguimiento en línea de los mismos y la comunicación con el equipo de despliegue.

Más allá de los beneficios mencionados antes, surgieron limitaciones por el hecho de haber utilizado para la solución un workflow como JBPM, que sin duda es una herramienta muy potente, pero quizá demasiado compleja para este problema en particular. Los problemas detectados fueron los siguientes:

- Dificultad para la creación de una nueva versión de un determinado proceso cuando el despliegue se está ejecutando y se detecta un error. Hay que hacer una serie de pasos de los cuales los más significativos son: editar el cronograma project, importar dicho cronograma desde la herramienta P.E.P. para generar el archivo xml, crear en Kie [17] un nuevo proceso, importar el grafo (archivo xml del paso anterior) compilar y generar el JAR, luego con el cronograma y el JAR crear el proceso en la herramienta P.E.P. para luego poder iniciar el despliegue con las modificaciones. Claramente toda esta iteración de pasos para introducir cambios, insumía mucho tiempo que es muy valioso cuando se está desplegando nueva funcionalidad.
- Imposibilidad de modificar en línea un proceso que se está ejecutando, por ejemplo, al requerir anular una tarea y hacer que el despliegue continúe normalmente sin tenerla en cuenta.

- Otra limitación importante que trajo problemas fue la interfaz gráfica de JBPM, que hacía que a un proyecto de despliegue con más de 100 nodos comprometiera el normal desempeño de la interfaz gráfica, enlenteciéndola al extremo de hacer inviable su utilización debido a los tiempos de respuesta. Por ese motivo, para editar un cronograma ya cargado, había que repetir el procedimiento comentado en el primer punto.
- El equipo de testing no podía efectuar todos los casos de pruebas necesarios para garantizar la calidad del producto debido a que no contaba con el conocimiento de JBPM que le permitiera utilizar el cliente Kie [17].
- La infraestructura necesaria para la solución, fue otra limitación relevante, siendo que era necesario un servidor dedicado en cada uno de los ambientes (desarrollo, testing y producción) para que ejecute el motor de JBPM.

A raíz de los inconvenientes anteriormente mencionados, sumado al conjunto de nuevos requerimientos que fueron surgiendo, hicieron que se tomara la decisión de planificar la creación de una nueva versión de la herramienta (P.E.P. V2) que explicaremos en el siguiente capítulo.

5. Nueva versión de P.E.P

A partir de los resultados obtenidos con el uso del primer prototipo de la herramienta, y motivado por la permanente mejora continua se planificó una nueva versión.

5.1. Objetivos de la nueva versión

- Quitar el motor de JBPM de la solución. Reemplazando la funcionalidad que proveía JBPM por un desarrollo propio en la herramienta P.E.P.
- Mejorar los tiempos de pruebas y ajustes de los procesos de despliegue.
- Brindar una vista alternativa en el dashboard que permita una mejor visualización de la ejecución de los procesos de despliegue.
- Obtener más información de la ejecución de procesos, para brindar mejores estadísticas.
- Generalizar el P.E.P. para que pueda ser utilizado en cualquier proceso que requiera efectuar tareas y no únicamente en los procesos de despliegue.
- Agregar nuevos requerimientos como anular tarea, gestión de usuario y grupos.

5.2. Cambios funcionales

A continuación, se explicarán los cambios funcionales planificados para la nueva versión de P.E.P.

5.2.1. Casos de uso

Actores

En la nueva versión, se hizo una reestructuración de los roles y los actores del sistema. El rol de usuario registrado ya no es necesario, porque todo usuario registrado es un ejecutor de tareas, administra las tareas que es responsable, y puede consultar tareas y procesos a los que fue asignado. El rol iniciador de procesos de la primera versión, pasa a tener un rol más analítico del proceso, heredando toda la funcionalidad de un usuario

ejecutor de tareas más la capacidad de gestionar los procesos para los que es responsable como así también anular tareas del proceso, modificando su workflow. El administrador hereda toda la funcionalidad del ejecutor de tareas y del analista de procesos, más la nueva funcionalidad para gestionar usuarios y grupos.

En el siguiente diagrama, se muestra los actores con la funcionalidad representada en cada caso de uso.

Diagramas de caso de usos

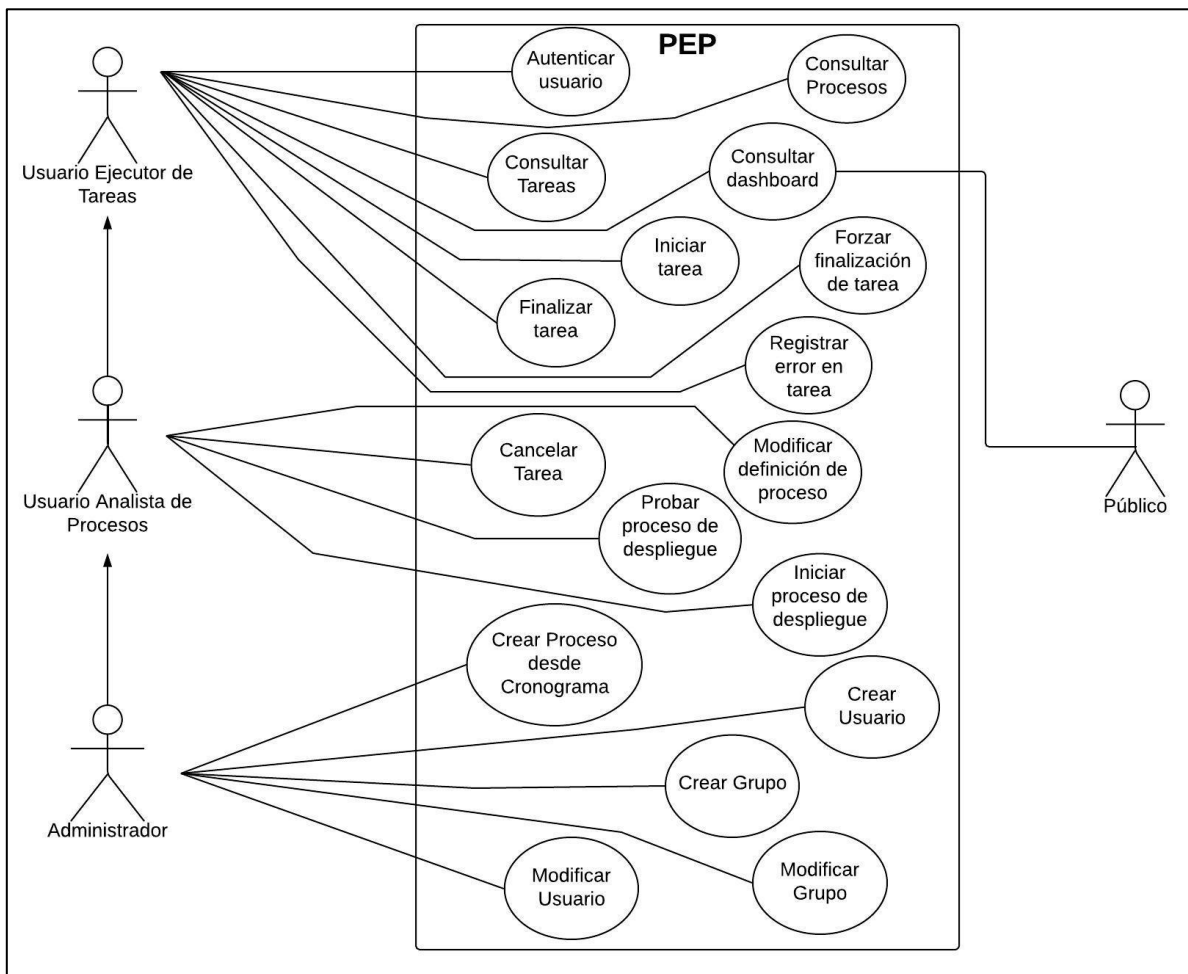


Figura 28 – Diagrama de casos de uso, nueva versión.

Especificación de los casos de uso

A. Casos de uso sin cambios

Los siguientes casos de uso, no han recibido modificaciones con respecto a la versión anterior:

- **CU - Autenticar Usuario**
- **CU - Consultar Tareas**
- **CU - Consultar Procesos**
- **CU - Consultar Dashboard**
- **CU - Iniciar Tarea**
- **CU - Forzar finalización de tarea**
- **CU - Finalizar tarea**
- **CU - Registrar error en tarea**
- **CU - Probar proceso de despliegue**
- **CU - Iniciar proceso de despliegue**

B. Caso de uso eliminado

CU - Crear BPMN desde cronograma

Este caso de uso, ha sido eliminado de esta versión. Su objetivo era transformar un proceso de despliegue en notación BPMN para luego con él, poder importar un proceso desde JBPM. Teniendo en cuenta que se ha decidido quitar JBPM de la solución, la funcionalidad ya no es necesaria.

C. Caso de uso modificado

CU - Crear proceso desde cronograma

Este caso de uso, además de cambiar su nombre (en la versión anterior se llamaba “Crear proceso de despliegue”) ha sido actualizado dado que ahora no se carga más el JAR que era generado desde el Kie, el único archivo que se ingresa es el cronograma proyect. Finalmente se agrega en este CU un chequeo de integridad del cronograma. En el siguiente cuadro, se visualiza el caso de uso con los cambios incorporados.

Datos Generales	
<i>Descripción Contextual</i>	Permite crear un proceso.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario Analista de procesos
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> • El usuario inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> • El sistema crea un nuevo proceso.
<i>Observaciones</i>	

Tabla 27 – CU Crear proceso desde cronograma, datos generales nueva versión.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> 1. Un usuario selecciona la opción de crear un nuevo proceso. 2. El sistema muestra la pantalla donde debe ingresar los datos necesarios. 3. El usuario ingresa los datos para poder crear el proceso: <ol style="list-style-type: none"> 3.1. Ingresa los siguientes datos: Nombre del proceso, email de prueba, asunto del mail y una descripción del proceso. 3.2. Selecciona un archivo MSP [16]. 4. El usuario ejecuta el chequeo de integridad. 5. El sistema hace el chequeo de integridad del archivo MSP que contiene el cronograma. 6. El sistema muestra el resultado del chequeo. 7. El usuario confirma la operación. 8. El sistema informa que se ha creado el proceso correspondiente con éxito.
Curso alternativo – No está cargado el archivo.
<ol style="list-style-type: none"> 3.1.1. El sistema verifica que no se han cargado el archivo del punto 3.2 3.1.2. El sistema informa que el archivo es obligatorio para crear el proceso.
Curso alternativo – Cancelar la creación del proceso.
<ol style="list-style-type: none"> 7.1. El usuario decide cancelar la operación. 7.2. El sistema vuelve al punto 3.

Tabla 28 – CU Crear proceso desde cronograma, desarrollo nueva versión.

D. Nuevos casos de uso

Los siguientes casos de uso, forman parte de los nuevos requerimientos que fueron surgiendo y se decidió incorporarlos en la nueva versión de la herramienta:

CU – Modificar definición de proceso

Datos Generales	
<i>Descripción Contextual</i>	Permite modificar la definición de un proceso ya creado.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario Analista de procesos
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> • El usuario inició una sesión. • El proceso existe en el sistema.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> • El sistema modifica un proceso.
<i>Observaciones</i>	Por el momento solo se puede actualizar los usuarios administradores

Tabla 29 – CU Modificar definición de proceso, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
<ol style="list-style-type: none"> 1. Un usuario selecciona la definición de un proceso. 2. El sistema muestra la pantalla los datos del proceso. 3. El usuario modifica los datos del proceso que pueden ser: nombre, descripción, mail de pruebas, asunto del mail y los usuarios administradores. 4. El usuario confirma la operación 5. El sistema actualiza los datos del proceso. 	

Tabla 30 – CU Modificar definición de proceso, desarrollo.

CU - Cancelar Tarea

Datos Generales	
<i>Descripción Contextual</i>	Permite cancelar una tarea para que pueda ejecutarse el proceso sin tenerla en cuenta.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario analista de procesos.
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> • La Tarea debe existir en el sistema y encontrarse en estado "pendiente" o "disponible". • El usuario inició una sesión. • El usuario gestiona el proceso que contiene la tarea a cancelar.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> • El usuario cancela la tarea.

<i>Observaciones</i>	<ul style="list-style-type: none"> Ninguna
----------------------	---

Tabla 31 – CU Cancelar tarea, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> El usuario indica que la tarea ha se ha cancelado. El sistema actualiza el estado de la tarea a "cancelada". El sistema verifica si corresponde cambiar el estado de las tareas sucesoras. <ol style="list-style-type: none"> El sistema cambia el estado de las tareas sucesoras a "disponible". El sistema notifica a los usuarios que pueden tomar dicha tarea.

Tabla 32 – CU Cancelar tarea, desarrollo.

CU - Crear usuario

En la versión anterior, se hizo una carga inicial por script. Y cada usuario nuevo debía cargarse también por script.

Datos Generales	
<i>Descripción Contextual</i>	Permite crear un usuario en el sistema.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> Usuario administrador.
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> El usuario que se va a crear, no existe en el sistema. El usuario administrador inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> El usuario administrador crea un nuevo usuario en el sistema.
<i>Observaciones</i>	<ul style="list-style-type: none"> Ninguna

Tabla 33 – CU Crear usuario, datos generales.

Desarrollo del Caso de Uso
Curso Normal
<ol style="list-style-type: none"> El usuario administrador indica que se va a crear un nuevo usuario. El usuario administrador ingresa todos los datos obligatorios para dar de alta el usuario. El Sistema agrega el nuevo usuario al sistema.

Tabla 34 – CU Crear usuario, desarrollo.

CU - Modificar usuario

En la versión anterior, las modificaciones debían realizarse mediante script SQL.

Datos Generales	
<i>Descripción Contextual</i>	Este caso de uso permite modificar un usuario.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario administrador
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> • El usuario debe existir en el sistema. • El usuario administrador inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> • El usuario administrador modifica un usuario.
<i>Observaciones</i>	<ul style="list-style-type: none"> • Ninguna

Tabla 35 – CU Modificar usuario, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
<ol style="list-style-type: none"> 1. El usuario administrador selecciona un usuario. 2. El sistema habilita los datos editables. 3. El usuario administrador modifica el/los datos del usuario. 4. El usuario administrador confirma la operación. 5. El sistema actualiza los datos del usuario. 	

Tabla 36 – CU Modificar usuario, desarrollo.

CU - Crear grupo

En la versión anterior, se hizo una carga inicial por script. Y cada grupo nuevo debía cargarse también por script.

Datos Generales	
<i>Descripción Contextual</i>	Permite crear un nuevo grupo en el sistema.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> • Usuario administrador.
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	

<i>Pre-condiciones</i>	<ul style="list-style-type: none"> El grupo que se va a crear, no existe en el sistema. El usuario administrador inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> El usuario administrador crea un nuevo grupo en el sistema.
<i>Observaciones</i>	<ul style="list-style-type: none"> Ninguna

Tabla 37 – CU Crear grupo, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
1.	El usuario administrador indica que se va a crear un nuevo grupo.
2.	El usuario administrador ingresa todos los datos obligatorios para dar de alta el grupo.
3.	El sistema agrega el nuevo grupo al sistema.

Tabla 38 – CU Crear grupo, desarrollo.

CU - Modificar grupo

En la versión anterior, las modificaciones debían realizarse mediante script SQL.

Datos Generales	
<i>Descripción Contextual</i>	Permite modificar un grupo.
<i>Actores Involucrados</i>	<ul style="list-style-type: none"> Usuario administrador
Relaciones	
<i>Extiende</i>	
<i>Usa</i>	
Proceso	
<i>Pre-condiciones</i>	<ul style="list-style-type: none"> El usuario y el grupo deben existir en el sistema. El usuario inició una sesión.
<i>Post-condiciones</i>	<ul style="list-style-type: none"> El usuario modifica un grupo.
<i>Observaciones</i>	<ul style="list-style-type: none"> Ninguna

Tabla 39 – CU Modificar grupo, datos generales.

Desarrollo del Caso de Uso	
Curso Normal	
1.	El usuario administrador selecciona un grupo.
2.	El sistema habilita los datos editables
3.	El usuario administrador modifica datos del grupo.
4.	El usuario administrador confirma la operación
5.	El sistema actualiza el grupo.

Tabla 40 – CU Modificar grupo, desarrollo.

5.2.2. Vistas

En las siguientes imágenes, haremos un recorrido por las diferentes vistas de la nueva versión de la aplicación P.E.P. Además, se indicarán los puntos de entrada a los casos de uso que han sido mencionados previamente.

CU autenticar usuario

Con la misma funcionalidad de la primera versión del P.E.P, al hacer click en el botón enviar se efectúa la autenticación. La diferencia más grande es en el cambio en el diseño y estilos de la aplicación.

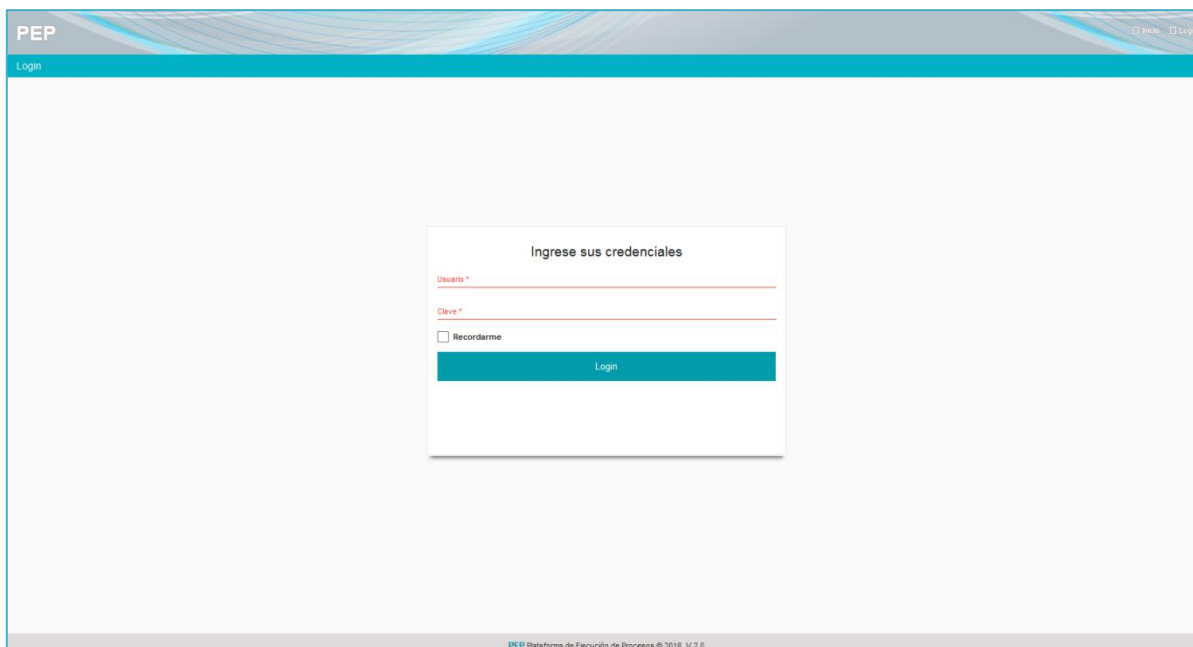


Figura 29 – Vista de login, nueva versión.

CU Crear proceso desde cronograma

Este caso de uso a diferencia de la primera versión, solo se requiere un archivo para crear un proceso, que es el cronograma proyect. Como se explicó anteriormente esto simplifica la creación de un proceso. Al hacer click en el botón validar se realiza un chequeo de integridad del cronograma, para luego importar y crear el proceso. El botón limpiar, como su nombre lo indica sirve para limpiar la información cargada en el formulario.

Nuevo Proceso

Nombre de proceso:

Email de prueba:

Asunto de email:

Descripción:

Seleccionar archivo Ningún archivo seleccionado

Limpiar Validar

Figura 30 – Vista de creación de proceso, nueva versión.

En la siguiente pantalla, se muestra el resultado de la validación, los resultados posibles pueden ser tanto "error" como "warning" tal como muestra la imagen. Luego haciendo click en el botón importar, se confirma la operación creando el proceso.

Resultado de la validacion

Filter _____

Nombre	Tipo	Descripcion
Iniciar el despliegue - DES_Linead27_Noviembre_2017	WARNING	La tarea de la fila [3] es un nodo inicial

Cancelar Importar

Figura 31 – Diálogo de resultado de la validación del cronograma.

Luego de haber importado el cronograma, se muestra la definición del proceso que se ha creado.

En dicha definición se puede ver el listado de las tareas. Además, en esta pantalla se puede observar dos botones en el margen superior derecho. El botón Modo Test y Modo Producción similar al comportamiento de los CU iniciar proceso de despliegue y CU probar proceso de despliegue de la primera versión de PEP.

Entorno	Nombre	Tareas predecesoras	Responsable	Descripción
[PROD PROD46 PROD801]	T3 - Iniciar el despliegue - DES_LineasD27_Noviembre_2017		EQUIPO_DESPLIEGUE	
[PROD PROD46 PROD801]	T4 - Envío de mail interno informando el inicio del despliegue	3	EQUIPO_DESPLIEGUE	
[PROD PROD46 PROD801]	T5 - Envío de mail a las autoridades informando el inicio del despliegue	3	EQUIPO_DESPLIEGUE	

Figura 32 – Vista de definición de proceso.

CU Modificar definición de proceso

En la imagen anterior (figura 32), vemos en la parte superior, un botón "guardar" el mismo sirve para confirmar los datos que se han modificado en una definición de proceso. Por el momento solo se pueden modificar los usuarios administradores del proceso (analistas de proceso) pero en el futuro inmediato también se podrán modificar los siguientes campos: nombre, descripción, email de pruebas y título del mail.

CU Consultar Procesos

Este CU se llama desde el panel izquierdo, haciendo click en el menú "Mis procesos". En la imagen siguiente se puede ver como muestra el resultado. A diferencia con la primera versión, esta pantalla muestra dos botones, el primer botón es para ir a las instancias de un proceso y otro botón es para poder mostrar la definición del proceso.

The screenshot shows a web interface titled 'Mis Procesos'. On the left, there is a sidebar with navigation options: 'Procesos en Ejecucion', 'Administrar Procesos', 'Mis Procesos', 'Nuevo Proceso', 'Procesos Iniciados', 'Administrar Tareas', and 'Administrar Usuarios'. The main content area has a search bar labeled 'Filtrar' and a table with the following data:

Nombre del proceso	Nombre de archivo	Descripción	Usuario creador	Fecha de creación ↓		
prueba de cronograma	DES_Linead27_Noviembre_2017.mpp		Orias, Mauro (morias)	12-04-2018 10:38:09		
crear procesos	DES_Linead27_Noviembre_2017.mpp		Orias, Mauro (morias)	08-04-2018 20:43:42		
prueba de pep2	DES_Linead27_Noviembre_2017.mpp		Orias, Mauro (morias)	06-04-2018 10:26:57		

At the bottom right of the table, there is a pagination control: 'Items per page: 10' and '1 - 3 of 3' with navigation arrows.

Figura 33 – Vista de mis procesos.

CU Probar proceso y CU Iniciar proceso

Ambos casos de usos, se invocan desde los botones *Modo Test* y *Modo Producción*, que se muestran en la definición de proceso, explicados en el CU Crear proceso desde cronograma.

Ambos casos de uso, crean una instancia del proceso. La diferencia son las direcciones de email que se utilizan para las notificaciones. Para acceder al listado de las instancias que han sido creadas, hay que hacer click en el menú "Mis instancias de procesos". En la siguiente imagen se puede ver la vista con el listado de las instancias.

The screenshot shows a web interface titled 'Mis Instancias de Procesos'. It features a search bar labeled 'Filtrar' and a table with the following data:

Nombre del proceso	Estado	Fecha de creación ↓	Usuario iniciador			
prueba de pep2	Iniciado	08-04-2018 18:47:38	Orias, Mauro (morias)			

At the bottom right of the table, there is a pagination control: 'Items per page: 10' and '1 - 1 of 1' with navigation arrows.

Figura 34 – Vista de mis instancias de procesos.

A diferencia de la primera versión del PEP, por cada instancia de procesos, se dibujan 3 botones. El primer botón, que sirve para iniciar o pausar una ejecución, el segundo para detener una ejecución y el último para ver información detallada de una instancia, mostrando nombre, persona que la inició, el estado, la fecha de creación y las tareas a ejecutarse.

Se puede observar en la siguiente imagen como se muestra dicha información. Los botones de iniciar/pausar y parar una instancia, en esta versión está funcionalidad aún no está implementada.

The screenshot displays a web interface for managing process instances. At the top, it says 'Mis Instancias de Procesos / prueba de pep2'. Below this, there's a section titled 'Instancia de Proceso' with the following details:

- Nombre: prueba de pep2
- Iniciador: Onias, Mauro (morias)
- Estado: Iniciado
- Fecha de Creación: 08-04-2018 18:47:38

Below the instance details is a section titled 'Tareas de la Instancia'. It includes a 'Filtrar' input field and a table of tasks:

Entorno	Nombre	Tareas predecesoras	Responsable	Estado	
[PROD/PROD45/PROD8 01]	T7 - Inicar los controles previos	3	EQUIPO_DESPLIEGUE	Disponible	⊕
[PROD/PROD45/PROD8 01]	T6 - Reprogramar el backup diario del día 09 de Noviembre para que no se realice en eSidif - SC - DW	3	UNIX	Disponible	⊕

Figura 35 – Vista de una instancia de proceso.

CU Iniciar Tarea

Este caso de uso, no tiene diferencia respecto a la primera versión. Una tarea la podemos iniciar desde la aplicación, como se muestra en la siguiente imagen.

Tarea

Entorno:
[PROD/PROD46/PROD801]

Nombre de la tarea:
T3 - Iniciar el despliegue - DES_LineaD27_Noviembre_2017

Notas:
-

Estado:
Disponible

Equipo asignado:
EQUIPO_DESPLIEGUE

Empezar

Figura 36 – Vista de gestión de tarea, cuando la tarea está en estado disponible.

También se puede iniciar desde el mail que es enviado a todos los integrantes del grupo que tiene asignada la tarea.

De: despliegue@mecon.gov.ar

Asunto: [PROD/PROD46/PROD801] - prueba de pep2 - T3-Iniciar el despliegue - DES_LineaD27_Noviembre_2017

A: Mauro Orias

Por favor realizar la siguiente tarea:

Entorno: [PROD/PROD46/PROD801]
Tarea: T3-Iniciar el despliegue - DES_LineaD27_Noviembre_2017
Descripción: -
Responsable: EQUIPO_DESPLIEGUE

- Para comenzar la ejecución de la tarea, haga click en el siguiente link [Iniciar](#).
- Recuerde que una vez concluída la tarea deberá darla por cumplida en la misma página.

Muchas gracias,
Saludos.

Figura 37 – Ejemplo de mail enviado a los usuarios que tienen asignada una tarea para que la puedan iniciar.

CU Forzar finalización de tarea, CU Finalizar tarea y CU Registrar error en tarea

En la siguiente imagen, se puede ver la vista de una tarea. En la misma se muestran los detalles de una tarea y además cuando la tarea se encuentre en estado "Empezada", aparecerán los 3 botones desde los cuales se pueden invocar los tres casos de uso (Forzar finalización de tarea, Finalizar tarea y Registrar error en tarea), al igual que en la versión anterior. Un cambio importante en la vista de tarea para la nueva versión de la herramienta, fue la incorporación de la lista de "log de transiciones", donde se puede ver la historia de los estados por los que fue pasando la tarea.

Tarea

Entorno
[PROD]

Nombre de la tarea
T106 - Iniciar el despliegue - DES LineaD27 Noviembre 2017 - Deshabilitación e instalación

Notas
Verificar cumplimiento de las tareas previas.

Estado
Empezada

Equipo asignado
EQUIPO DESPLIEGUE

✔ Hecho
⚡ Forzar Hecho
⚠ Error

Log de transiciones

Filtrar

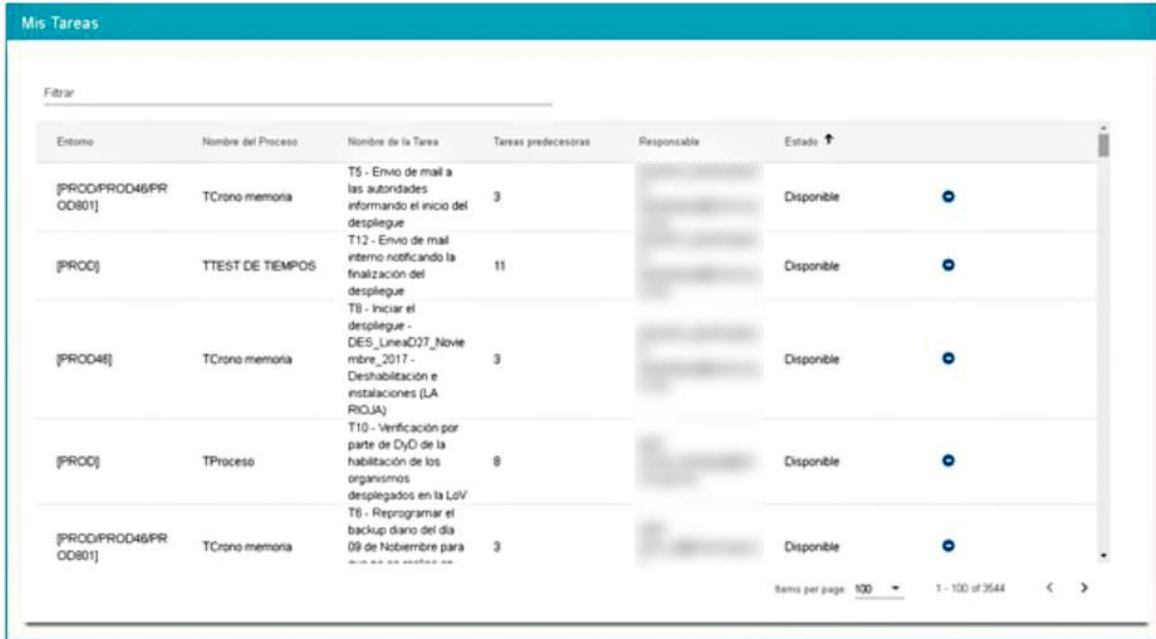
Estado inicial	Estado final	Usuario Ejecutor	Inicio ↓	Fin	Comentarios
Error	Empezada	Orias, Mauro (morias)	11-04-2018 23:29:01		
Empezada	Error	Orias, Mauro (morias)	11-04-2018 23:28:49	11-04-2018 23:29:01	Error en la tarea
Disponible	Empezada	Orias, Mauro (morias)	11-04-2018 23:28:19	11-04-2018 23:28:49	
Pendiente	Disponible	Orias, Mauro (morias)	11-04-2018 23:13:03	11-04-2018 23:28:19	
Inicial	Pendiente	Orias, Mauro (morias)	08-04-2018 18:47:41	11-04-2018 23:13:03	

Items per page: 10 1 - 5 of 5 < >

Figura 38 – Vista de gestión de tarea cuando se encuentra en estado empezada. Nueva versión.

CU Consultar Tareas

Este CU no tiene cambios con respecto a la primera versión, muestra de manera análogo la información de las tareas, incorporando un botón para poder navegar al detalle de la tarea y su historial de transiciones.



The screenshot shows a web interface titled 'Mis Tareas' with a search filter and a table of tasks. The table has the following columns: Entorno, Nombre del Proceso, Nombre de la Tarea, Tareas predecesoras, Responsable, and Estado. The tasks listed are:

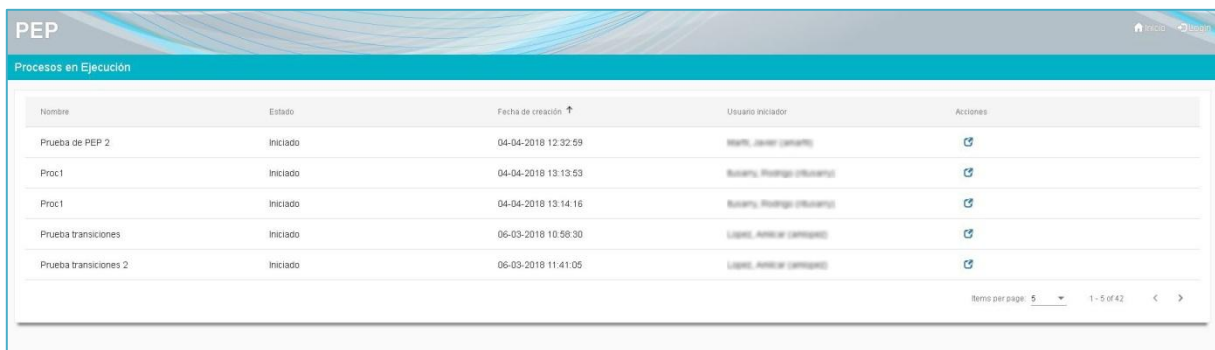
Entorno	Nombre del Proceso	Nombre de la Tarea	Tareas predecesoras	Responsable	Estado
[PROD PROD46PR00801]	TCrono memoria	T5 - Envío de mail a las autoridades informando el inicio del despliegue	3	[Redacted]	Disponible
[PROD]	TTEST DE TIEMPOS	T12 - Envío de mail interno notificando la finalización del despliegue	11	[Redacted]	Disponible
[PROD46]	TCrono memoria	T8 - Iniciar el despliegue - DES_Lines027_Noviembre_2017 - Deshabilitación e instalaciones (LA RIQUA)	3	[Redacted]	Disponible
[PROD]	TProceso	T10 - Verificación por parte de CyO de la habilitación de los organismos desplegados en la LoV	8	[Redacted]	Disponible
[PROD PROD46PR00801]	TCrono memoria	T6 - Reprogramar el backup diario del día 09 de Noviembre para [Redacted]	3	[Redacted]	Disponible

At the bottom right of the table, there is a pagination control showing 'Items per page: 100' and '1 - 100 of 3544'.

Figura 39 – Vista de de tareas, nueva versión.

CU Consultar Dashboard

Al igual que la versión previa, para invocar este caso de uso, no hace falta que un usuario se haya autenticado. En la versión anterior el caso de uso, solo se podía ejecutar a partir de un link que era enviado por mail a todos los interesados en el despliegue. Ahora en la nueva versión, además de continuar enviando dicho link por mail, también se puede acceder desde la pantalla inicial de la herramienta. Dado que puede haber muchos procesos ejecutándose de manera simultánea, la pantalla inicial muestra un listado de los mismos tal como se puede apreciar en la siguiente imagen.

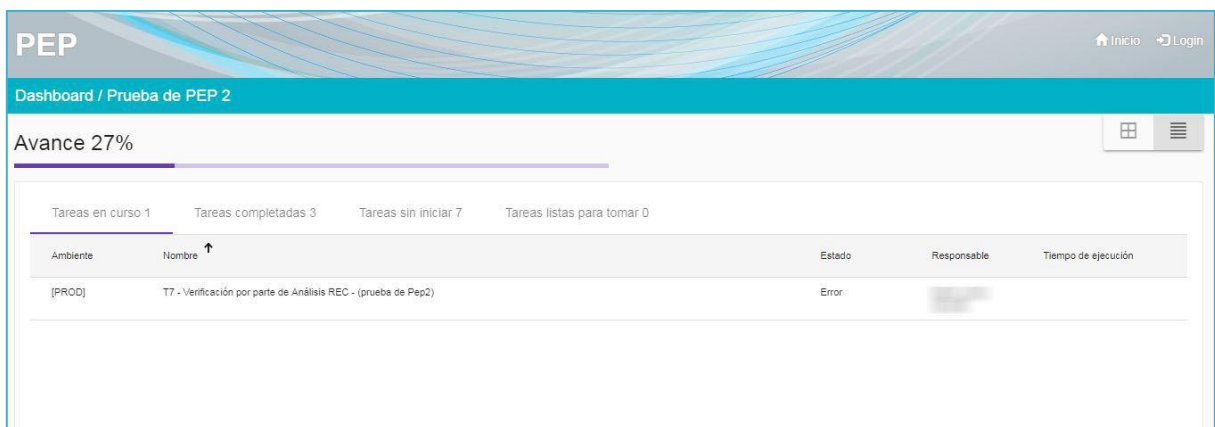


Nombre	Estado	Fecha de creación ↑	Usuario iniciador	Acciones
Prueba de PEP 2	Iniciado	04-04-2018 12:32:59	Marta, Javier (carrito)	🔗
Proci1	Iniciado	04-04-2018 13:13:53	Bosch, Rodrigo (bosch)	🔗
Proci1	Iniciado	04-04-2018 13:14:16	Bosch, Rodrigo (bosch)	🔗
Prueba transiciones	Iniciado	06-03-2018 10:58:30	Lopez, Amílcar (carrito)	🔗
Prueba transiciones 2	Iniciado	06-03-2018 11:41:05	Lopez, Amílcar (carrito)	🔗

Figura 40 – Vista de procesos en ejecución, nueva versión.

Para poder consultar un dashboard en particular, se debe hacer click en el botón de la columna "acciones" y se abrirá la vista del dashboard. En la nueva versión, la consulta del dashboard tiene dos formas de visualizarse, es decir por un lado se mantiene la vista por solapas igual a la versión previa, pero además se incorpora una segunda forma de visualizarse y es mediante cuatro cuadros repartidos en una misma pantalla. Ambas visualizaciones muestran en la parte superior izquierda una barra de progreso junto con el porcentaje de avance del despliegue. También se agregó información acerca de los tiempos de las tareas en las dos vistas.

En la siguiente imagen se puede ver la vista por solapas de la nueva versión.



Avance 27%	Tareas en curso 1	Tareas completadas 3	Tareas sin iniciar 7	Tareas listas para tomar 0
Ambiente	Nombre ↑	Estado	Responsable	Tiempo de ejecución
[PROD]	T7 - Verificación por parte de Análisis REC - (prueba de Pep2)	Error		

Figura 41 – Vista del dashboard por solapas, nueva versión.

Y en la siguiente imagen, se ve la nueva vista por cuadros repartidos en una misma pantalla. Esta vista mejoró mucho la experiencia del usuario al facilitar el seguimiento visual de las tareas del despliegue.

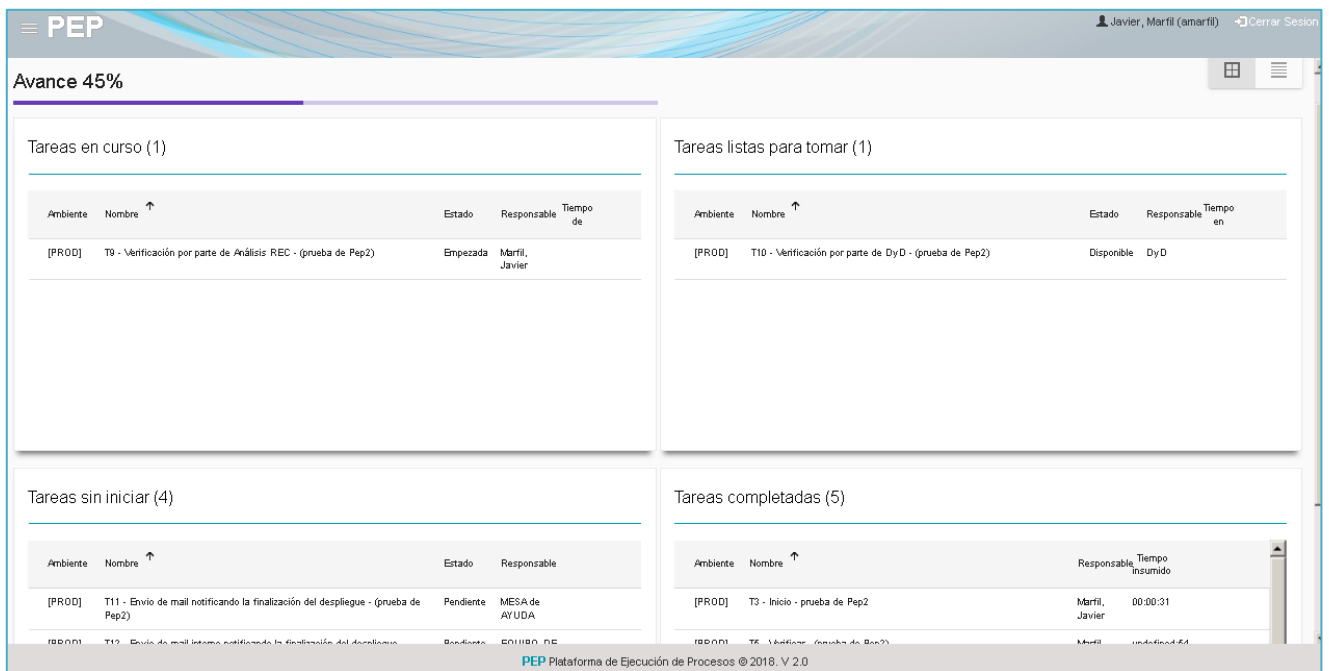


Figura 42 – Vista del dashboard por cuadros de estados, nueva versión.

CU Crear usuario y CU Modificar usuario

La creación y la modificación de usuario, son parte de los nuevos requerimientos que incorpora la nueva versión. Para acceder a la vista de usuarios, hay que hacer click en el menú "usuarios" del panel lateral izquierdo. En la vista de usuario, se pueden dar de alta nuevos usuarios. Para ello hay que hacer click en el botón "nuevo" y de esa manera a la derecha de la pantalla, se habilitan los campos para completar los datos. Finalmente haciendo click en el botón "guardar" se confirma la operación o bien con el botón "cancelar" se aborta la operación.

Para modificar los datos de un usuario, primero se debe seleccionar uno del listado y hacer click en el botón "editar". De esta forma parecerán a la derecha los campos editables para su modificación, una vez modificado los campos, con el botón guardar se confirma la operación o bien con el botón "cancelar" se aborta la operación. En la siguiente imagen, se puede ver la vista de usuarios.

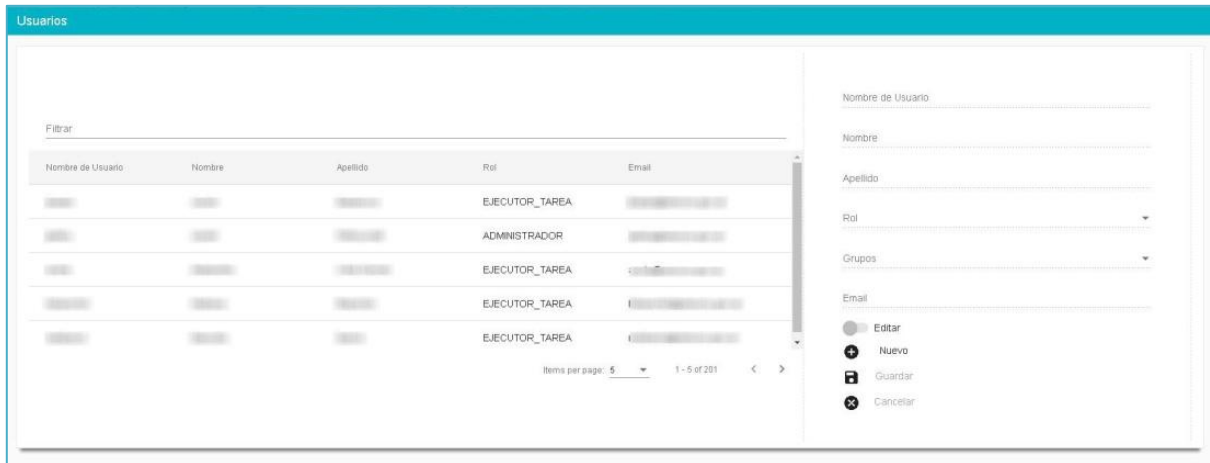


Figura 43 – Vista de usuarios.

CU Crear grupo y CU Modificar grupo

La creación y la modificación de grupos, también son parte de los nuevos requerimientos que incorpora la nueva versión. Para acceder a la vista de grupos, hay que hacer click en el menú "grupos" del panel lateral izquierdo. Esta vista tanto para la creación como para la modificación funciona igual que la vista de usuarios. En la siguiente imagen, se puede ver la vista de grupos.

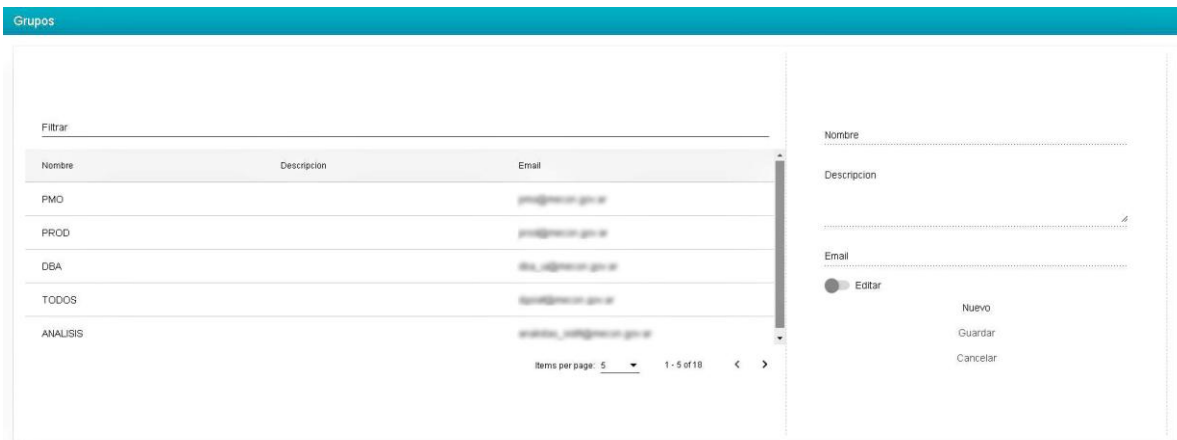


Figura 44 – Vista de grupos.

5.3. Cambios en la arquitectura general y herramientas

El principal cambio de la nueva versión de P.E.P. es el reemplazo de la funcionalidad que antes aportaba el motor de JBPM por nuevo desarrollo incorporado en la herramienta. Logrando de esta forma simplificar la infraestructura de la solución, al no necesitar más el servidor para JBPM.

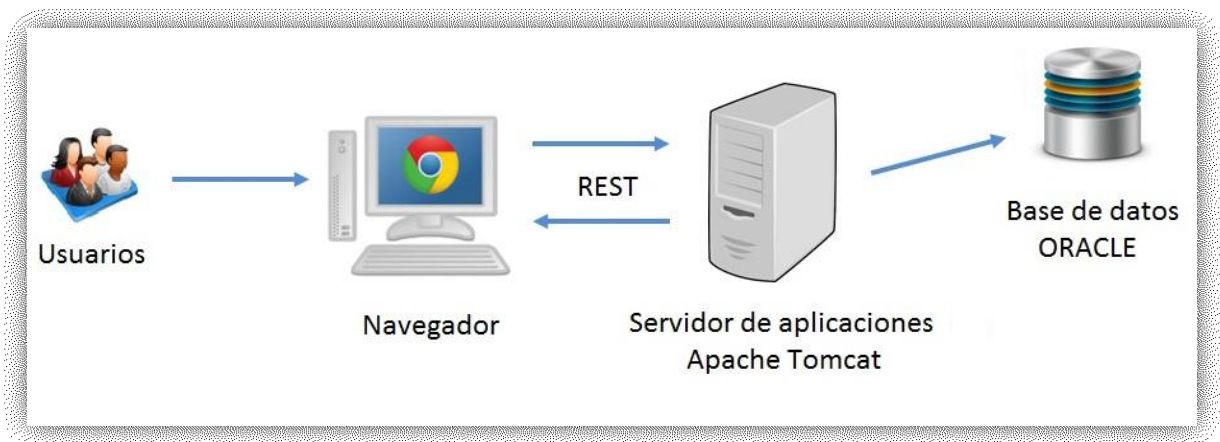


Figura 45 – Esquema de la arquitectura general de la nueva versión.

En el nuevo esquema, al igual que en la versión anterior, los usuarios se conectan a la aplicación P.E.P. a través de HTTPS por lo que deben tener el certificado de seguridad apropiado instalado y configurado en su navegador web.

A su vez el cliente P.E.P. se comunica con el servidor de aplicaciones a través de HTTPS (vía API REST) para invocar los servicios que sean necesarios de acuerdo a la funcionalidad utilizada. La información que viaja entre el cliente y el servidor, en ambos sentidos lo hace en archivos con formato JSON.

5.3.1. Cambios en las tecnologías utilizadas

Para el desarrollo de la nueva versión de P.E.P., también se han incorporado cambios en cuanto a las tecnologías y herramientas utilizadas:

- El lenguaje utilizado para el desarrollo continúa siendo Java, por los mismos motivos mencionados en el capítulo 4 (Punto 4.5.1).

- En esta versión se decidió utilizar dos IDEs de desarrollo; tanto el código del servidor como de la API REST, se continúa desarrollando con Eclipse. Y para desarrollar la capa del cliente, se decidió utilizar Visual Studio Code [25].
- En cuanto al desarrollo en el servidor se decidió cambiar al framework Springboot [33]. Dado que incorpora muchas novedades que facilitan el desarrollo. Tanto el manejo de las dependencias como el deploy de la aplicación son muy fáciles de realizar. Además, proporciona una serie de características no funcionales muy necesarias en los proyectos (por ejemplo, servidores embebidos, seguridad, indicadores, configuración externalizada).
- En cuanto al mapeo objeto-relacional, el framework continúa siendo Hibernate.
- El servidor de aplicaciones también es Apache Tomcat 8.0.36. al igual que la versión anterior.
- Para el desarrollo del cliente, se decidió en esta versión utilizar Angular2 [23]. Este framework, permite organizar el código del cliente aplicando el patrón MVC. Además, utiliza una comunicación bidireccional entre la vista y el modelo (Two-Way Data Binding) logrando ventajas en las actualizaciones de los clientes sin necesidad de recarga de páginas. Otro beneficio fue la utilización del módulo Material [24], dado que mejoró el desarrollo de los componentes gráficos. Se aprovechó además que en el proyecto muchos recursos comenzaron a capacitarse en dicha tecnología.
- Para el manejo de las dependencias para obtener mayor flexibilidad y velocidad, se optó por cambiar de Maven a Gradle [26].
- Como motor de base de datos se continúa utilizando Oracle y para consultar la base de datos la herramienta Toad.
- El control de versiones se hizo a través de la herramienta GIT [27].
- Finalmente, para facilitar el deploy de la aplicación en los diversos ambientes, se incorporó la herramienta Jenkins [28].

5.4. Cambios en el diseño

Nuevo diagrama de clases

En la siguiente imagen, se puede visualizar el diagrama de clases diseñado para la nueva versión de la herramienta. Luego se explicará los cambios más relevantes y los motivos.

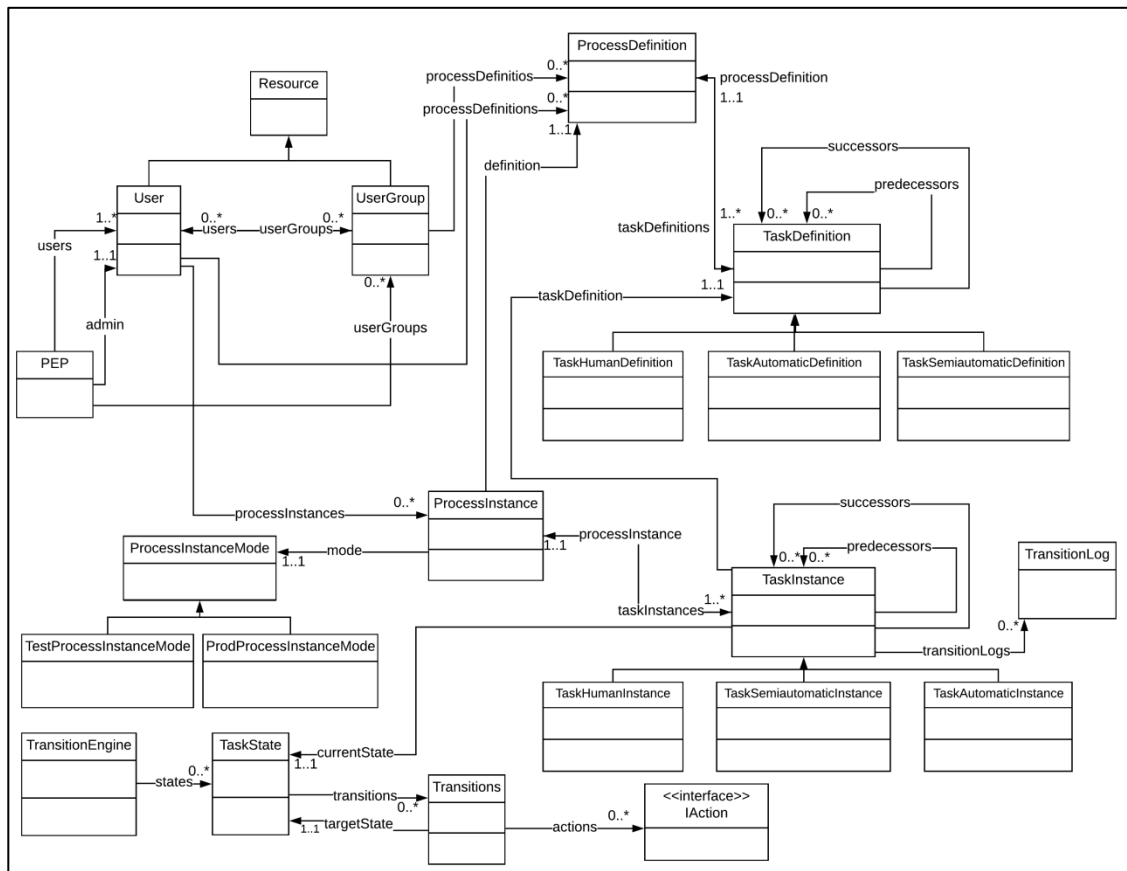


Figura 46 – Diagrama de clases de la nueva versión.

- La clase PEP, se mantuvo en el nuevo modelo con la misma idea que la versión anterior.
- Otra característica que se mantuvo fue la utilización del concepto de persistencia por alcance. Para mantener los beneficios mencionados para la versión anterior.
- Para la generación de OID de los objetos, se decidió continuar utilizando la misma fórmula de **(UUID)"Identificador único universal"**.
- Para modelar el modo de ejecución, se mantuvo la misma jerarquía que la versión anterior, únicamente se modificaron los nombres de las clases para que reflejen de forma más adecuada lo que representan.
- Para el nuevo modelo, se decidió representar el grafo de tareas, sin utilizar las clases que representaban al grafo en la versión anterior como (Edge, Node y ProcessGraph). Para la nueva versión, las definiciones de tareas tienen una lista de tareas iniciales. A su vez, las tareas tienen dos listas, una con las tareas sucesoras y otra con las tareas predecesoras. De esta manera el grafo está implícitamente representado por las dos clases, quedando de esta forma un modelo de menor complejidad y con mejor representación del dominio. Hay dos

tipos de grafos que se pueden conformar en la nueva versión. Por un lado, la definición de proceso (**ProcessDefinition**) con las definiciones de tareas (**TaskDefinition**) que lo componen. Y, por otro lado, los grafos que se corresponden con las ejecuciones de los procesos. Es decir, las instancias de procesos (**ProcessInstance**) con las instancias de tareas (**TaskInstance**) que lo conforman.

- Otra modificación importante en el modelo, fue la representación de las tareas, en la versión anterior los nodos (clase Node y su jerarquía) del grafo representaban también las tareas. Esto se prestaba a confusión siendo que no solo era el nodo como parte de la estructura grafo, sino también la entidad tarea del dominio.

En esta versión, se decidió la aplicación del patrón **typeObject**, ya no solo para modelar los procesos al igual que la versión anterior, sino también para el modelado de las tareas. Por tal motivo se incorporaron las clases **TaskDefinition** y **TaskInstance** con sus respectivas jerarquías.

Las jerarquías de tareas, tanto en las definiciones como en las instancias de las mismas, vemos tres subclases:

TaskHuman: Representa las tareas que son efectuadas completamente por las personas. Son las únicas implementadas en esta versión de la herramienta.

TaskAutomatic: En el futuro representarán las tareas que pueden ser ejecutadas íntegramente por el sistema sin intervención de las personas.

TaskSemiautomatic: En el futuro representarán un híbrido de los dos tipos de tareas mencionados previamente. Es decir, serán tareas que requerirán intervención de personas en alguna etapa y ejecución automática en otro tramo para poder completarse.

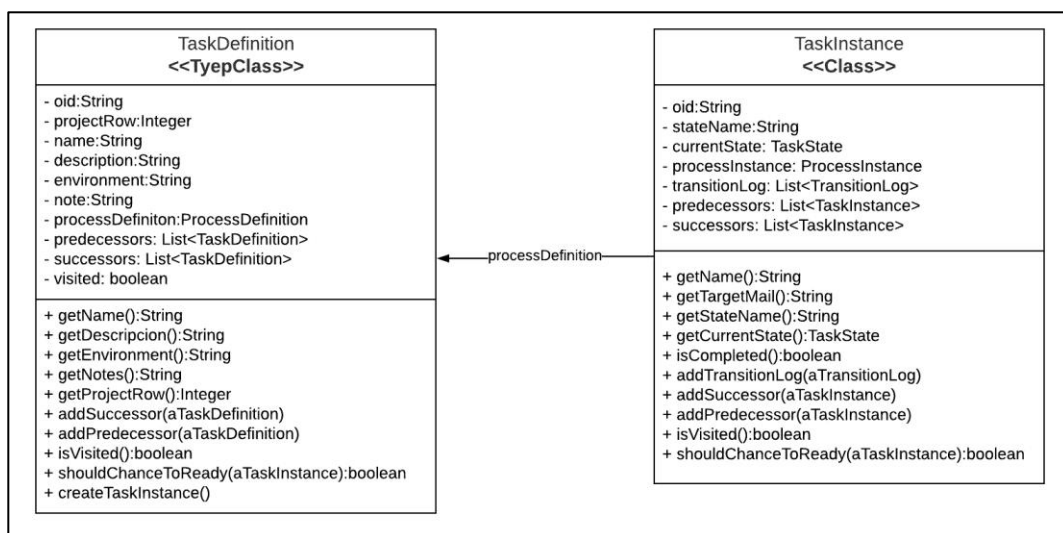


Figura 47 – Clases TaskDefinition y TaskInstance, aplicación de patrón TypeObject.

- En esta versión se decidió implementar una máquina de estados para las instancias de tareas (**TaskInstance**). De esta manera las tareas tienen asociado un **TransitionEngine**, clase que representa el DTE, en él se definen los estados por los que puede pasar una tarea como así también las diferentes transiciones que permiten pasar de un estado a otro. La clase **TransitionEngine** permite además configurar tanto de las acciones que se van a ejecutar en las transiciones como también su orden de prioridad para el momento de la ejecución. Las clases (**TransitionEngine**, **TaskInstance** y **Transitions**) no son persistentes, la información que se persiste, es el estado actual en la instancia de la tarea (**TaskInstance**) y también se persisten las transiciones que se ejecutan en una tarea mediante la clase **TransitionLog**.

La máquina de estados, permitió tener un control de los estados por los que pueden pasar las tareas, y configurar las acciones a ejecutar. Además, va a permitir en el futuro cuando se implementen las tareas automáticas y semiautomáticas, poder definir diferentes máquinas de estados si es que las necesidades que cada una de ellas lo requiere.

En la siguiente figura se muestran las clases involucradas con mayor detalle.

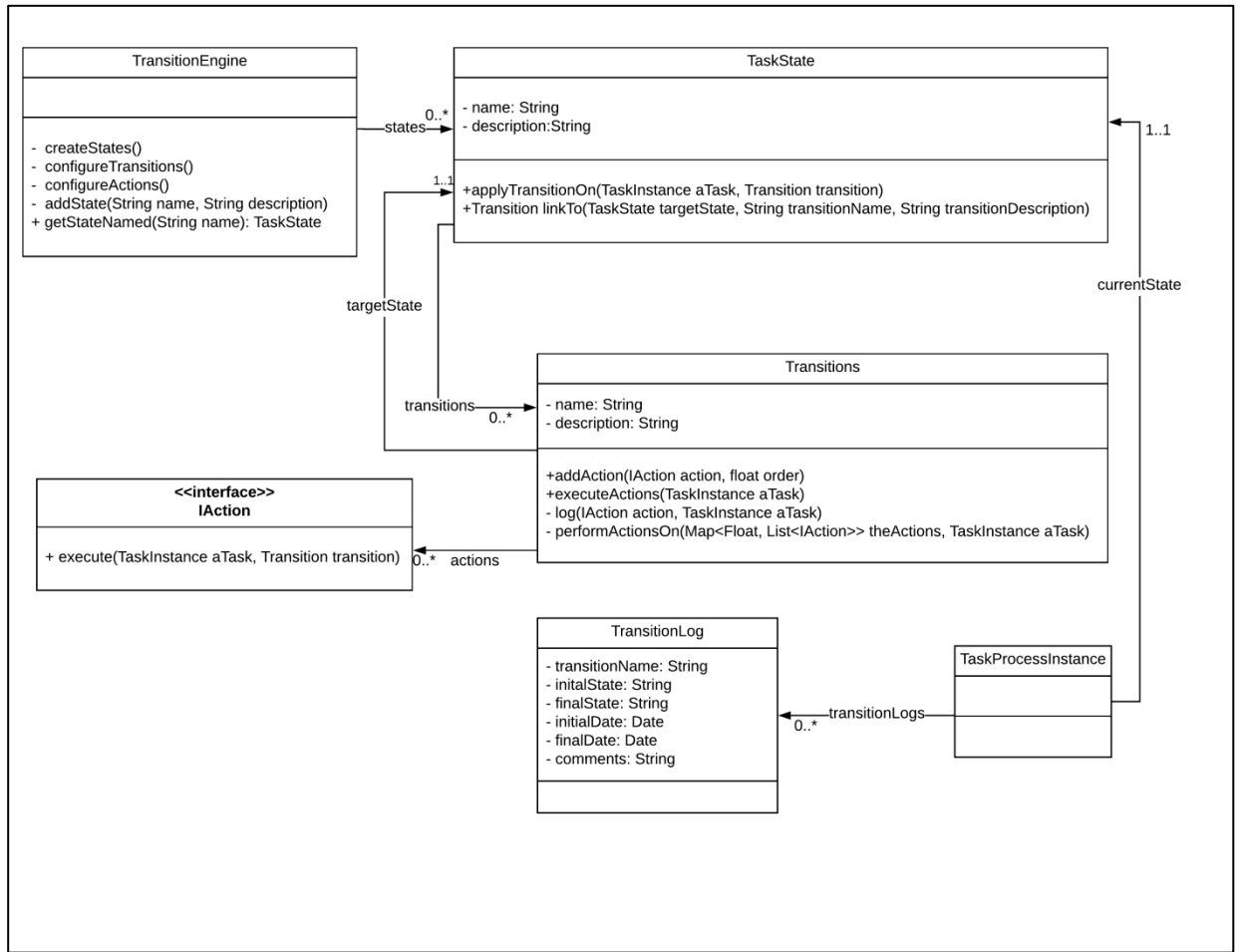


Figura 48 – Clases que conforman la máquina de estados.

En la siguiente imagen se puede ver el DTE (diagrama de transición de estados) de una instancia de tarea. En el mismo, se detallan los diferentes estados por los que puede pasar la tarea como así también las transiciones definidas para cambiar de un estado a otro.

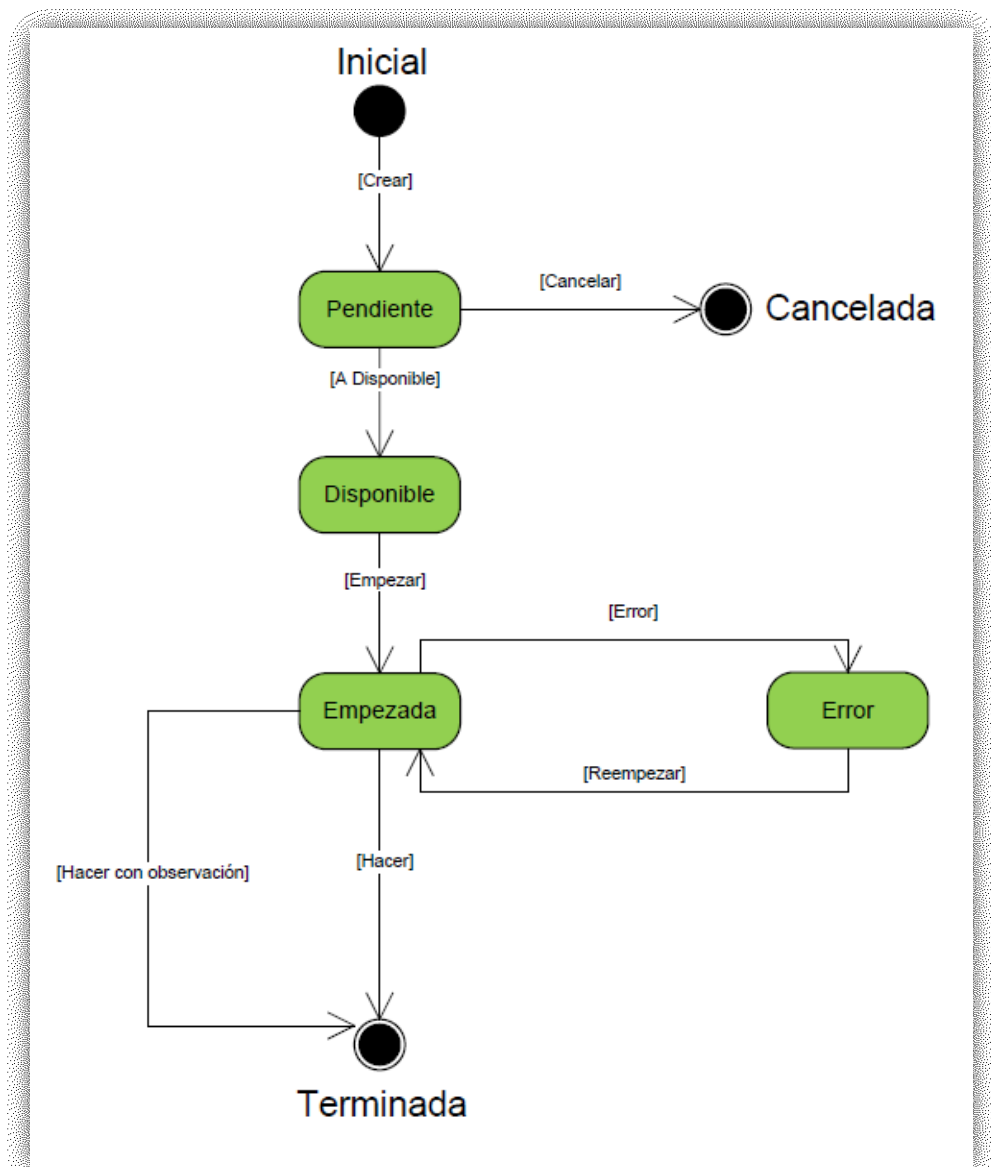


Figura 49 – Diagrama de transición de estados de una tarea.

Inicial: Representa una tarea que se está creando.

Pendiente: Representa que la tarea fue creada.

Disponible: Representa que la tarea está lista para ser tomada por un usuario.

Empezada: Indica que la tarea ha sido iniciada por un usuario.

Error: Indica que una tarea tuvo algún error.

Terminada: Representa que la tarea ha sido finalizada.

Cancelada: Indica que la tarea fue cancelada para el proceso que se está ejecutando.

- Si bien, tanto la clase **User** como **UserGroup** se mantuvieron en el nuevo modelo, se hizo una generalización de las mismas que ahora son hijas de la clase **Resource**. Dicho cambio fue motivado para poder asignar una tarea no solo a un grupo como en la versión anterior sino también a un usuario específico. De esta forma ahora las tareas tienen asignado un recurso (**Resource**).
- La clase Operation se quitó en el nuevo modelo, siendo que la seguridad ahora es manejada de acuerdo al role que posee el usuario. Para ello se definieron tres roles posibles (Ejecutor de tarea, Analista de proceso y Administrador).

5.5. Simulación de una ejecución de despliegue

Para mostrar el funcionamiento de la herramienta, se hizo un breve video donde se puede ver como se crea una definición de proceso a partir de un cronograma MS-Project. Luego desde la definición de proceso creada, se ejecuta en modo test el proceso para poder observar cómo se van gestionando las tareas y como va progresando el despliegue en la vista del dashboard.

5.6. Resultados obtenidos

Como vimos a lo largo de este capítulo, la segunda versión de la herramienta P.E.P. incorporó muchos cambios, funcionales, de diseño, en la arquitectura como así también en las tecnologías utilizadas. Como consecuencia de dichos cambios y a partir de la puesta en marcha de la nueva versión, se vislumbraron los siguientes resultados:

- La nueva versión de la herramienta P.E.P. logró fundamentalmente mantener la principal funcionalidad de la versión anterior, pero con una arquitectura mucho más simple en cuanto a infraestructura utilizada, siendo que ahora no se necesita el motor de JBPM para la resolución de dependencias de tareas, ejecución del proceso y envío de notificaciones.
- Lo mencionado en el primer punto, tuvo también consecuencias muy positivas para los usuarios de la herramienta, dado que simplificó las tareas que deben hacer los analistas de procesos, es decir pudieron ejecutar la misma funcionalidad,

pero en un tiempo menor y llevando a cabo una menor cantidad de pasos en comparación con la versión previa de P.E.P. Ahora el analista de proceso solo debe ajustar el cronograma e importarlo nuevamente para reacomodar un determinado proceso. Por consiguiente, mejoraron de manera notable los tiempos que insumían los ciclos de prueba y ajuste de los procesos.

- La nueva vista del dashboard también mejoró mucho la forma en que los usuarios pueden ver el avance de los procesos, siendo que la misma se actualiza en línea sin necesidad de hacer constantes recargas de la página. De esta manera todos los usuarios pueden visualizar rápidamente el progreso de las tareas, observando cómo las mismas van trasladándose de un cuadro a otro a medida que cambian de estado. También pueden observar información relacionada con los tiempos de las tareas y el porcentaje de avance de la ejecución de todo el proceso.
- La incorporación de la lista de transiciones de estados por los que pasó una determinada tarea en el modelo (**TransitionLog**), permitió el registro histórico tanto de los tiempos en que fue cambiando de estado una determinada tarea como así también, el registro de las causas de los errores que se fueron dando en dicha tarea.
- Tanto los cambios funcionales como en el modelo, permitieron generalizar lo suficiente para que P.E.P. en el futuro pueda ser utilizado también para la gestión de otros tipos de procesos que requieran la ejecución de tareas y no únicamente procesos de despliegue.

Los resultados obtenidos en esta versión son muy satisfactorios en cuanto a los objetivos planteados cuando se comenzó a gestarse el proyecto, siendo conscientes que hay muchos requerimientos pendientes de implementación que sin lugar dudas van a enriquecer aún más los beneficios de la herramienta P.E.P.

6. Trabajos relacionados

Por las características de la herramienta PEP, vemos que se relaciona con las siguientes aplicaciones:

JBPM

Es un conjunto de herramientas flexible para la gestión de procesos empresariales (BPM - Business Process Management). Es un software de código abierto, publicado bajo Apache Software License 2.0.

Establece el puente entre los analistas de negocios y los desarrolladores. Los motores BPM tradicionales tienen un enfoque que se limita únicamente a personas no técnicas. JBPM tiene un doble enfoque: ofrece funciones de administración de procesos de manera que es atractivo tanto para los usuarios del negocio como para los desarrolladores.

Un proceso de negocios permite modelar sus objetivos describiendo los pasos que deben ejecutarse para lograrlos y como también el orden, usando un diagrama de flujo. Esto mejora en gran medida la visibilidad y agilidad de la lógica del negocio, resulta en representaciones de alto nivel y específicas de dominio que pueden ser entendidas por los usuarios de negocios y es más fácil de monitorear.

El núcleo de JBPM es un motor de flujo de trabajo ligero y extensible escrito en Java puro que le permite ejecutar procesos de negocios utilizando la última especificación BPMN 2.0. Se puede ejecutar en cualquier entorno Java, incrustado en su aplicación o como un servicio.

Además del motor central, se ofrecen muchas funciones y herramientas para respaldar los procesos de negocio a lo largo de todo su ciclo de vida.

Bonita Workflow [30]

Bonita BPM es un paquete ofimático para la Gestión de procesos de negocio (BPM) y realización de Flujos de trabajo, creada en 2001. Es código abierto y puede ser descargado bajo GPL v2. Desde 2009, el desarrollo de Bonita está soportado por una empresa dedicada a esta actividad llamada Bonitasoft.

Bonita BPM está compuesta de 3 partes principales:

- **Bonita Studio:** permite al usuario modificar gráficamente los procesos de negocio siguiendo el estándar BPMN. Éste puede también conectar procesos a otras piezas del sistema de información (tales como: mensajería, ERP, ECM, bases de datos, etc.) para generar una aplicación de negocios autónoma accesible como formulario web. Bonita Studio permite también al usuario diseñar gráficamente el formulario web que será mostrado al usuario final para interactuar con el proceso. Además, Bonita Studio permite al usuario comenzar con procesos diseñados con otros estándares y tecnologías tales como XPD L o jBPM. Para ello, confía en Eclipse.
- **Bonita BPM Engine:** El motor BPM es una JAVA API que permite al usuario interactuar programáticamente con el proceso o los procesos. Está disponible bajo licencia LGPL. Confía en Hibernate.
- **Bonita UserExperience:** es un portal web que permite a cada usuario final gestionar en una interfaz similar a la del correo web todas las tareas y procesos en las cuales está involucrado. El portal también permite al propietario de un proceso administrarlo y obtener informes sobre procesos. Éste confía en GWT.

JIRA [31]

JIRA es una aplicación web para el seguimiento de errores, incidentes y para la gestión operativa de proyectos. Jira también se utiliza en áreas no técnicas para la administración de tareas. La herramienta fue desarrollada por la empresa australiana Atlassian. Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estatus y más tarde para el seguimiento de errores. Jira puede ser utilizado para la gestión de procesos y para la mejora de procesos, gracias a sus funciones para la organización de flujos de trabajo.

Jira está basada en Java EE que funciona en varias bases de datos y sistemas operativos. La herramienta dispone también de paneles de control adaptables, filtros de búsqueda, estadísticas, RSS y función de correo electrónico.

La flexible arquitectura de Jira permite al usuario crear ampliaciones específicas que pueden incluirse en la Jira extension library.

A pesar de que Jira es un producto comercial, se dan licencias gratis para proyectos de Código abierto, instituciones sin ánimo de lucro, organizaciones caritativas y personas individuales.

Bizagi [32]

Bizagi es una suite ofimática con dos productos complementarios, un Modelador de Procesos y una Suite de BPM.

Bizagi ProcessModeler es un Freeware utilizado para diagramar, documentar y simular procesos usando la notación estándar BPMN (Business Process Modeling Notation).

Bizagi BPM Suite es una solución de Gestión de procesos de negocio (BPM) que les permite a las organizaciones tanto ejecutar como automatizar procesos o flujos de trabajo (workflows). Existe una edición de nivel de entrada (Xpress Edition2) y dos ediciones corporativas (Enterprise .NET y Enterprise JEE).

Características:

- Bizagi ProcessModeler: es un freeware para diagramar, documentar y simular procesos de manera gráfica en un formato estándar conocido como BPMN (Business Process Modeling Notation). Los procesos y su documentación correspondiente pueden exportarse a Word, PDF, Visio, la web o SharePoint4 para compartirlos.
- Bizagi BPM Suite: consiste de dos herramientas; Bizagi Studio, el módulo de construcción y Bizagi BPM Server para ejecución y control. En Bizagi Studio el usuario define el modelo asociado al proceso de negocio (flujograma, reglas de negocio, interfaz de usuario, etc.) para la ejecución del mismo. Los modelos se guardan en una base de datos y son utilizados posteriormente en la ejecución por Bizagi BPM Server. Bizagi BPM Server ejecuta un Portal de Trabajo para los usuarios finales en un PC o cualquier dispositivo móvil.

Bizagi BPM Suite tiene varias características como: Seguimiento y monitoreo, alarmas y notificaciones, análisis de desempeño y reportes, auditoría y trazabilidad, enrutamiento de la carga de trabajo y movilidad. Bizagi BPM Suite se puede integrar con sistemas CRM y ERP.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

El despliegue de las aplicaciones, es una etapa muy importante fundamentalmente en proyectos de gran tamaño, donde los requerimientos no funcionales pasan a ser críticos y relevantes dentro de una organización. Dichos requerimientos son cumplir con un despliegue ágil, seguro y eficiente que permita entregar nueva funcionalidad a los usuarios en el menor tiempo posible sin afectar la operatoria normal del sistema. Sin embargo, notamos que desde la ingeniería de software no se le da la suficiente importancia que merece.

Durante la etapa del despliegue, vemos que se pueden aplicar las ideas que propone la metodología ágil SCRUM, haciendo algunas adaptaciones. Obteniendo muy buenos resultados en cuanto a la organización y planificación de los despliegues.

En los proyectos de gran tamaño, es crucial la utilización de una herramienta para la ejecución y gestión de las tareas que conforman un proceso de despliegue. A fin de evitar errores humanos y agilizar la comunicación entre los participantes.

El desarrollo de la primera versión de PEP, utilizando el motor de JBPM como parte de la solución para mejorar la ejecución de los despliegues fue muy positiva. Dado que generó mejoras notables comparando la forma manual en que se hacía la gestión de los despliegues previamente.

El motor JBPM, sin duda es una herramienta muy poderosa, pero para este problema en particular generó complicaciones derivadas tanto de las limitaciones de JBPM como de la infraestructura necesaria para que funcione. Esto fue una de las principales motivaciones para la planificación de una nueva versión de la herramienta PEP.

En la nueva versión, se logró mediante desarrollo propio, reemplazar la funcionalidad que aportaba JBPM a la solución. Obteniendo resultados muy buenos, dado que se logró simplificar la infraestructura necesaria para obtener las mismas prestaciones, incluso mejorando la experiencia de los usuarios. El analista de proceso pudo mejorar de manera notable el ciclo de prueba y ajuste en tiempo y esfuerzo. En tanto que la nueva vista del dashboard mejoró la experiencia de todo el público interesado en consultar el avance de los despliegues.

Durante la etapa de diseño y desarrollo de la herramienta, se aplicaron patrones de diseño los cuales otorgaron múltiples beneficios, tanto en el primer prototipo como en la nueva versión.

La utilización de nuevas tecnologías como Springboot y Angular2 para la implementación de la nueva versión de la herramienta generó beneficios notorios. Springboot por su parte facilitó el trabajo de los desarrolladores en el servidor. A su vez Angular además de organizar el código en el cliente mediante la aplicación del patrón MVC, mejoró la experiencia de los usuarios gracias a sus características de refresco de la vista y el modelo (Two-Way Data Binding).

Para finalizar es importante destacar que fue una experiencia muy positiva la incorporación tanto de metodologías ágiles como la utilización de herramientas para mejorar la etapa de despliegue, que a nuestro entender es un punto crítico en los proyectos de gran tamaño.

7.2. Trabajos futuros

Con la nueva versión de la herramienta PEP, encontramos que pueden surgir nuevos requerimientos que ayuden y potencien aún más un proceso de despliegue. Entre los cuales consideramos:

- Extender la herramienta para que se realicen tareas automáticas o semi automáticas a través de la interacción con sistemas colaborativos.
- Añadir la posibilidad de vincular procesos, de modo tal que la finalización de un proceso sea el cumplimiento de una tarea de otro proceso.
- Realizar una versión para dispositivos móviles, al menos de consulta.
- Funcionalidad de monitoreo (alertar cuando los tiempos de las tareas se pasen de máximos y mínimos estimados).
- Agregar nuevas formas de notificación, mensajes a celulares, mensajes por chat, por aplicación PEP, etc.
- Hacer una estimación de tiempo de inicio de las tareas, útil para que los usuarios sepan en que horario deben estar atentos para participar en el despliegue.

- Permitir la carga del cronograma directamente en el P.E.P. y habilitar la posibilidad de exportar a MS-Project de ser necesario. Es decir, lo inverso a como se hace ahora.
- Evaluar la aplicación de otras metodologías ágiles al proceso de despliegue.

Referencias

- [1] IEEE. IEEE Std 1074-1997. IEEE Standard 1074 for Developing Software LifeCycleProcesses. IEEE, 1998.
- [2] ISO/IEC 12207. "International Standard: InformationTechnology. Software Life Cycle Processes". ISO/IEC. Standard 12207-1995/Amd. 2008.
- [3] CMMI-DEV 1.3. Liberada en 2010. URL:
https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf
- [4] PMBOK Guide, I. P. "A Guide to the Project Management Body of Knowledge" Project Management Institute Inc. Pennsylvania, USA, 2004.
- [5] ITIL V3 Foundation Handbook, Ashley Hanna, John Windebank, Simon Adams, John Sowerby, Stuart Rance, Alison Cartlidge, TSO (TheStationery Office), 2009.
- [6] Kruchten, P. (2000). The Rational Unified Process An Introduction, Second Edition (2nd ed., p. 320). Addison Wesley.
- [7] IEEE. (2004). SWEBOK. KnowledgeCreationDiffusionUtilization (p. 204).
- [8] Mon, Alicia; Estayno, Marcelo; López Gil, Fernando; De María, Eduardo. (2011) Evaluación de la Implantación de Sistemas. WICC 2011.
- [9] Mon, Alicia; Estayno, Marcelo; López Gil, Fernando; De María, Eduardo. (2011) Definición de un proceso de implantación de sistemas. Infonor 2011.
- [10] Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press.
- [11] Wake, W. (2000). Extreme Programming Explored. Addison-Wesley.
- [12] Mon, Alicia.; López Gil, Fernando: (Agosto 2014) Guía para la Implantación de Sistemas. <http://200.69.224.205/implantacion/inicio1.asp>
- [13] Mon, A.; De María, E., Estayno, M.; López Gil, F.: Tres pilares para la Implantación de Sistemas. WICC 2012. (2012)
- [14] Convivencia de metodologías: Scrum y Rup en un proyecto de gran escala. Autores: Juan Manuel Fernández, Sebastián Cadelli - Tesis de grado en Licenciatura en Sistemas, Facultad de Informática, UNLP (Universidad Nacional de La Plata). Directora: Dra. Claudia Pons. Junio 2014.
- [15] Agilfant URL: <https://www.agilfant.com/open-source/>

[16] Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. https://es.wikipedia.org/wiki/Microsoft_Project

[17] Kie es un proyecto de código abierto, que agrupa varias herramientas para la automatización y gestión de sistemas empresariales, entre ellas JBPM. <http://www.kiegroup.org/>

[18] BPMN es una especificación que brinda una notación gráfica para especificar procesos de negocio. <http://www.omg.org/bpmn/index.htm>

[19] <https://wicket.apache.org/>

[20] Prof. Mg. Javier Bazzocco 2011. URL:
<http://catalogo.info.unlp.edu.ar/meran/opac-detail.pl?id1=5895#.WqqoxkbhDAU>

[21] The Type Object Pattern - Ralph Johnson and Bobby Woolf
URL: <http://www.cs.ox.ac.uk/jeremy.gibbons/dpa/typeobject.pdf>

[22] Design Patterns. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns. Addison Wesley, 1995.

[23] Angular 2. URL <https://angular.io/>

[24] Material. URL <https://material.angular.io/>

[25] Visual Studio Code. URL <https://code.visualstudio.com/>

[26] Gradle. URL <https://gradle.org/>

[27] Git. URL: <https://git-scm.com/>

[28] Jenkins. URL: <https://jenkins.io/>

[29] JBPM. URL: <https://www.jbpm.org/>

[30] Bonita. URL: https://es.wikipedia.org/wiki/Bonita_Open_Solution

[31] JIRA. URL: <https://es.wikipedia.org/wiki/JIRA>

[32] Bizagi. URL: <https://es.wikipedia.org/wiki/Bizagi>

[33] Spring boot. URL: <https://projects.spring.io/spring-boot/>