

**UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA**



ESCOLA TÉCNICA SUPERIOR DE INGENIERÍA

MÓDULO DE MONITORIZACIÓN

Plataforma de Interoperabilidad

Autor:

Sergio Nogueiras Fernández

Directores:

José Ángel Taboada González

Efrén Arias Jordán

Grado en Ingeniería Informática
Julio 2014

D. Efrén Arias Jordán y **D. José Ángel Taboada**, profesores del departamento de Electrónica y Computación de la Universidad de Santiago de Compostela informan:

Que la presente memoria, titulada *Módulo de Monitorización – Plataforma de Interoperabilidad*, presentada por **D. Sergio Nogueiras Fernández** para superar los créditos correspondientes al Trabajo de Fin de Grao de la titulación de Grao en Ingeniería Informática, se realizó bajo nuestra dirección en la Escuela Técnica Superior de Ingeniería de la Universidad de Santiago de Compostela.

Y para que así conste a efectos oportunos, expiden el presente informe en Santiago de Compostela, a _____:

Director,

Director,

José Ángel Taboada

Efrén Arias Jordán

Alumno

Sergio Nogueiras Fernández

Agradecimientos

Antes de iniciar la lectura de la memoria de este proyecto me gustaría agradecer a un conjunto de personas su apoyo y ayuda a la hora de realizar este proyecto:

- A J. Ángel Taboada González y Efrén Arias Jordan por el apoyo y esfuerzo dedicado.
- A mis compañeros y amigos de trabajo, por sus ánimos en el día a día.
- A mi Familia, por su cariño y ayuda para finalizar el Grado en Ingeniería Informática.

Índice de contenidos

1.	Introducción	17
1.1.	Organización del documento	20
2.	Definición del Ámbito del proyecto	23
2.1.	Bus de servicios	25
2.2.	Módulos comunes.....	26
2.3.	Portal de Interoperabilidad.....	27
2.4.	Tareas programadas.....	29
3.	Definición del Ámbito tecnológico.....	31
3.1.	Mule ESB	33
3.2.	Spring Framework	33
3.3.	Hibernate.....	33
3.4.	Quartz y Spring.....	33
3.5.	Liferay y Spring MVC Portlet	34
4.	Definición del Alcance	35
4.1.	Objetivos	37
4.2.	Criterios de aceptación	38
4.3.	Restricciones	38
4.4.	Suposiciones.....	38
4.5.	Entregables.....	38
5.	Gestión del proyecto	39
5.1.	Gestión de la configuración.....	41
5.1.1.	<i>Sistema de almacenamiento</i>	<i>41</i>
5.1.2.	<i>Almacenamiento del código.....</i>	<i>41</i>
5.1.3.	<i>Nomenclatura.....</i>	<i>42</i>
5.1.4.	<i>Elementos de la configuración y líneas base.....</i>	<i>43</i>
5.1.5.	<i>Control de cambios.....</i>	<i>44</i>
5.2.	Gestión temporal	46
5.2.1.	<i>Elección del ciclo de vida</i>	<i>47</i>
5.2.2.	<i>Planificación</i>	<i>47</i>
5.3.	Gestión de RRHH y Costes.....	53
5.3.1.	<i>Estimación de recursos humanos.....</i>	<i>53</i>
5.3.2.	<i>Costes recursos humanos.....</i>	<i>53</i>

5.3.3.	<i>Costes de material</i>	54
5.3.4.	<i>Costes indirectos</i>	55
5.3.5.	<i>Coste total proyecto</i>	55
5.4.	Plan de calidad	56
5.4.1.	<i>Seguimiento y control</i>	56
5.4.2.	<i>Calidad de código</i>	57
5.4.2.1.	<i>Formato</i>	57
5.4.2.2.	<i>Métricas y convenciones</i>	58
5.4.3.	<i>Resultados</i>	60
6.	Análisis.....	61
6.1.	Monitorización	63
6.1.1.	<i>Procedimientos e indicadores</i>	65
6.1.2.	<i>Niveles de servicio</i>	66
6.2.	Consola de monitorización.....	66
6.3.	Cambios en la Plataforma de Interoperabilidad y dependencias	67
6.4.	Casos de uso.....	68
6.4.1.	<i>Actores</i>	68
6.4.2.	<i>Escenarios</i>	68
6.4.3.	<i>Escenario 1: Procesos Batch</i>	69
6.4.4.	<i>Escenario 2: Consola de monitorización</i>	72
6.5.	Matriz de trazabilidad	83
6.6.	Requisitos funcionales.....	85
6.7.	Requisitos de diseño	89
6.8.	Requisitos de seguridad	89
7.	Diseño.....	91
7.1.	Arquitectura	93
7.1.1.	<i>Integración arquitectura Plataforma de Interoperabilidad</i>	95
7.2.	Modelo de datos	96
7.3.	Capa de acceso a datos	101
7.3.1.	<i>Patrón DAO</i>	102
7.3.2.	<i>Framework Hibernate</i>	102
7.3.3.	<i>Refinamiento del diseño de acceso a datos</i>	105
7.3.4.	<i>Diagrama de clases persistentes</i>	106
7.4.	Capa de servicios.....	107

7.4.1.	<i>Spring Framework</i>	107
7.4.2.	<i>Patrón CTO</i>	108
7.4.2.1.	<i>Dozer</i>	109
7.4.3.	<i>Servicios</i>	110
7.4.3.1.	<i>Patrón Fachada</i>	110
7.4.3.2.	<i>Servicios de Maestros</i>	111
7.4.3.3.	<i>Servicios de entidad</i>	111
7.4.3.4.	<i>Otros servicios</i>	117
7.4.4.	<i>Procedimientos</i>	118
7.4.4.1.	<i>Patrón comando</i>	119
7.5.	<i>Batches</i>	121
7.5.1.	<i>Batch de Monitorización</i>	123
7.5.2.	<i>Batch Análisis Proactivo</i>	125
7.5.2.1.	<i>Método de las Dos Medias</i>	127
7.6.	<i>Capa de presentación</i>	128
7.6.1.	<i>Patrón Modelo Vista Controlador</i>	129
7.6.2.	<i>Portlet de Consulta de monitorización</i>	129
7.6.3.	<i>Spring MVC Portlet</i>	129
7.6.4.	<i>Control de acceso</i>	133
7.6.5.	<i>Organización y diseño de las vistas</i>	135
8.	<i>Pruebas</i>	145
8.1.	<i>Pruebas unitarias y de integración</i>	147
8.2.	<i>Pruebas de aceptación</i>	148
9.	<i>Conclusiones y mejoras</i>	159
Referencias	163
Apéndice A: Manual de despliegue	165
Apéndice B: <i>Manual de usuario</i>	167
Apéndice C: <i>Material presentado</i>	187

Índice de Tablas

Tabla 1. Funcionalidades zona de Administración del Portal.....	28
Tabla 2. Nomenclatura de los elementos de la configuración.....	42
Tabla 3. Elementos de la configuración	43
Tabla 4. Planificación final- Iniciación	49
Tabla 5. Planificación final- Estudio Herramienta Nagios	50
Tabla 6. Planificación final- Análisis	50
Tabla 7. Planificación final- Diseño.....	50
Tabla 8. Planificación final- Desarrollo.....	51
Tabla 9. Planificación final- Documentación y Gestión calidad, riesgos y cambios	52
Tabla 10. Recursos humanos.....	53
Tabla 11. Costes - Empleados.....	54
Tabla 12. Costes - Datos básicos	54
Tabla 13. Costes - Porcentajes Seguridad Social.....	54
Tabla 14. Costes- Recursos subcontratados.....	54
Tabla 15. Material proyecto.....	54
Tabla 16. Coste total proyecto	55
Tabla 17. Resultados calidad código Monitorizacion-Model	60
Tabla 18. Resultados calidad código Monitorizacion-Portlet.....	60
Tabla 19. Resultados calidad código Análisis-Proactivo-Batch	60
Tabla 20. Resultados calidad código Monitorización-Batch	60
Tabla 22. Casos de uso escenario Procesos Batch	72
Tabla 24. Casos de uso Consola de monitorización	82
Tabla 25. Matriz trazabilidad Casos de uso / Requisito	84
Tabla 26. Niveles de Servicio ejemplo Método dos Medias	127

Índice de Figuras

Ilustración 1. Estructura de carpetas proyecto	41
Ilustración 2. Diagrama EDT	46
Ilustración 3. Planificación inicial	47
Ilustración 4. Planificación inicial - Iniciación/Nagios	48
Ilustración 5. Planificación inicial – Análisis/Diseño.....	48
Ilustración 6. Planificación inicial - Desarrollo/Pruebas/Documentación.....	48
Ilustración 7. Planificación inicial- Control y seguimiento	49
Ilustración 8. Configuración guardar-formatear.	58
Ilustración 9. Configuración formato HTML.....	58
Ilustración 10. Estructura monitorización PLATINT	63
Ilustración 11. Umbrales de indicadores.....	64
Ilustración 12. Diagrama casos de uso Procesos Batch.....	69
Ilustración 13. Diagrama casos de uso Consola monitorización	72
Ilustración 14. Arquitectura Módulo de Monitorización	94
Ilustración 15. Arquitectura Plataforma de Interoperabilidad / Módulo de Monitorización	95
Ilustración 16. Diagrama Entidad-Relación	97
Ilustración 17. Diagrama de tablas base de datos	100
Ilustración 18. Capa de acceso a datos	101
Ilustración 19. Patrón DAO.....	102
Ilustración 20. DAO's genéricos	103
Ilustración 21. Registro DAO	104
Ilustración 22. Diagrama de clases de persistencia	106
Ilustración 23. Capa de servicios	107
Ilustración 24. Diagrama secuencia Dozer	110
Ilustración 25. Diagrama Servicios Entidades Maestro.....	111
Ilustración 26. Jerarquía de interfaces e implementaciones Servicios Entidad	112
Ilustración 27. Diagrama servicios entidad Módulo de Monitorización	115
Ilustración 28. Diagrama servicio de Monitorización.....	117
Ilustración 29. Diagrama servicio Consulta de prueba.....	118
Ilustración 30. Patrón Comando	119
Ilustración 31. Diagrama de clases Procedimientos	120
Ilustración 32. Capa de Batches	121
Ilustración 33. Estructura Spring Batch.....	121
Ilustración 34. Batch del Módulo de Monitorización.....	122
Ilustración 35. Diagrama secuencia Monitorizar	123
Ilustración 36. Diagrama secuencia Análisis Proactivo	125
Ilustración 37. Método de las Dos Medias.....	128
Ilustración 38. Capa de Presentación.....	128
Ilustración 39. Diagrama de clases controladores genéricos.....	131
Ilustración 40. Controlador portlet	132
Ilustración 41. Vista Reglas con rol con todos los permisos	134
Ilustración 42. Vista Reglas con rol de consulta.....	134
Ilustración 43. Menú portlet Consulta monitorización	135

Ilustración 44. Vista General	135
Ilustración 45. Vista Detalle configuración	136
Ilustración 46. Vista Editar configuración.	136
Ilustración 47. Vista Búsqueda servicio.....	137
Ilustración 48. Vista de Detalle de un servicio	137
Ilustración 49. Vista de Configuración de un servicio	138
Ilustración 50. Vista Ping a servicio	139
Ilustración 51. Vista Registros	140
Ilustración 52. Vista Búsqueda incidencias	140
Ilustración 53. Vista Detalle incidencia	141
Ilustración 54. Vista Búsqueda de reglas	142
Ilustración 55. Vista de Detalle de regla.....	142
Ilustración 56. Vista de Creación de una regla.....	143
Ilustración 57. Acceso al módulo de monitorización	167
Ilustración 58. Sección General.....	167
Ilustración 59. Últimas incidencias.....	168
Ilustración 60. Sección de configuración.....	168
Ilustración 61. Edición de configuración	169
Ilustración 62. Editar fecha inicio análisis proactivo	169
Ilustración 63. Acceder a crear perfil	170
Ilustración 64. Aceptar y guardar cambios perfil	170
Ilustración 65. Acceso a editar un perfil.....	171
Ilustración 66. Aceptar y guardar cambios.....	171
Ilustración 67. Eliminar un perfil	171
Ilustración 68. Botón de activación del servicio.....	172
Ilustración 69. Selección perfil	172
Ilustración 70. Servicio activado.....	172
Ilustración 71. Acceso a editar configuración	173
Ilustración 72. Pantalla de edición de la configuración de un servicio	173
Ilustración 73. Asignar otro perfil.....	174
Ilustración 74. Añadir plantilla de prueba.....	174
Ilustración 75. Subir fichero	174
Ilustración 76. Editar o eliminar plantilla de prueba.....	175
Ilustración 77. Frecuencia de ejecución de los procedimientos	175
Ilustración 78. Regular indicadores.....	175
Ilustración 79. Acceder a detalles servicio	176
Ilustración 80. Detalles servicio.....	176
Ilustración 81. Pantalla ping servicio.....	177
Ilustración 82. Respuesta a una petición de prueba.....	178
Ilustración 83. Registros	178
Ilustración 84. Ejemplo búsqueda de registros.....	179
Ilustración 85. Creación de una regla.....	180
Ilustración 86. Destinatarios del aviso	181
Ilustración 87. Acceso a editar una regla	181

Ilustración 88. Editar una regla	182
Ilustración 89. Eliminar regla.....	182
Ilustración 90. Ver incidencias	183
Ilustración 91. Ver detalles incidencia	184
Ilustración 92. Detalles incidencia.....	184
Ilustración 93. Resolver incidencia.....	185
Ilustración 94. Confirmar resolver incidencia	185

Glosario

Debido a la gran importancia de algunos términos para la correcta comprensión de los apartados iniciales de este documento, se ha decidido anteponer el Glosario a estos apartados.

Catálogo de Productos: módulo de la Plataforma de Interoperabilidad encargado de gestionar los servicios y las aplicaciones ofrecidas por esta.

Clases persistentes: término utilizado para referirse a las clases utilizadas para mapear las tablas de bases de datos.

Cliente: representa cualquier aplicación o persona que realice una petición a la Plataforma de Interoperabilidad.

Conector: artefacto software que forma parte de la Plataforma de Interoperabilidad y que se corresponde con un punto de entrada de un consumidor.

Consulta de prueba: acción que realiza una petición a un servicio de la Plataforma de Interoperabilidad solicitando datos ficticios que han sido creados con anterioridad para realizar pruebas.

Consumidor: punto de entrada a un servicio publicado en la Plataforma de Interoperabilidad.

Datos de monitorización: conjunto de información creada por el Módulo de Monitorización. Entre esta información estarán los registros, las incidencias, las reglas...

Endpoint: representa un punto de entrada y o salida de la Plataforma de Interoperabilidad a través de una URL concreta. Consumidores y productores tendrán uno o varios.

Indicador: parámetro de un componente de monitorización que aporta información sobre el estado de este.

Interoperar: intercambio de datos e información entre distintos organismo de la Administración Pública.

Nivel de servicio: resumen de los distintos valores de un indicador en un rango de tiempo.

Ping: procedimiento a través del cual se conoce el estado de los distintos conectores a través de los cuales se ofrece un servicio.

Portal de Interoperabilidad: aplicación web a través de la cual se permite la interacción con la Plataforma de Interoperabilidad.

Procedimiento: función que calcula uno o más indicadores de un componente, registra su valor y devuelve los registros generados.

Productor: punto de salida de los servicios publicados en la Plataforma de Interoperabilidad, se corresponde con un punto de entrada en un servicio final.

Registro: almacenamiento en base de datos del valor de un indicador para un componente en una fecha dada.

Servicio: funcionalidad ofrecida por los distintos organismos de la Administración Pública. La Plataforma de Interoperabilidad será una puerta de entrada a estas funcionalidades. Un ejemplo de servicio es el de Consulta de Identidad ofrecido por la Administración General del Estado.

Tarea programada: Proceso batch de la Plataforma de Interoperabilidad encargado de alguna tarea que se deba ejecutar de manera programada.

Abreviaturas

MM: Módulo de Monitorización.

PLATINT: resumen, en este caso resumen de casos clínicos.

SCSP: Protocolo de Sustitución de Certificados en Soporte de Papel.

1. Introducción

En los tiempos en los que nos encontramos no resulta extraño dentro del ámbito de las Administraciones Públicas escuchar expresiones como “*Vuelva usted mañana*”, “*Esta no es la ventanilla correcta*” o “*Necesita usted presentar el justificante de...*”. Este conjunto de expresiones representa un claro perjuicio hacia los ciudadanos, los cuales en muchos casos acuden a una administración para realizar un trámite y al final acaban teniendo que acudir a varias administraciones hasta poder realizarlo.

Este problema se debe a que la Administración Pública no sigue una organización orientada al ciudadano, sino que es el ciudadano quien debe saber a qué organismo dirigirse para realizar sus trámites.

Ante la necesidad de reencaminar la orientación de las Administraciones Públicas hacia el servicio al ciudadano, se está produciendo una progresiva transformación de las Administraciones Públicas.

“Los sistemas interoperan, las personas se relacionan, o estaría bien que lo hicieran, ¿verdad?”. Paula Keen

Uno de los puntos más importantes de la transformación tiene que ver con la cita anterior y es que uno de los principales puntos para que la administración esté al servicio del ciudadano, es permitir que las distintas administraciones interoperen entre ellas.

En un contexto ideal, el ciudadano para realizar un trámite administrativo solo tendría que dirigirse a una única Administración Pública. Esto sería posible si las distintas Administraciones Públicas se comunicasen entre sí, con el único objetivo de ofrecer al ciudadano un servicio. En la vida real esto no es así, y por ejemplo, para dar de alta una empresa, un ciudadano tiene que tratar con la *Agencia Tributaria, Registro Mercantil, Seguridad Social...*

Las TIC son la principal herramienta para llevar a cabo esta transformación de la Administración Pública. A través de las nuevas tecnologías nace la Administración electrónica lo cual supone un gran avance dentro de las Administraciones Públicas, permitiendo la prestación de servicios a través de canales telemáticos así como el intercambio telemático de información entre administraciones.

La Unión Europea consciente de la gran ventaja que suponen las nuevas tecnologías a las Administraciones Públicas, impulsa una serie de políticas y planes de acción con el objetivo de conseguir una Administración Pública en línea e interoperable.

A nivel nacional el hito que marca un antes y un después en la transformación de las Administraciones Públicas es la Ley 11/2007 de acceso electrónico de los ciudadanos a los servicios públicos (LAECSP). Los dos objetivos principales de la LAECSP son, por un lado reconocer el derecho de los ciudadanos a comunicarse con las AAPP por medios electrónicos y por otro, obligar a las AAPP a comunicarse y compartir información por medios electrónicos.

Con el fin de normalizar el intercambio de información y la seguridad en las relaciones electrónicas a nivel estatal, se definen el Esquema Nacional de Seguridad y el Esquema Nacional de Interoperabilidad.

En el Esquema Nacional de Interoperabilidad se introduce un concepto clave para la transformación de las Administraciones Públicas, el concepto de Nodo de Interoperabilidad.

El Nodo de Interoperabilidad se define como: organismo que presta servicios de interconexión técnica, organizativa y jurídica entre sistemas de información para un conjunto de Administraciones Públicas bajo las condiciones que éstas fijan.

Con el Nodo de Interoperabilidad se introduce una nueva herramienta para facilitar la interoperabilidad entre las Administraciones Públicas. Su función será facilitar a los sistemas de información y a los procedimientos a los que éstos dan soporte, la capacidad de compartir datos e intercambiar información. Entendamos entonces que la interoperabilidad no sólo es tecnología, sino que responde a más dimensiones necesarias para crear un contexto de políticas, reglas y estándares en el que se interpreten correctamente los datos que se intercambian.

Desde las Administraciones Públicas se ha decidido dar un paso adelante respecto a la transformación de éstas e implantar una Plataforma de Interoperabilidad.

La Plataforma de Interoperabilidad será la base tecnológica que dará el soporte necesario para que las distintas Administraciones Públicas puedan intercambiar datos e información dentro de un contexto marcado por unas políticas, unas reglas y unos estándares que regulen el intercambio.

Uno de los puntos más importantes a la hora de implantar la Plataforma de Interoperabilidad, es que debe de prestar servicio las 24 horas del día, los 365 días del año. Para poder asegurar la prestación de servicio es necesario realizar una monitorización del estado de la Plataforma de Interoperabilidad.

El proyecto que a continuación se presenta se basará en la construcción de un Módulo de Monitorización para la Plataforma de Interoperabilidad. El Módulo de Monitorización deberá permitir conocer el estado de la Plataforma de Interoperabilidad así como avisar ante posibles incidentes que se pudieran producir.

1.1. Organización del documento

A lo largo de este documento se expondrán todos los procesos y metodologías utilizados para alcanzar los objetivos del proyecto Módulo de Monitorización.

El documento se divide en los siete apartados que se presentan a continuación:

El primer apartado será la *Definición del Ámbito del proyecto*, que describirá el contexto de negocio en el cual se desarrolla el proyecto.

El segundo apartado será la *Definición del Ámbito tecnológico*, en el cuál se introducirán las principales tecnologías utilizadas en el proyecto.

El tercer apartado será la *Definición del alcance*, que establecerá los objetivos del proyecto, restricciones, suposiciones y entregables.

El cuarto apartado será el de *Gestión del proyecto*, que definirá los procesos de gestión de proyectos utilizados para controlar el desarrollo del proyecto.

El quinto apartado será el de *Análisis*, en el cual se establecerá la funcionalidad del Módulo de Monitorización a través de la definición de casos de uso y requisitos.

El sexto apartado será el de *Diseño del proyecto*, en el que se explicarán la arquitectura, los diseños, los patrones y los frameworks utilizados para la construcción del Módulo de Monitorización.

Por último tendremos un apartado de *Conclusiones*, en el que se detallarán las conclusiones obtenidas durante el desarrollo del proyecto. Además también se expondrán diferentes mejoras que se podrían añadir al Módulo de Monitorización.

2. Definición del Ámbito del proyecto

Plataforma de Interoperabilidad

El Módulo de Monitorización es un proyecto que se enmarca dentro del Proyecto de la Plataforma de Interoperabilidad de una Administración Pública.

Realizaremos una presentación del proyecto de la Plataforma de Interoperabilidad, de ahora en adelante *PLATINT*, para ver cuál es su estructura y así poder encuadrar el proyecto del Módulo de Monitorización.

La PLATINT tiene por objetivo permitir interoperar a las distintas Administraciones Públicas, intercambiando información a través del consumo y la cesión de servicios. Los organismos de la administración podrán ser parte de la PLATINT tanto como consumidores de servicios como de productores de estos. Interoperar será por lo tanto que un organismo acceda a la PLATINT para consultar un servicio que otro organismo proporciona.

Para cumplir con su función la PLATINT se estructura en cuatro partes, el Bus de Servicios, los Módulos Comunes, el Portal de Interoperabilidad y las Tareas programadas.

2.1. Bus de servicios

El Bus de servicios es el núcleo de la Plataforma de Interoperabilidad, se trata del componente encargado de procesar las peticiones de consumo de servicios, haciendo posible el tratamiento intermedio de estas y gestionando la comunicación entre productores y consumidores de la información. Proporciona las siguientes funcionalidades principales:

- Conectividad: empleando el estándar SCSP versión 3 a través de webservices tipo SOAP.
- Encaminamiento: actúa mediando entre consumidores y productores encaminando los mensajes basándose en reglas definidas o en el contenido de los mensajes.
- Gestión de mensajes: manejando cifrado, firmado, transformaciones, colas, priorización, reintentos, etc.
- Calidad en las comunicaciones: gestionando errores en entregas, mensajes mal formados, etc.

El modelo bajo el que se realiza la comunicación puede ser síncrono o asíncrono, en función de si se trata de consultas unitarias que deben responderse online (síncrono) o si las consultas pueden ser diferidas y agrupar múltiples peticiones (asíncrono).

La publicación de un servicio en el Bus se basa en la creación de un conector en la entrada del Bus que reciba las peticiones para dicho servicio, y en la configuración de los datos del servicio a través del Catálogo de Productos. Un servicio puede contar con varios conectores de entrada y con varios conectores finales del productor. En caso de que un servicio tenga varios conectores de productor, el Bus puede reencaminar las peticiones a uno u otro.

Cuando el Bus recibe una petición, ésta se valida y en caso de ser correcta se le aplican las modificaciones oportunas para posteriormente reencaminarla al servicio final. Un ejemplo de modificación que se le puede aplicar a la petición antes de reencaminarla puede ser firmarla.

Durante su ejecución, el Bus tiene en cuenta también el tipo de información que se intercambia, con el fin de establecer unos niveles de seguridad y auditoría que aseguren un correcto tratamiento de los datos.

2.2. Módulos comunes

La PLATINT cuenta con una serie de módulos, cada uno de ellos destinado a cumplir una funcionalidad dentro de la plataforma. Estos módulos son utilizados tanto por el Bus de servicios como por el Portal de Interoperabilidad. A continuación se presentan los distintos módulos y su funcionalidad.

Catálogo de Productos

Dentro de la Plataforma de Interoperabilidad los servicios y aplicaciones son identificados como Productos. Se considera Producto aquel servicio o aplicación de interoperabilidad que permite el intercambio de datos o documentos entre Administraciones Públicas.

El Catálogo de Productos es el módulo encargado de tener identificados y clasificados los productos. Actúa como repositorio central de información común de los productos de interoperabilidad que están disponibles desde la plataforma.

La funcionalidad que ofrece el catálogo es:

- Contiene los detalles de los productos a los que pueden acceder los organismos, información descriptiva, funcional y técnica.
- Permite la configuración de los servicios, esto permitirá al bus adaptar su comportamiento en función del tipo de organismo productor de los datos, el formato del mensaje de petición o la modalidad de consumo.
- Actúa de pieza central que cohesiona el resto de módulos, ofreciendo información a los mismos en tiempo de ejecución de forma que permite “configurar” el comportamiento de la plataforma.

Catálogo de Organismos

El Catálogo de Organismos es el módulo de la Plataforma de Interoperabilidad que almacena la información de los organismos registrados. La información de organismos registrada en el Catálogo es empleada en diferentes módulos comunes de la Plataforma de Interoperabilidad para:

- Definir organismos responsables de productos de interoperabilidad del Catálogo de productos.
- Registrar los organismos autorizados a realizar consultas al Nodo de Interoperabilidad en el módulo de Seguridad.
- Realizar las comprobaciones necesarias para asegurar la autenticidad y autorización de los organismos.

Módulo de Seguridad

El Módulo de Seguridad es el módulo responsable de gestionar la autenticación y la autorización de las peticiones de consumo de servicios por parte de los organismos y clientes que acceden a la plataforma.

Los clientes son tanto las aplicaciones consumidoras de servicios como los usuarios que acceden directamente al Portal de Interoperabilidad. Para el control de usuarios el módulo de seguridad se basa en la declaración de roles y autorizaciones.

Las principales funcionalidades de este módulo son:

- Autenticación: Proceso que comprueba que un cliente es quien dice ser.
- Autorización: Proceso que comprueba que un cliente (previamente autenticado) tiene acceso a una aplicación o servicio determinado.

Módulo de Auditoría

El Módulo de Auditoría es el módulo responsable de gestionar la trazabilidad de las transacciones de interoperabilidad del bus de servicios.

Las funcionalidades del módulo emanan directamente de la resolución de 28 de junio de 2012, de la Secretaría de Estado de Administraciones Públicas, por la que se aprueba la Norma Técnica de Interoperabilidad de Protocolos de intermediación de datos. En dicha resolución se establecen las medidas, de conformidad con el Real Decreto 3/2010, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica, de trazabilidad y auditoría de los intercambios de información entre distintos organismos.

El objetivo principal del módulo es facilitar la identificación de las transacciones que se realizan en la Plataforma de Interoperabilidad.

2.3. Portal de Interoperabilidad

El Portal de Interoperabilidad es el frontal web de la Plataforma de Interoperabilidad, facilitando el acceso y uso por parte de usuarios de los productos de la plataforma. Proporciona un interfaz web de los componentes de interoperabilidad ofreciendo a los usuarios un espacio adaptado a su trabajo y componentes que pueden utilizar.

Con el Portal de Interoperabilidad se pretende ofrecer una ventana de acceso común para los diversos organismos que utilicen los servicios de la Plataforma de Interoperabilidad. El Portal supone una ventaja más a la hora de utilizar los servicios de la Plataforma ya que, evita que cada organismo tenga que realizar un desarrollo para poder consumir los servicios.

Actualmente el Portal se encuentra dividido en dos partes, por un lado la aplicación de Consulta de Servicios y por otro la zona de Administración de la Plataforma de Interoperabilidad.

Aplicación de Consulta de Servicios

Entre las aplicaciones del Portal destaca la Aplicación de Consulta de Servicios. Esta aplicación permite el acceso a los usuarios a los servicios de interoperabilidad de la plataforma.

A través de la aplicación el usuario puede solicitar la consulta de los servicios que le han sido autorizados y obtener la respuesta de servicios síncronos y asíncronos.

Un escenario de uso de la aplicación de Consulta de Servicios podría ser cuando un funcionario necesitase verificar la identidad de un ciudadano o su título de familia numerosa. Lo que haría el funcionario sería acceder con sus credenciales al Portal, entrar en la zona de Consulta de servicios y seleccionar el Servicio de Consulta de Identidad o el de Título de Familia Numerosa. Se le mostraría un formulario para que rellenase con los datos del ciudadano y se enviaría la petición al servicio ofrecido por el organismo responsable de este.

De este modo lo que se está facilitando es que un funcionario de un organismo pueda consultar información ofrecida por otro organismo sin necesidad de obligar al ciudadano a desplazarse de un lado a otro en busca de la información.

Zona de Administración

La Zona de Administración del Portal de Interoperabilidad permite realizar de forma centralizada la administración de los diferentes módulos que componen la plataforma. La Zona de Administración cuenta con tantas divisiones como módulos tiene la Plataforma de Interoperabilidad.

En la siguiente tabla se muestran las divisiones de la Zona de Administración junto con las operaciones que se pueden realizar desde cada una de ellas:

CATÁLOGO DE PRODUCTOS	
Alta de un nuevo servicio/aplicación	Establecer desactivación programada
Activación de un servicio/aplicación	Búsqueda de un productos
Desactivación de un servicio/aplicación	Consulta detalles de un producto
Modificación de un servicio/aplicación	
SEGURIDAD	
Alta/Baja/Editar roles	Alta/Baja/Editar procedimientos
Alta/Baja/Editar autorizaciones	Alta/Baja/Editar aplicaciones cliente
Alta/Baja/Editar usuarios	
CATÁLOGO DE ORGANISMOS	
Alta de un nuevo organismo	Búsqueda de organismos
Baja de un organismo	Consulta de organismo
Modificación de un organismo	
AUDITORÍA	
Búsqueda de auditorías	Configuración de parámetros de auditoría
Búsqueda de trazas	

Tabla 1. Funcionalidades zona de Administración del Portal

2.4. Tareas programadas

La Plataforma de Interoperabilidad cuenta con una serie de procesos que se ejecutan automáticamente con el fin de ayudarla a cumplir su funcionalidad. Las tareas programadas que se están ejecutando actualmente son:

Auditoría eliminación

Por ley se deben conservar las auditorías por un cierto tiempo, actualmente 2 años. El proceso de eliminación de auditorías revisa las auditorías y elimina aquellas que no sea necesario conservar.

Auditoría securización

La función de este proceso es sellar las auditorías para asegurar que no fueron modificadas desde cierta fecha.

Auditoría volcado

Con el volcado de auditorías se traspasan las auditorías a la tabla de auditorías históricas.

Solicitud de respuesta

Proceso que comprueba si el Bus de servicios ya tiene la respuesta a una petición asíncrona, de ser así avisa al usuario.

El proyecto del Módulo de Monitorización en este contexto de la Plataforma de Interoperabilidad se adaptará como un módulo común con un conjunto de tareas programadas y una interfaz web disponible a través de la zona de administración del Portal de Interoperabilidad. Su misión dentro de PLATINT será vigilar el correcto funcionamiento de todos los componentes, permitiendo asegurar la calidad de servicio de PLATINT.

3. Definición del Ámbito tecnológico

Introducción

Después de haber descrito el ambiente de negocio del Módulo de Monitorización, en este apartado se describirá el conjunto de tecnologías con las que el Módulo de Monitorización se tendrá que integrar y que guiarán en cierta medida el desarrollo de este.

Todos los componentes de la Plataforma de Interoperabilidad se despliegan sobre un cluster de servidores Weblogic. La base de datos utilizada para dar soporte a la Plataforma es una base de datos Oracle. A continuación se definirán las tecnologías utilizadas y se relacionaran con los componentes de PLATINT.

3.1. Mule ESB

Mule Enterprise Server Bus es un servidor de mensajería ligero basado en Java y que permite la conexión de aplicaciones de manera rápida y sencilla permitiendo el intercambio de mensajes.

Se ha decidido utilizar Mule para la construcción del Bus para poder definir un artefacto de control de flujos de servicios que sea adaptable a diferentes tipos de servicio. Los servicios publicados en el bus de servicios pasarán por un artefacto conector encargado de validar las peticiones y prepararlas para el procesamiento común que se realizará en el Core del mismo.

La principal ventaja que aporta Mule es la posibilidad de poder definir un control de flujos de servicio centralizado e independiente de los tipos de servicios que se estén tratando.

3.2. Spring Framework

Para el desarrollo de los distintos módulos de PLATINT se utiliza el framework Spring, que es un framework de código abierto para plataformas Java. La funcionalidad más utilizada del framework dentro del desarrollo de los módulos es la implementación del patrón de Inversión de control. Este patrón permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java, utilizando la inyección de dependencias.

3.3. Hibernate

Aunque en la introducción al ámbito del proyecto no se nombró, todos los módulos comunes utilizan un módulo que representa las entidades de PLATINT y el acceso a datos. Para implementar esta capa de persistencia se utiliza el framework Hibernate.

Hibernate es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML o anotaciones en las clases Java.

3.4. Quartz y Spring

Quartz es un framework para la planificación y gestión de tareas. Para la implementación de las Tareas Programadas se utiliza el framework Quartz integrado con Spring, esto permite definir procesos batch que se ejecutan cada cierto tiempo para realizar funcionalidades de la PLATINT.

3.5. Liferay y Spring MVC Portlet

El Portal de Interoperabilidad se basa en la tecnología de Liferay y Spring MVC Portlet.

Liferay es un portal web que permite la implementación de aplicaciones y sitios web junto con la gestión de contenidos. PLATINT basa su Portal de Interoperabilidad en una instancia de Liferay a través de la cual se permite el acceso a la web que contiene las distintas aplicaciones.

Para el uso que PLATINT hace de Liferay es imprescindible el concepto de Portlet. Un Portlet es una aplicación web igual que cualquier otra aplicación web que se pueda utilizar para desplegar en un servidor de aplicaciones como Tomcat o Glashfish, pero con la diferencia de que se despliega en un contenedor de portlets, en este caso Liferay.

Un Portlet es por lo tanto una aplicación web que se puede situar en cualquier parte de la web del Portal de Interoperabilidad. Como aplicaciones web que son, los portlets se encapsulan en archivos .war para ser desplegados.

Para el desarrollo de los Portlets se utiliza el framework Spring MVC Portlet, el cual aporta funcionalidades como la inyección de dependencias o la implementación del patrón modelo vista controlador a través del uso de anotaciones en las clases java.

4. Definición del Alcance

Introducción

Adentrándonos ya por completo en el Módulo de Monitorización, de ahora en adelante MM, lo primero que haremos será describir el alcance del proyecto, así como, sus objetivos, criterios de aceptación, restricciones, suposiciones y entregables.

El Módulo de Monitorización será el responsable de vigilar el correcto funcionamiento de la Plataforma de Interoperabilidad. Para cumplir su función, el Módulo de Monitorización deberá realizar un seguimiento de aquellos parámetros que informen acerca del estado del sistema.

La formalización de los parámetros que dan información acerca del estado se realizará por medio de Indicadores. Los indicadores contarán con una serie de umbrales que delimitan los distintos estados en los que puede encontrarse un indicador. Al realizar su seguimiento se utilizarán estos umbrales para deducir si la medida obtenida corresponde a un estado u a otro.

Para realizar el seguimiento de los indicadores se utilizarán Procedimientos. Los Procedimientos serán ejecutados automáticamente cada cierto tiempo y calcularán una serie de indicadores.

Además de controlar el estado actual del sistema, el MM realizará una función de análisis proactivo que consistirá en calcular los niveles de servicio para detectar posibles degradaciones de este.

Los niveles de servicio permitirán dar una dimensión temporal a los indicadores, por ejemplo, si tenemos un indicador que informe del tiempo que tarda un servicio en resolver una petición, el nivel de servicio correspondiente a este indicador sería la media de tiempo que tarda un servicio en resolver una petición durante una semana. Al calcular los niveles de servicio se podrán detectar tendencias que sirvan para alertar de manera proactiva ante posibles degradaciones de servicio.

Cuando se produzcan situaciones que puedan perjudicar el correcto funcionamiento del sistema, el MM tendrá que avisar a los usuarios responsables en cada caso. Estas situaciones anómalas se corresponderán con indicadores que se encuentran en un umbral de alerta o niveles de servicio que siguen una tendencia de degradación.

Para realizar la administración y permitir el acceso a los datos de monitorización se dispondrá de una aplicación web. Esta aplicación web consistirá en un portlet que se integre con el resto de portlets del Portal de Interoperabilidad.

A través del portlet de MM se podrán visualizar los registros de los indicadores, visualizar las incidencias registradas, configurar los indicadores y configurar la monitorización.

Como funcionalidades adicionales, el portlet de MM permitirá realizar consultas de prueba a los servicios desplegados en la PLATINT, comprobar en tiempo real el estado de los servicios y exportar los datos de monitorización.

4.1. Objetivos

Con el fin de resumir el alcance del Módulo de Monitorización se marcan los siguientes objetivos que el Módulo de Monitorización debe cumplir:

- Realizar un seguimiento de los componentes de la Plataforma de Interoperabilidad con el fin de conocer su estado en cada momento. Este seguimiento se deberá de realizar automáticamente durante las 24 horas del día los 365 días del año.
- Analizar los niveles de servicio ofrecidos por cada componente para detectar posibles degradaciones de servicio.
- En caso de que se detecte un estado de alerta en alguno de los componentes se deberá enviar un aviso a los responsables.
- Se dispondrá de un interfaz web a través de la cual se podrá:
 - Visualizar y exportar los datos de monitorización.
 - Configurar el Módulo de Monitorización.
 - Realizar consultas de prueba a los servicios.
 - Comprobar el estado de los servicio en tiempo real.
 - Controlar el acceso al interfaz web.

4.2. Criterios de aceptación

El proyecto Módulo de Monitorización será aceptado si se realiza un seguimiento de todos los componentes de la Plataforma de Interoperabilidad, enviando avisos cada vez que se encuentre un componente en un estado de alerta.

Además se deberán de calcular y analizar los niveles de servicio para alertar ante tendencias de degradación de éstos.

Por último, el interfaz web deberá cumplir con toda la funcionalidad definida en los objetivos y requisitos del Módulo de Monitorización.

4.3. Restricciones

- *Temporal*: el proyecto debe constar de 420 horas de trabajo.
- *Código*: el código a entregar como Proyecto de Fin de Grado no se podrá ejecutar debido a que solo se podrá entregar el código implementado por Sergio Nogueiras Fernández para el Módulo de Monitorización. Al no entregar el código de los módulos comunes que utiliza el MM y no contar con los permisos de acceso a infraestructura y servicios de PLATINT el código no se podrá compilar ni ejecutar.

4.4. Suposiciones

El equipo de desarrollo de la Plataforma de Interoperabilidad realizará los cambios necesarios en esta con el fin de facilitar al Módulo de Monitorización la obtención de datos acerca de las peticiones que se hagan a la Plataforma de Interoperabilidad.

Estos cambios se basarán principalmente en la creación de trazas, informando de los tiempos de las peticiones y su estado.

4.5. Entregables

Al finalizar el proyecto se deberán entregar los siguientes elementos:

- Código de fuente
- Scripts de creación de base de datos
- Memoria
- Manual de usuario y de despliegue

5. Gestión del proyecto

5.1. Gestión de la configuración

La gestión de la configuración permite establecer y mantener la integridad de los productos de trabajo utilizando la identificación de la configuración y el control de ésta.

En los siguientes sub-apartados se podrá encontrar información acerca del sistema de almacenamiento, de la nomenclatura utilizada, de los elementos de la configuración, del sistema de control de cambios y de las herramientas utilizadas en el proyecto.

5.1.1. Sistema de almacenamiento

Para almacenar todos los elementos del proyecto se utilizará un sistema de carpetas en la nube lo cual permitirá evitar la pérdida de los datos debido a problemas en algún componente físico de los equipos del proyecto. Además este sistema de almacenamiento en la nube permitirá compartir los elementos del proyecto con el resto de personas implicadas.

La estructura de carpetas que se utilizará será la siguiente:

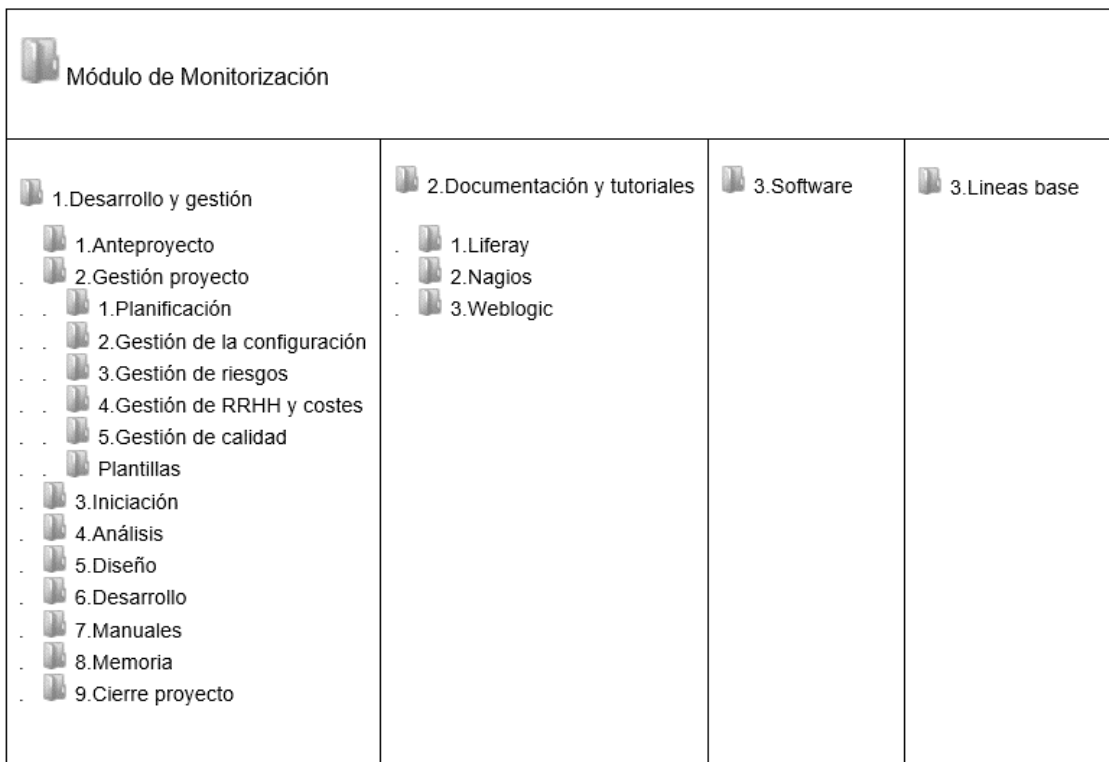


Ilustración 1. Estructura de carpetas proyecto

5.1.2. Almacenamiento del código

En la fase de desarrollo además de utilizar el almacenamiento en la nube se hará uso del repositorio SVN que se utilizó previamente para desarrollar la Plataforma de Interoperabilidad y en el que están presentes diversos proyectos evolutivos de la Plataforma de Interoperabilidad, entre los que se encuentra el proyecto MM. El uso de este repositorio SVN permitirá llevar un control de las distintas versiones del código del proyecto.

Para la utilización del repositorio se utilizará un plugin dentro del entorno de desarrollo.

5.1.3. Nomenclatura

El nombre de los documentos seguirá el siguiente formato:

[MM-<Código>]-< Nombre >-< versión >

En el campo código se especificará el código del tipo de documento que podrá ser obtenido a través de la tabla de nomenclatura.

Además de la nomenclatura general existen una serie de elementos que tendrán una nomenclatura específica estos son:

- Líneas bases: no se utilizará el campo *versión* y el campo *nombre* será la fecha (DDMMAAA) de liberación de la línea base.
- Actas de reunión: en lugar de *versión* se pondrá la fecha de la reunión en formato (DDMMAAAA).

A continuación se presenta una tabla con los tipos de documentos y sus respectivos códigos.

Nombre	Descripción	Código
Memoria proyecto	Documento general del proyecto que integrará a los demás documentos del proyecto con el fin de ofrecer una visión completa y conjunta de todo el proyecto.	MP
Gestión de la configuración	Documento de texto dónde se identifican los elementos de la configuración así como el sistema de gestión de la configuración y el de control de cambios.	GC
Historial de cambios	Hoja de cálculo que contiene los cambios solicitados, el autor del cambio, estado del cambio y la repercusión del mismo.	HC
Planificación	Permite ver una visión global del plazo de ejecución del proyecto.	PLF
Plan de Calidad	Documento que planifica los procesos que se llevarán a cabo para realizar un seguimiento del proyecto y además define buenas prácticas para asegurar el cumplimiento de los objetivos del proyecto.	PC
Plan de pruebas	Documento en el que se definen las pruebas a realizar.	PP
Plan de Costes y Recursos Humanos	Planificación de los costes y los recursos humanos.	PCRH
Historial de riesgos	Hoja de cálculo con el historial de riesgos identificados en el proyecto.	HR
Análisis	Documento que recoge los requisitos, el modelo funcional, los casos de uso y las entidades identificadas en el proyecto	AF
Diseño	Documento que describe la arquitectura del sistema, el modelo de datos, la capa de servicios, la capa de acceso a datos y la capa cliente.	DT

Tabla 2. Nomenclatura de los elementos de la configuración

La nomenclatura de los proyectos Java que se crean para implementar el Módulo de Monitorización será la utilizada en el resto de proyectos de la Plataforma de Interoperabilidad:

- Proyecto de modelo: [Nombre]-model
- Proyecto de portlet: [Nombre]-portlet
- Proyecto batch: [Nombre]-batch

5.1.4. Elementos de la configuración y líneas base

Dentro del ciclo de vida del proyecto existen diversos elementos a los que es necesario realizar un seguimiento con el fin de controlar los cambios que se producen sobre dichos elementos y además asegurar que siempre esté disponible una versión estable del mismo para cualquier persona que lo solicite.

Los elementos de la configuración identificados para el proyecto son los siguientes:

Nombre	Tipo documento
[MM-GC]-GestiónConfiguración-version.doc	Documento de Word 97-2003
Directorio	
[MM]\1.Desarrollo y gestión\2.Gestión proyecto\2.Gestión de la configuración	

Nombre	Tipo documento
[MM-PLF]-Planificacion-version.mpp	Microsoft Project 2007
Directorio	
[MM]\1.Desarrollo y gestión\2.Gestión proyecto\1.Planificación	

Nombre	Tipo documento
[MM-PC]-PlanCalidad-version.doc	Documento de Word 97-2003
Directorio	
[MM]\1.Módulo de monitorización\2.Gestión proyecto\5.Gestión de calidad	

Nombre	Tipo documento
[MM-PCRH]-PlanCostesyRRHH-version.doc	Documento de Word 97-2003
Directorio	
[MM]\1.Módulo de monitorización\2.Gestión proyecto\4.Gestión de RRHH y costes	

Nombre	Tipo documento
[MM-PP]-PlanPruebas-version.doc	Documento de Word 97-2003
Directorio	
[MM]\1.Módulo de monitorización\2.Gestión proyecto\5.Gestión de Calidad	

Nombre	Tipo documento
[MM-ANA]-Análisis-version.doc	Documento de Word 97-2003
Directorio	
[MM]\ 1.Módulo de monitorización\4.Análisis	

Nombre	Tipo documento
[MM-DIS]-Diseño-version.doc	Documento de Word 97-2003
Directorio	
[MM]\ 1.Módulo de monitorización\5.Diseño	

Tabla 3. Elementos de la configuración

Durante el desarrollo del proyecto se van creando los elementos de la configuración citados anteriormente, cada uno en la fase correspondiente según la planificación temporal. Al final de cada fase del proyecto se realizarán unas tareas de seguimiento, es en este punto en el que se libera una nueva línea base.

Liberar una línea base supone integrar todos los nuevos documentos que se hayan generado o modificado desde la anterior línea base en la *Memoria del proyecto*. Para facilitar esta tarea se utilizará el control de cambios que se explica en el apartado de *Control de cambios*.

Además de la *Memoria del proyecto* la línea base también contendrá el código implementado hasta el momento, para ello se descargará la última versión del código y se comprimirá junto con la *Memoria del proyecto* en un archivo zip que representará la línea base.

Las actas de reunión, el historial de cambios y el historial de riesgos no son considerados elementos de configuración. El motivo es que la vida de las actas de reunión es muy corta y que los otros dos historiales son archivos de log.

5.1.5. Control de cambios

Uno de los puntos importantes de la gestión de la configuración es el control del cambio ya que el problema no es que se produzcan cambios sino que se produzcan de forma incontrolada.

Para mantener los cambios controlados el proceso que se seguirá será el siguiente:

- **Realizar petición de cambio**

Se dispondrá del documento *[MM-HC]-HistorialCambios-version.gsheets* para realizar las peticiones de cambio. Este documento aunque pertenece al sistema de ficheros común del proyecto es una hoja de cálculo de Google y como tal puede ser compartida con distintas personas. Se utilizará esta posibilidad para que todo el mundo relacionado con el proyecto pueda realizar una petición de cambio.

Para realizar la petición de cambio solamente será necesario cubrir la descripción del cambio, el nombre del solicitante y la fecha de solicitud. Automáticamente cuando se solicite un nuevo cambio el responsable del proyecto será notificado mediante un email.

- **Evaluar petición de cambio**

El responsable del proyecto será el encargado de evaluar todas las peticiones de cambio, deberá de analizar los elementos del proyecto, el coste del cambio y finalmente decidir si se realiza o no el cambio.

El responsable del proyecto decidirá cuándo evaluar las peticiones de cambio teniendo como única restricción que antes de las tareas de seguimiento y control deben estar todas las peticiones anteriores evaluadas y aplicadas.

- **Realizar cambio**

Todos los cambios que hayan sido aprobados serán realizados por el responsable del proyecto. Dependiendo de la dimensión del cambio se valorará si es necesario reflejarlo en la planificación del proyecto o no.

Los elementos de la configuración irán aumentando su versión a medida que se vayan aplicando los cambios, pudiendo concentrar varios cambios en un solo cambio de versión. Además del historial de cambios cada elemento de la configuración tendrá su tabla de cambios en la que

quedarán reflejadas las versiones y sus fechas, lo cual permitirá ver qué cambios hay que incluir en la memoria de las líneas base.

Por último el control de cambios y versiones de las distintas implementaciones se realizará utilizando el repositorio SVN al que accederemos con el plugin del entorno de desarrollo y que registrará todos los cambios.

5.2. Gestión temporal

Para gestionar los tiempos del proyecto se comenzará por identificar los trabajos necesarios mediante la realización de un EDT. Con el EDT realizado se creará un diagrama de Gantt para realizar la estimación temporal de las tareas.

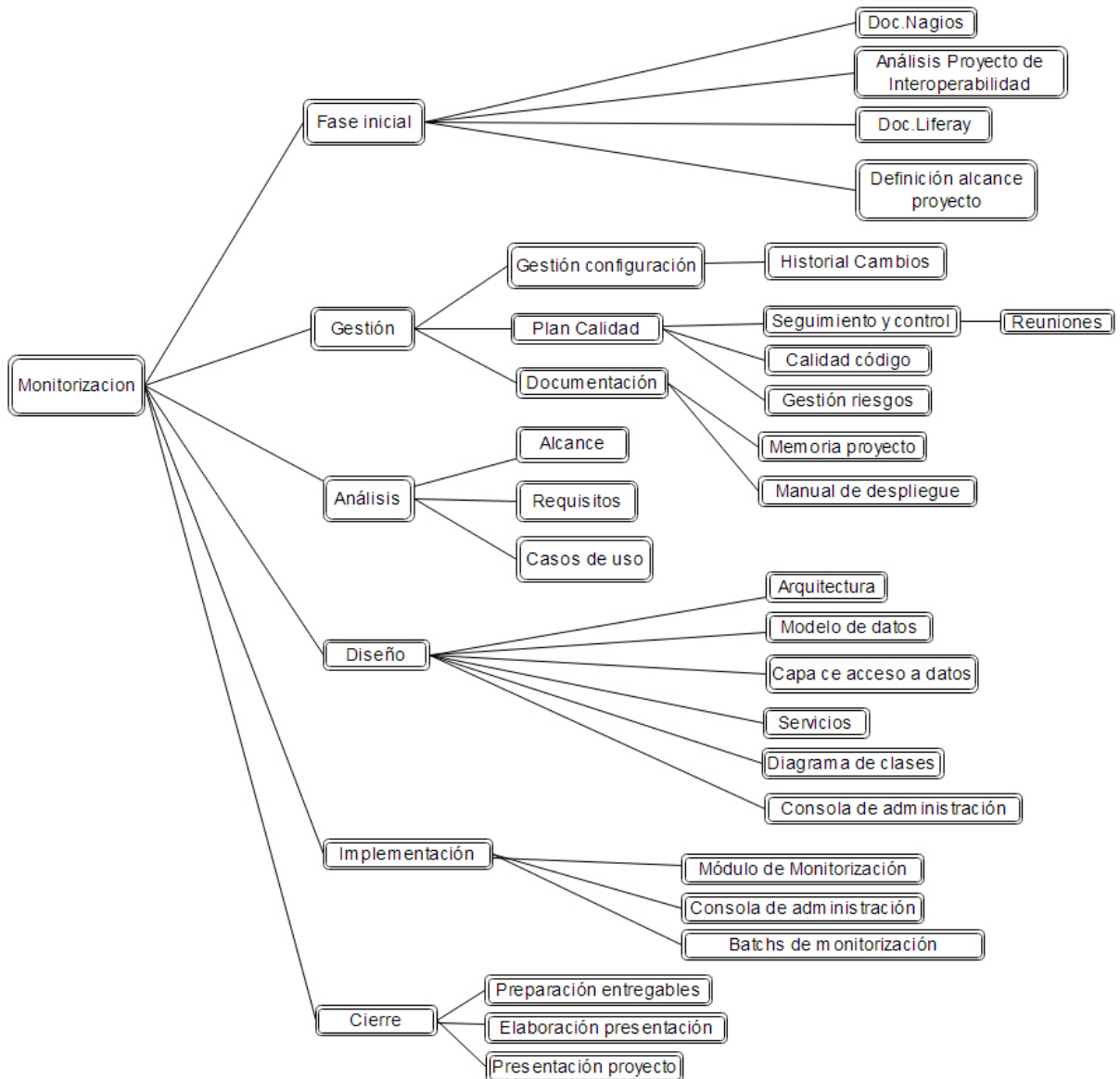


Ilustración 2. Diagrama EDT

5.2.1. Elección del ciclo de vida

Desde el punto de vista del Proyecto de la Plataforma de Interoperabilidad el ciclo de vida utilizado es un ciclo por incrementos. Sin embargo, a la hora de hablar del Módulo de Monitorización como proyecto, el ciclo de vida elegido es en cascada.

Se utiliza un ciclo de vida distinto debido a que el desarrollo del Módulo de Monitorización se corresponderá con un incremento del proyecto que lo engloba, y como tal, constará de un solo entregable.

El ciclo de vida en cascada destaca por su sencillez en la gestión, ya que todas las actividades se desarrollan de manera lineal. El desarrollo secuencial de las actividades también favorece la estimación temporal, ya que, se basa exclusivamente en la suma de la estimación de cada actividad.

Uno de los posibles inconvenientes que podría tener utilizar un ciclo de vida en cascada sería que si no se cuenta con los todos requisitos al inicio del proyecto, se pueden producir retrasos muy amplios debido a que haya que retomar etapas anteriores al detectar algún error. Este inconveniente no es considerado en el Módulo de Monitorización debido a que la extensión total del proyecto es pequeña y el alcance de este es claro.

Además el cliente tiene conocimientos técnicos que le permiten tener una comprensión clara de los requisitos y podrá validarlos sin dificultad.

5.2.2. Planificación

Respecto a la planificación temporal, el proyecto comienza el 13 de Febrero de 2014 y se estima su finalización para el 17 de Junio de 2014. Se divide la planificación inicial en 6 fases: *Iniciación, Estudio Herramienta Nagios, Análisis, Diseño, Desarrollo, Pruebas y Documentación*. Además se añadirán a esta planificación dos tareas, relacionadas con la presentación del proyecto como Trabajo de Fin de Grado, estas son *Documentación y Cierre del proyecto*, ambas tareas se basarán en preparar la Memoria del proyecto y la Presentación del mismo.

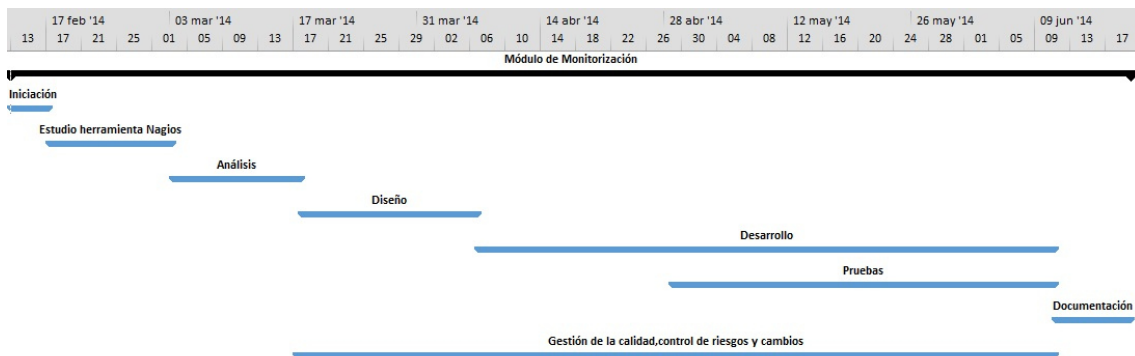


Ilustración 3. Planificación inicial

A continuación se muestra la planificación por separado de cada fase. La definición de cada tarea se pospone a la planificación final debido a que es la planificación que refleja con mayor realidad las actividades desarrolladas en el proyecto.

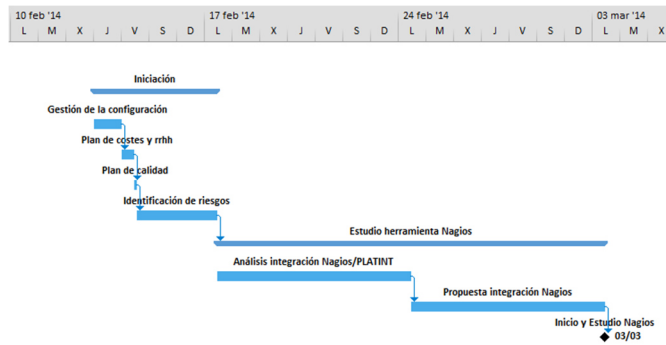


Ilustración 4. Planificación inicial - Iniciación/Nagios

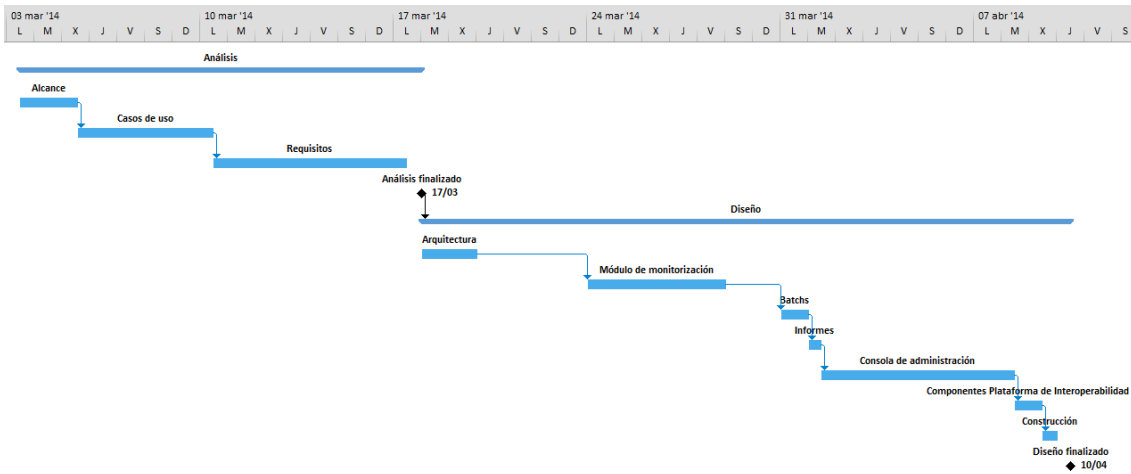


Ilustración 5. Planificación inicial - Análisis/Diseño

De los siguientes diagramas de Gantt es preciso destacar que, debido a cuestiones gráficas, tienen una escala temporal mayor que los dos anteriores.

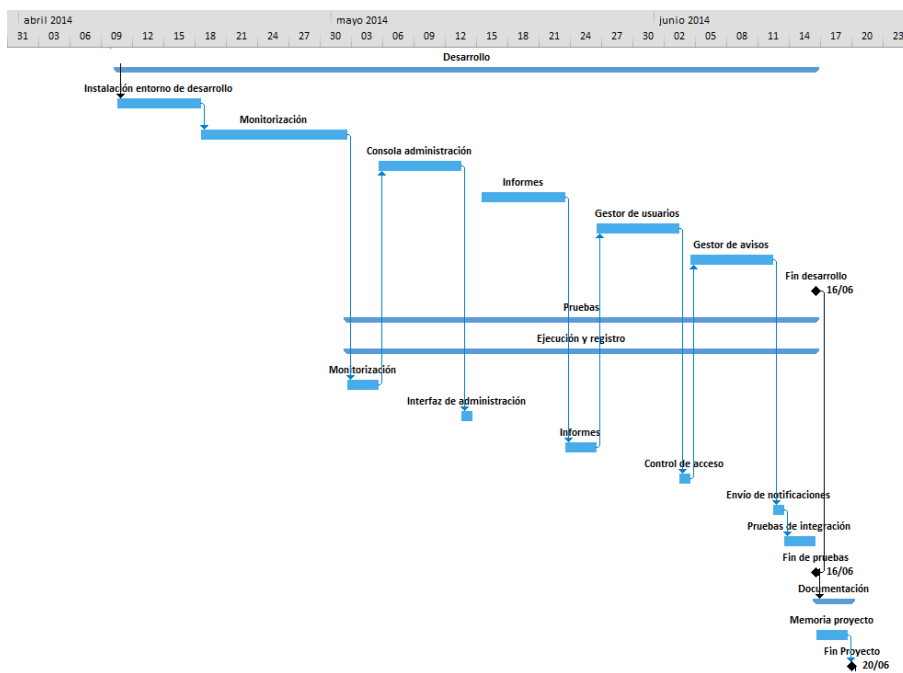


Ilustración 6. Planificación inicial - Desarrollo/Pruebas/Documentación

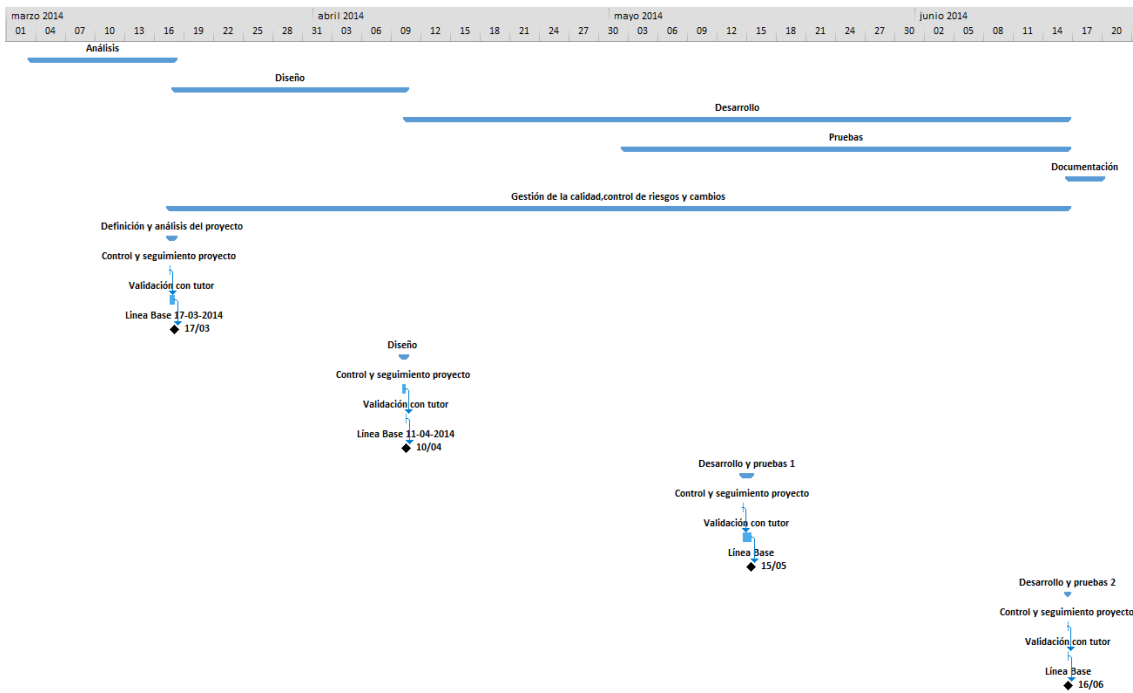


Ilustración 7. Planificación inicial- Control y seguimiento

En todos los proyectos surgen cambios durante el desarrollo, bien sea por cambios en las estimaciones, en los requisitos, en los recursos... Muchas veces estos cambios provocan que haya que volver a planificar, con el fin de no perder el hilo del proyecto y tenerlo controlado.

En el Módulo de Monitorización se produjeron cambios que provocaron tener que volver a planificar. Los cambios que más influyeron fueron:

- La necesidad de descartar la utilización de la herramienta Nagios para monitorizar la infraestructura, por no ser prioritario desde el punto de vista del MM, ya que las organizaciones cuentan con sus propias herramientas de monitorización a tal fin, pudiendo considerarse un desaprovechamiento de tiempo y recursos el incluir este aspecto en el MM y siendo preferible dedicar esos medios a la monitorización de los servicios y las tareas programadas de la Plataforma de Interoperabilidad.
- Retrasos acaecidos en las tareas iniciales, respecto a la estimación original..
- La necesidad de definir de una manera más detallada las tareas planificadas para la fase de desarrollo, a fin de poder realizar un seguimiento apropiado de esta fase del proyecto.

Para detallar la planificación final se utilizarán una serie de tablas con las tareas del proyecto y su fecha de inicio y fin. Si se desea se puede consultar el documento *[MM-PLF]-Planificación-v1.1*, adjunto a esta memoria, donde se detalla el diagrama de Gantt.

Iniciación	jue 13/02/14	lun 17/02/14
Gestión de la configuración	jue 13/02/14	jue 13/02/14
Plan de costes y RRHH	vie 14/02/14	vie 14/02/14
Plan de calidad	vie 14/02/14	vie 14/02/14
Historial de riesgos	vie 14/02/14	lun 17/02/14

Tabla 4. Planificación final- Iniciación

Las tareas iniciales del proyecto son las de Gestión de Proyecto, en las cuales se realizan los planes y estimaciones necesarios para dar comienzo al proyecto. Los productos de esta fase son de gran importancia para asegurar el desarrollo del proyecto.

Estudio Herramienta Nagios	lun 17/02/14	lun 03/03/14
Análisis integración Nagios/PLATINT	lun 17/02/14	lun 24/02/14
Propuesta integración Nagios	lun 24/02/14	lun 03/03/14
Iniciación terminada	lun 03/03/14	lun 03/03/14

Tabla 5. Planificación final- Estudio Herramienta Nagios

La fase de Estudio Herramienta Nagios es una fase dedicada al estudio de la herramienta Nagios y su posible integración con la Plataforma de Interoperabilidad. En un primer momento del proyecto se consideró utilizar Nagios para monitorizar la infraestructura de PLATINT, para ello se analizó como funciona Nagios y se creó una propuesta de cómo integrarlo. Finalmente la propuesta fue rechazada por considerar que no es prioritario monitorizar la infraestructura en el Módulo de Monitorización ya que las organizaciones cuentan con sus propias herramientas para tal fin.

Análisis	lun 03/03/14	mar 18/03/14
Alcance	lun 03/03/14	mié 05/03/14
Requisitos	mié 05/03/14	lun 10/03/14
Casos de uso	lun 10/03/14	lun 17/03/14
Análisis finalizado	mar 18/03/14	mar 18/03/14

Tabla 6. Planificación final- Análisis

La fase de Análisis tiene el objetivo de fijar el alcance del proyecto, establecer los requisitos y definir los casos de uso. Es una fase muy importante ya que al seguir un modelo en cascada realizar un mal análisis puede producir errores en etapas futuras que conlleven un gran retraso para el proyecto.

Al final de esta fase se realizará una validación del análisis por parte del cliente, con el fin de evitar posibles errores futuros.

Diseño	mar 18/03/14	jue 10/04/14
Arquitectura	mar 18/03/14	jue 20/03/14
Módulo de monitorización	lun 24/03/14	vie 28/03/14
Batches	lun 31/03/14	lun 31/03/14
Exportación	mar 01/04/14	mar 01/04/14
Consola de administración	mar 01/04/14	mar 08/04/14
Componentes Plataforma de Interoperabilidad	mar 08/04/14	mié 09/04/14
Construcción	mié 09/04/14	mié 09/04/14
Diseño finalizado	jue 10/04/14	jue 10/04/14

Tabla 7. Planificación final- Diseño

Las tareas de la fase de Diseño tienen el objetivo de definir cómo se va a construir el Módulo de Monitorización. Al final de la fase se debería de contar con todos los planos para realizar la

implementación del módulo. Un aspecto importante de la fase de diseño es el análisis del actual diseño de la PLATINT con el fin de seguir los mismos diseños.

Desarrollo	196 horas	vie 11/04/14	lun 16/06/14
Capa de acceso a datos	35 horas	vie 11/04/14	lun 21/04/14
Entidades	15 horas	vie 11/04/14	mar 15/04/14
Dao's	20 horas	mié 16/04/14	lun 21/04/14
Inicialización capa de servicios	5 horas	mié 30/04/14	mié 30/04/14
Cto's	1 hora	mié 30/04/14	mié 30/04/14
Conversores/Utilidades	1 hora	mié 30/04/14	mié 30/04/14
Servicios de maestros	3 horas	mié 30/04/14	mié 30/04/14
Configuración monitorización	38 horas	jue 01/05/14	lun 12/05/14
Servicios	20 horas	jue 01/05/14	mar 06/05/14
Configuración general	10 horas	jue 01/05/14	vie 02/05/14
Configuración componentes	10 horas	lun 05/05/14	mar 06/05/14
Portlet	18 horas	mié 07/05/14	lun 12/05/14
Procedimientos sin trazas	60 horas	lun 12/05/14	mié 28/05/14
Servicios	35 horas	lun 12/05/14	mié 21/05/14
Ping a servicios	15 horas	lun 12/05/14	jue 15/05/14
Consulta real	20 horas	jue 15/05/14	mié 21/05/14
Portlet	15 horas	mié 21/05/14	lun 26/05/14
Ping a servicios	5 horas	mié 21/05/14	jue 22/05/14
Consulta real	10 horas	jue 22/05/14	lun 26/05/14
Batch	10 horas	lun 26/05/14	mié 28/05/14
Batch monitorización	10 horas	lun 26/05/14	mié 28/05/14
Procedimientos sobre trazas	38 horas	jue 29/05/14	mar 10/06/14
Servicios	28 horas	jue 29/05/14	vie 06/06/14
Trazas peticiones	10 horas	jue 29/05/14	lun 02/06/14
Trazas tareas programadas	8 horas	lun 02/06/14	mié 04/06/14
Integración gestión de avisos	10 horas	mié 04/06/14	vie 06/06/14
Portlet	10 horas	vie 06/06/14	mar 10/06/14
Visualización registros/incidencias	10 horas	vie 06/06/14	mar 10/06/14
Análisis proactivo	15 horas	mar 10/06/14	vie 13/06/14
Servicios	10 horas	mar 10/06/14	jue 12/06/14
Análisis de los niveles de servicio	10 horas	mar 10/06/14	jue 12/06/14
Batches	5 horas	jue 12/06/14	vie 13/06/14
Batch análisis proactivo	5 horas	jue 12/06/14	vie 13/06/14
Exportación	5 horas	vie 13/06/14	lun 16/06/14
Exportación de registros e incidencias	5 horas	vie 13/06/14	lun 16/06/14
Desarrollo finalizado	0 horas	lun 16/06/14	lun 16/06/14

Tabla 8. Planificación final- Desarrollo

La fase de desarrollo comienza con las tareas de creación de entidades y de acceso a datos para después implementar las distintas funcionalidades. Para la implementación de cada funcionalidad se empezará siempre por los servicios para luego realizar la parte de las vistas o de los batch dependiendo de la funcionalidad.

Debido a la magnitud del proyecto y a su vez al límite de tiempo como TFG se ha decidido integrar las pruebas con la implementación. De este modo a la vez que se va implementando se van creando los test oportunos. Además para conocer en todo momento el estado de las implementaciones se creará progresivamente un conjunto de pruebas de regresión.

Con el anterior método quedan cubiertas tanto las pruebas unitarias como las de integración ya que los test se crearán tanto para la capa de acceso a datos como para la de servicios.

Documentación	30 horas	lun 16/06/14	mar 24/06/14
Memoria proyecto	25 horas	lun 16/06/14	lun 23/06/14
Manual despliegue	5 horas	lun 23/06/14	mar 24/06/14
Fin Proyecto	0 horas	mié 25/06/14	mié 25/06/14
Gestión de la calidad, control de riesgos y cambios	52 horas	lun 17/03/14	mié 25/06/14
Definición y análisis del proyecto	13 horas	lun 17/03/14	mar 18/03/14
Control y seguimiento proyecto	1 hora	lun 17/03/14	lun 17/03/14
Validación con tutor	12 horas	lun 17/03/14	mar 18/03/14
Línea Base	0 horas	mar 18/03/14	mar 18/03/14
Diseño	13 horas	jue 10/04/14	jue 10/04/14
Control y seguimiento proyecto	1 hora	jue 10/04/14	jue 10/04/14
Validación con tutor	12 horas	jue 10/04/14	jue 10/04/14
Línea Base	0 horas	jue 10/04/14	jue 10/04/14
Desarrollo y pruebas	13 horas	mié 28/05/14	jue 29/05/14
Control y seguimiento proyecto	1 hora	mié 28/05/14	mié 28/05/14
Validación con tutor	12 horas	mié 28/05/14	jue 29/05/14
Línea Base	0 horas	jue 29/05/14	jue 29/05/14
Desarrollo y pruebas final	13 horas	mar 24/06/14	mié 25/06/14
Control y seguimiento proyecto	1 hora	mar 24/06/14	mar 24/06/14
Validación con tutor	12 horas	mar 24/06/14	mié 25/06/14
Línea Base	0 horas	mié 25/06/14	mié 25/06/14

Tabla 9. Planificación final- Documentación y Gestión calidad, riesgos y cambios

Por último, las fases de Documentación y de Gestión de calidad, riesgos y cambios se introducen al final de la planificación con el fin de agruparlas, pero estas tareas se realizan a lo largo de todo el proyecto.

La tarea de Memoria consistirá en ir creando el documento a presentar para el TFG a medida que se los contenidos están disponibles. El manual de despliegue se realizará al final de la implementación y contará con todo lo necesario para permitir desplegar el Módulo de Monitorización en los diferentes entornos de la empresa/cliente.

Las tareas de Gestión de la calidad, riesgos y cambios se realizan en puntos estratégicos de la planificación con el fin de tener controlado el desarrollo del proyecto.

5.3. Gestión de RRHH y Costes

En esta apartado se comenzará estimando los recursos humanos necesarios para el proyecto y se finalizará calculando la estimación de los costes en función de los recursos humanos, el material y los costes indirectos.

5.3.1. Estimación de recursos humanos

En la gestión de recursos humanos se estiman las personas necesarias para realizar el proyecto, el rol que desempeñan dentro de este, su tiempo de participación en el mismo y su coste.

Para el Módulo de Monitorización se contará con cuatro recursos humanos, tres de ellos participarán realizando funciones de seguimiento y dirección y el otro será el responsable del proyecto.

Nombre	Rol	Participación
Jefe proyecto PLATINT	Jefe de proyecto en la empresa	No se planifica la participación ya que esta se basará en pequeñas intervenciones desde el inicio al fin del proyecto.
José Ángel Taboada	Director	Participa en las tareas de evaluación y seguimiento programadas en la planificación del proyecto.
Efrén Arias Jordan	Director	Participa en las tareas de evaluación y seguimiento programadas en la planificación del proyecto.
Sergio Nogueiras Fernández	Responsable proyecto	Participa en todas las tareas del proyecto, asumiendo funciones de gestor de proyecto, analista, diseñador y programador.

Tabla 10. Recursos humanos

5.3.2. Costes recursos humanos

Para calcular los costes del proyecto se utilizarán los rangos salariales obtenidos de un estudio que ha realizado la empresa *Vitae Consultores* sobre los sueldos TIC en Galicia en los años 2013-2014.

El rango salarial para un Jefe de proyecto de 4 a 8 años de experiencia se sitúa entre 30.000 y 36.000 euros. Al Jefe de Proyecto de la Plataforma de Interoperabilidad se le asignarán 34.000 euros. Para calcular su coste se realizará calculando su coste por hora y multiplicándolo por el total de horas estimadas que dedicará al Módulo de Monitorización.

El rango salarial para un programador junior se sitúa entre 14.000 y 18.000 euros. Sergio Nogueiras Fernández, empleado asignado al proyecto Plataforma de Interoperabilidad tendrá un sueldo bruto anual de 16000 €. Para calcular su coste se calculará su coste por hora y se multiplicará por el total de horas dedicadas al Módulo de Monitorización.

José Ángel Taboada y Efrén Arias Jordan constarán como autónomos que prestan un servicio al Módulo de Monitorización y que tienen un coste de 17 euros/hora con IVA excluido.

Recurso	Salario bruto/mes	Salario neto/mes	Horas proyecto	Coste empresa
Sergio Nogueiras	16000	12416	465	5515
Jefe de proyecto	34000	25024	10	252.1

Tabla 11. Costes - Empleados

Para calcular el salario neto del Jefe de proyecto se aplica una retención del IRPF es del 20% y la aportación a la Seguridad Social del 6,4%. Para calcular el de Sergio Nogueiras se aplica una retención del IRPF del 16% y una aportación a la Seguridad Social de 6,4%.

El coste por hora que tiene cada recurso para la empresa se calcula sumándole a su salario bruto las aportaciones a la Seguridad Social por parte de la empresa dividido entre el total de horas anuales trabajadas. Para calcular el coste se multiplica el coste por hora por el total de horas dedicadas al proyecto.

A continuación se pueden ver las tablas con los datos utilizados para los cálculos:

Datos básicos	
Horas/día	8
Jornadas año	221
Horas/año	1768

Tabla 12. Costes - Datos básicos

Retenciones Seguridad Social en % 2014		
	Empresa	Trabajador
Desempleo	6,7	1,6
FOGASA	0,2	0
C.Comunes	23,6	4,7
Formación Profesional	0,6	0,1
Total aportación SS	0,311	0,064
Bases de cotización Contingencias Comunes 2014		
Base Máxima SS/mes (€) 2014	3597	
Base Mínima SS/mes (€) 2014	1051,5	

Tabla 13. Costes - Porcentajes Seguridad Social

Los costes de los demás recursos son:

Recurso	Coste/hora	Horas proyecto	IVA	Coste empresa
Efrén Arias Jordan	17	16	21%	329.12
José Ángel Taboada	17	16	21%	329.12

Tabla 14. Costes- Recursos subcontratados

5.3.3. Costes de material

El único coste de material atribuible al proyecto es el del portátil con el que se ha desarrollado el Módulo de Monitorización.

Material	Precio adquisición
Portátil Acer V3-571G i7-3630QM 2.40 GHz	800 euros

Tabla 15. Material proyecto

Los servidores utilizados tanto para desarrollo como para producción son aportados por el cliente, por lo que no representan un coste.

Dado que el portátil pertenece al inmovilizado de la empresa, el coste atribuible al proyecto es el de su amortización. Tomando como base de amortización el precio de adquisición del ordenador, 800 euros, suponiéndole una vida útil de cuatro años y un valor residual de 150 euros el coste de amortización anual es igual a:

$$(\text{Precio adquisición} - \text{Valor residual}) * \text{Coeficiente de amortización}$$

En este caso, suponiendo que la amortización es lineal, el coeficiente de amortización es del 25% de modo que la amortización anual será igual a:

$$(800-150)*0.25 = 162.5 \text{ euros}$$

Ponderándolo por el número de horas del proyecto con respecto a las horas anuales de uso del portátil. El coste de material para el proyecto asciende a:

$$162.5 *(465/1768) =42,73 \text{ euros}$$

5.3.4. Costes indirectos

Por último se deben calcular los costes indirectos. Los costes indirectos son aquellos costes que no pueden ser directamente repercutibles al proyecto. Ejemplos de estos costes son la luz, calefacción, agua...

El coste indirecto a imputar a los proyectos es establecido por la empresa. Este coste se establece por hora y en la empresa en la que se desarrolla el Módulo de Monitorización es de 6,5 euros.

La duración estimada del proyecto es de 465 horas, por lo que los costes indirectos supondrán un total de 3022,5 euros.

5.3.5. Coste total proyecto

Recurso	Coste total
Recursos humanos	6425.34 euros
Material	42.73 euros
Costes indirectos	3022.50 euros
Total	9490.57 euros

Tabla 16. Coste total proyecto

Como se puede ver en la anterior tabla el coste estimado del proyecto es de 9490.57 euros.

5.4. Plan de calidad

A continuación se definirá el Plan de Calidad del proyecto, el objetivo de este es definir una serie de procedimientos y reglas que ayuden a que el proyecto satisfaga las necesidades por las cuales se emprendió. Concretamente los procedimientos que formarán parte de la gestión de la calidad serán los procesos de Seguimiento y control del proyecto y el de Calidad de código.

Por último se mostrarán los resultados de calidad de código obtenidos al final del proyecto.

5.4.1. Seguimiento y control

Debido a que la duración del proyecto es amplia y a que el tiempo de rectificación es muy pequeño es necesario definir varias actividades de seguimiento en la planificación del proyecto, con el fin de obtener un feedback del estado del proyecto en tiempo para aplicar cambios.

Como ya se dijo en el apartado de *Gestión Temporal*, el proyecto sigue un ciclo de vida en cascada en el que podemos diferenciar claramente tres fases, *Análisis*, *Diseño* y *Desarrollo/Pruebas*. Utilizaremos esta división en fases para establecer las actividades de seguimiento y control al final de cada una. Además en medio de la fase de *Desarrollo/Pruebas* también se realizará un seguimiento, esto es así debido a que la duración de la fase de *Desarrollo/Pruebas* es muy amplia.

Cada actividad de seguimiento constará de las siguientes tareas:

- **Evaluación de cambios**

Se revisará el *Historial de cambios* para evaluar todas aquellas peticiones de cambio que aún no hayan sido evaluadas. En el caso de que existan peticiones de cambio aceptadas se deberán de realizar, para ello a la hora de planificar las tareas de seguimiento y control se dejará un pequeño margen de tiempo.

- **Seguimiento de riesgos**

Es importante identificar los riesgos del proyecto en una fase inicial del proyecto pero más importante es realizar un seguimiento del estado de estos e identificar nuevos. En las tareas de seguimiento de riesgos se evaluarán aquellos riesgos a los que se les esté haciendo un seguimiento con el fin de ver si alguno se produjo y aplicar el plan de corrección.

Además se eliminarán aquellos riesgos que ya no estén activos en el proyecto y se podrán identificar nuevos.

La identificación de los riesgos se basará en las listas de riesgos identificados en anteriores proyectos de la empresa y en el análisis de los elementos del proyecto y sus posibles amenazas.

El historial de riesgos se podrá consultar en el documento *[MM-HR]-HistorialRiesgos*.

- **Reunión con directores**

La última tarea de seguimiento será realizar una reunión con los directores del proyecto. El objetivo de estas reuniones será primero informar a los directores del estado del proyecto y segundo obtener su visión del proyecto así como las peticiones de cambio oportunas.

Para organizar las reuniones el proceso será el siguiente:

- Se enviará la documentación a los directores vía email a medida que se vaya generando, con el fin de dar un margen de tiempo para leerla antes de la reunión.
- Se compartirá un documento de texto online con los directores en el que se represente el *Acta de reunión*.

Las actas de reunión deberán tener los siguientes apartados:

- Fecha y lugar de la reunión.
- Nombre de los convocados.
- Objetivos de la reunión, debe de ser el índice que guíe la reunión.
- Conclusiones, permitirá tener un resumen de la reunión.

5.4.2. Calidad de código

5.4.2.1. Formato

El Módulo de monitorización se desarrolla dentro del entorno de desarrollo del Proyecto de Interoperabilidad, compartiendo recursos como son el repositorio de código y módulos comunes, esto provoca que para desarrollar ciertas funcionalidades haya que modificar módulos del Proyecto de Interoperabilidad.

La generación de código no puede ser descontrolada y debe de seguir unas reglas de formato, con el fin de facilitar la integración de los cambios con los demás desarrollos que se produzcan en la Plataforma de Interoperabilidad. De no seguir unas reglas de formateado, a la hora de integrar código nos encontraríamos con numerosos conflictos de código provocados por diferencias de formato.

Para que todos los desarrollos sigan un mismo formato se hará uso de las plantillas de formato que ofrece el entorno de desarrollo Eclipse. A continuación se muestra la configuración necesaria, dividida por lenguaje de programación. Para aquellos lenguajes que no se especifica la configuración deberá de definirse la configuración por defecto.

JAVA

En Windows->Preferences->Java->Code Style->Editor se importará la plantilla [MM]-CodeStyle-Formatter situada en el directorio *1.Módulo de monitorización/2.Gestión proyecto/5.Gestión de calidad*.

Con el fin de que el formateado de código sea una tarea automática se asociará la acción de *Guardar* con la de *Formatear*, de manera que cada vez que se guarde un archivo se le establezca el formato definido. En *Windows->Preferences->Java->Editor->Save Actions* definimos la siguiente configuración:

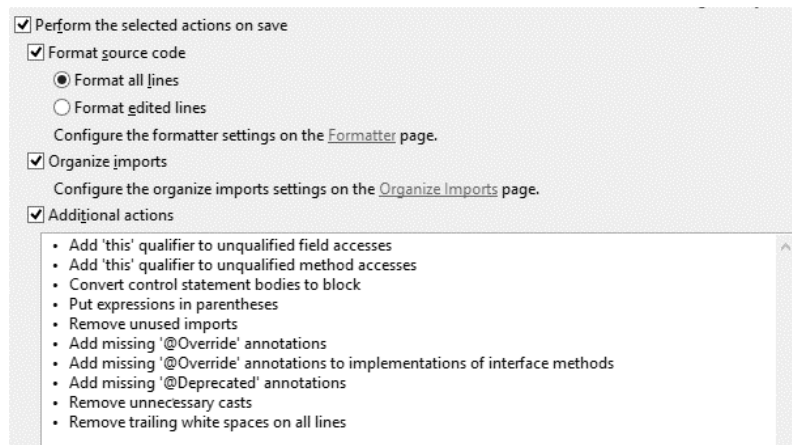


Ilustración 8. Configuración guardar-formatear.

HTML

En *Windows->Preferences->Web->HTML Files->Editor* establecemos la configuración como figura en la siguiente imagen:

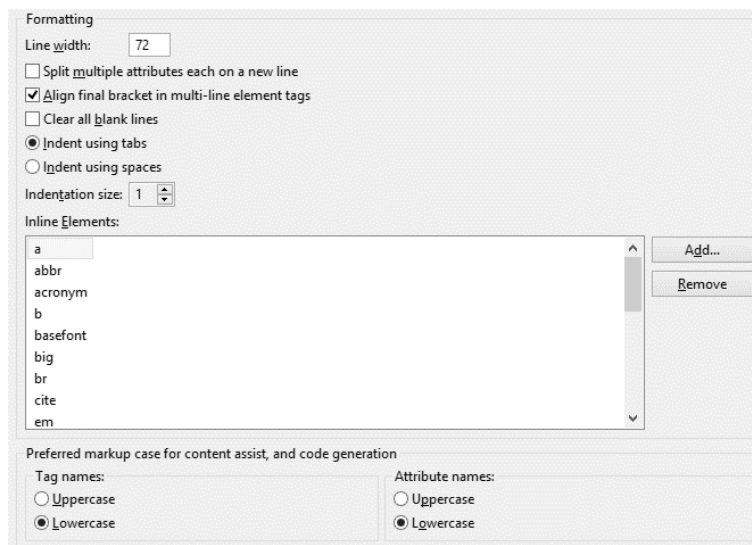


Ilustración 9. Configuración formato HTML

En el apartado de *Elementos en línea* debemos establecer solo las siguientes etiquetas:

a,abbr,acronym,b,basefont,big,br,cite,em,Font,i,img,q,s,select,small,strike,strong,sub,sup,title

5.4.2.2. Métricas y convenciones

Para revisar la calidad del código implementado en función de métricas y convenciones utilizaremos la herramienta de código libre *Sonar*. *Sonar* es una de las plataformas más populares para medir la calidad del código, permitiendo obtener métricas sobre diferentes lenguajes de programación. Su funcionamiento se basa en el uso de otras herramientas como *PMD* o *Checkstyle*.

En el entorno de desarrollo de la Plataforma de Interoperabilidad se cuenta con un servidor Sonar desplegado en la red interna del proyecto. Para integrar los proyectos del Módulo de Monitorización con este servidor se instalará el plugin de Sonar para Eclipse y se configurará la url del servidor.

Una vez configurado el plugin de Sonar podremos analizar cualquier proyecto utilizando el comando de Maven *sonar:sonar*, este comando subirá el código del proyecto al servidor Sonar. Con el proyecto importado al servidor Sonar podremos acceder a través de un navegador web y ver las diferentes métricas del código.

De toda la información que nos ofrece Sonar la que vamos a tener en cuenta van a ser la Complejidad ciclomática, Duplicación de código y las violaciones de buenas prácticas de programación.

Complejidad ciclomática

La complejidad ciclomática es una métrica que nos aporta medición cuantitativa de la complejidad lógica de un programa.

Sonar, calcula la complejidad ciclomática a través de un contador. Cuando el flujo de una función se altera, es decir, se produce un salto, este contador de complejidad aumenta en 1. Cada función tiene como mínimo una complejidad de 1.

En el caso de Java el contador aumenta cada vez que se encuentra con alguna de las siguientes sentencias:

if, for, while, case, catch, throw, return, &&, ||, ?, else. Cabe destacar que las llamadas a *getters* y *setters* no aumentan el contador.

Sonar aporta información de la complejidad total, de la complejidad de clase y de la complejidad de método. Para el Módulo de Monitorización se establecerá un máximo de 2.5 de media de complejidad en los métodos. El motivo de fijar la complejidad sobre los métodos y no sobre las clases o sobre el sistema en general es facilitar las posibles acciones de corrección, ya que, a través de Sonar podemos identificar aquellos métodos cuya complejidad es más crítica, por lo que tendríamos el problema mucho más acotado. Además se establece un valor de 2.5 con el fin de al menos igualar la complejidad del resto de módulos ya implementados de la Plataforma de Interoperabilidad.

Duplicación de código

Uno de los puntos que más perjudicial a la hora de que un código sea fácil de mantener son las duplicaciones de código. *Sonar* aporta información acerca de la cantidad de líneas duplicadas, se utilizará esta información para identificar código que pueda ser reutilizado y así evitar la duplicación.

En ciertas ocasiones la duplicación de código será inevitable por lo que se aceptará como máximo un 5% de código duplicado.

Convenciones de código

Sonar cuenta con una serie de reglas que permiten evaluar el cumplimiento de convenciones de programación Java, como pueden ser nomenclaturas, números mágicos, declaraciones, visibilidad de variables...

Se deberán cumplir las convenciones de programación Java en al menos un 90%, siguiendo el grado de cumplimiento de los anteriores módulos de la Plataforma de Interoperabilidad.

5.4.3. Resultados

A continuación se muestran los resultados obtenidos a la finalización del proyecto acerca de la calidad de código. La revisión de la calidad de código se hace por proyecto. El Módulo de Monitorización se divide en cuatro proyectos:

- Monitorización-Model: Contiene el código de los servicios del Módulo de Monitorización.
- Monitorización-Portlet: contiene el código de la Consola de monitorización.
- Análisis-Proactivo-Batch: contiene el código del batch que realiza el cálculo de los niveles de servicio y analiza su tendencia.
- Monitorización-Batch: contiene el código del batch que monitoriza los componentes de la Plataforma de Interoperabilidad.

Monitorizacion-Model	
Número de líneas de código	4188
Convenciones código Java	93.4%
Media complejidad ciclomática métodos	2.0
Duplicación código	0.9%

Tabla 17. Resultados calidad código Monitorizacion-Model

Monitorizacion-Portlet	
Número de líneas de código	3710
Convenciones código Java	91.3%
Media complejidad ciclomática métodos	1.5
Duplicación código	1%

Tabla 18. Resultados calidad código Monitorizacion-Portlet

Análisis-Proactivo-Batch	
Número de líneas de código	120
Convenciones código Java	97.5%
Media complejidad ciclomática métodos	2.3
Duplicación código	0 %

Tabla 19. Resultados calidad código Análisis-Proactivo-Batch

Monitorización--Batch	
Número de líneas de código	140
Convenciones código Java	97.9%
Media complejidad ciclomática métodos	2.1
Duplicación código	0%

Tabla 20. Resultados calidad código Monitorización-Batch

6. Análisis

Introducción

En este apartado de Análisis comenzaremos definiendo los parámetros concretos que se van a monitorizar y como se realizará el proceso de monitorización, para terminar reflejando la funcionalidad del sistema a través de los casos de uso y los requisitos.

6.1. Monitorización

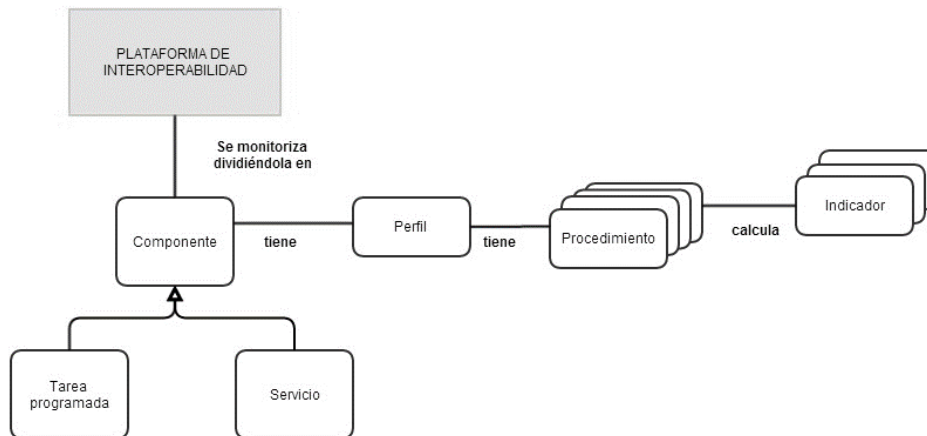


Ilustración 10. Estructura monitorización PLATINT

Para monitorizar la Plataforma de Interoperabilidad la dividiremos en componentes. Los componentes pueden ser o tareas programadas o servicios de la PLATINT que fueran dados de alta para la monitorización.

A cada uno de los componentes se le medirán unos indicadores con el fin de obtener información acerca de su estado. Para el cálculo de los indicadores se utilizarán diversos procedimientos que serán ejecutados automáticamente. Cada procedimiento puede calcular uno o varios indicadores. Los procedimientos son asignados a los componentes, pudiendo configurar la frecuencia de ejecución del procedimiento para cada componente concreto. Los valores de los indicadores calculados serán guardados como registros.

Un procedimiento será cualquier implementación cuya entrada sea un componente y cuya salida sean registros de los indicadores que calcula ese procedimiento, con el valor que tengan en el momento para el componente introducido. Los procedimientos se agruparán en Perfiles de monitorización, de este modo, se facilitará la asignación de procedimientos a los componentes. Como ya hemos visto, a cada componente se le puede configurar la frecuencia con la que se le ejecuta cada procedimiento, aun así, los procedimientos tendrán una frecuencia por defecto.

El valor de los indicadores para cada componente por sí solo no es suficiente para conocer el estado de un componente. Es necesario establecer una serie de umbrales para los indicadores de manera que se pueda conocer si el valor de un indicador está en un estado o en otro.

En la siguiente ilustración se puede ver gráficamente el sentido de los umbrales de los indicadores:

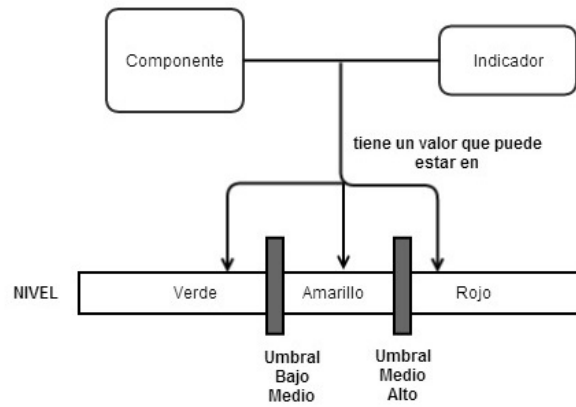


Ilustración 11. Umbrales de indicadores

Para obtener una información más sencilla de interpretar, se establecen tres niveles de monitorización para los valores de los indicadores:

- Rojo: El indicador está en un valor crítico.
- Amarillo: El indicador está en un valor de semi-alerta, puede ser crítico si se repite varias veces.
- Verde: El indicador está en un valor correcto.

Se utilizarán los siguientes umbrales para definir en qué valor se encuentra cada indicador:

- Umbral Bajo Medio: Establece el límite entre el nivel Rojo y el nivel Amarillo.
- Umbral Medio Alto: Establece el límite entre el nivel Amarillo y el Verde.

Si un indicador tiene el valor exacto de uno de los umbrales, su nivel de monitorización será el Rojo o el Verde, dependiendo de qué umbral sea.

Al igual que ocurre en los procedimientos con su frecuencia, los umbrales de los indicadores también tienen unos valores por defecto y también se pueden modificar en función de cada componente.

Uno de los objetivos del MM es avisar ante situaciones de alerta de la PLATINT. Para gestionar estos avisos se utilizarán reglas. Las reglas de avisos permiten establecer qué aviso hay que enviar y a quién en caso de que un indicador se encuentre cierto número de veces en un nivel de monitorización para un componente. Por ejemplo, se puede tener una regla para el servicio de Consulta de Identidad con los siguientes valores:

Indicador:*Tiempo petición***Nivel monitorización:***Amarillo***Máx. Repeticiones:***6***Mensaje:***El servicio Consulta de Identidad presenta un tiempo de petición demasiado alto.***Destinatarios:***Lista de email de administradores*

Con la anterior regla lo que ocurrirá es que, cuando se monitoriza el servicio de Consulta de Identidad, si los 6 últimos registros fueron amarillos, se enviará un aviso con el mensaje a los destinatarios establecidos. Además de enviar un aviso se creará una incidencia en el MM con el fin de tener un registro de las alertas que fueron generadas.

Las incidencias tendrán un estado que podrá ser *Resuelta* o *Sin Resolver*. Al crear una incidencia su estado siempre será *Sin Resolver*. Las incidencias se podrán resolver a través del portlet del MM.

6.1.1. Procedimientos e indicadores

A continuación se identifican los procedimientos e indicadores concretos que se van a utilizar en la primera versión del MM para vigilar el estado del sistema.

Procedimiento de Ping a Servicio

El procedimiento *Ping de Servicio* solo se aplicará en la monitorización de los servicios. Este procedimiento se basará en comprobar si los distintos webservices de los servicios están activos. El indicador calculado será:

- *Estado del Servicio*: podrá tomar los valores activo o no activo. En caso de que alguno de los webservices del servicio no esté activo, el indicador *Estado de Servicio* se considerará no activo.

Procedimiento Trazas de Servicio

El procedimiento de Trazas de Servicio será utilizado en la monitorización de los servicios. Analizando las peticiones que se realizan a los distintos servicios de la PLATINT podemos obtener distintos indicadores de los servicios.

Para extraer información acerca de las peticiones que se realizan a la PLATINT la fuente de información serán las trazas generadas por esta. Se deberán modificar las trazas que se generan

durante la petición a un servicio para que aporten la información necesaria para poder calcular los siguientes indicadores:

- *Estado petición*: toma los valores correcta o incorrecta en función de si la petición dio error o no. En caso de que una petición dé error este será el único indicador monitorizado para esa petición, ya que, al ser una petición errónea, el resto de indicadores carecen de valor.
- *Tiempo petición*: tiempo total que se tarda en tramitar una petición.
- *Tiempo Conector Ida*: tiempo que tarda el conector en validar y preparar la petición para enviarla al bus.
- *Tiempo Conector Vuelta*: tiempo que pasa la respuesta a la petición en el conector.
- *Tiempo Bus Ida*: tiempo que tarda el re-encaminamiento de la petición en el bus.
- *Tiempo Bus Vuelta*: tiempo que tarda la respuesta en el bus.
- *Tiempo Servicio Final*: tiempo que tarda la petición en ser respondida por el servicio final.

Procedimiento de Trazas de Tarea

El procedimiento de *Trazas de tarea* solo se aplicará para la monitorización de las Tareas programadas. Este procedimiento calculará el indicador:

- *Funcionamiento tarea*: podrá tomar los valores correcto o incorrecto en función de si se ejecuta correctamente la tarea o no.

Para cumplir con el procedimiento, las tareas programadas deberán generar trazas que informen acerca del estado de su ejecución.

6.1.2. Niveles de servicio

Los procedimientos e indicadores que acabamos de ver son los identificados para una primera versión del Módulo de Monitorización. No obstante, el Módulo de Monitorización se desarrollará de forma que se facilite la adición de más procedimientos e indicadores.

Además, para cada indicador se calculará un nivel de servicio. Los niveles de servicio son un resumen de los valores de los indicadores en un determinado tiempo.

El nivel de servicio para el indicador *Tiempo petición* ofrece por ejemplo, la velocidad media que tarda en ser resuelta una petición. Para calcularlo se mira el valor de los indicadores en la última semana (rango de tiempo por defecto) y se calcula la media del total de valores registrados.

6.2. Consola de monitorización

Con el fin de acercar la funcionalidad de la monitorización al usuario se dispondrá de una Consola de monitorización accesible para usuarios autorizados en la sección de administración del portal.

La Consola de monitorización deberá contar con una visión resumida del estado de los componentes. Se mostrará un resumen de los principales indicadores de todos los servicios como pueden ser el número de servicios activos, la tasa de errores general o el tiempo medio

de respuesta de los servicios. Además la consola facilitará el seguimiento de las últimas incidencias.

A través de la Consola de monitorización se podrán ver los servicios del Catálogo de Productos para activar o desactivar su monitorización. Al activar la monitorización de un servicio por primera vez se seleccionará el perfil que se le quiere asignar.

De cada servicio se podrán observar datos del servicio como nombre, estado servicio, estado monitorización, organismo responsable, perfil, incidencias y registros. A cada servicio se le podrán configurar los valores de la frecuencia con la que se ejecuta cada procedimiento, los umbrales de los indicadores, subir una plantilla para la petición de prueba y asignar o restablecer los valores de un perfil.

Como parte de la monitorización online, la Consola de monitorización permitirá realizar un ping a los servicios para conocer el estado de todos sus consumidores y productores. Si el servicio cuenta con datos de prueba y una plantilla de prueba configurada se podrá realizar una petición de prueba al servicio.

Otra de las funcionalidades que se deberá ofrecer a través de la Consola de monitorización es un buscador de incidencias y registros. A través de este buscador se podrán ver las incidencias y registros del sistema, hacer búsquedas por los campos más relevantes y exportar los datos a un fichero csv.

Por último, la Consola de monitorización deberá permitir la administración de la configuración de monitorización y la administración de las reglas. De la configuración de la monitorización se podrán modificar la fecha de inicio del análisis y los perfiles. Las reglas se podrán crear, modificar o eliminar. Para crear una regla será necesario indicar el componente, indicador, nivel de monitorización, máximo de repeticiones para que salte la regla, mensaje de aviso y la lista de destinatarios y/o lista de emails.

6.3. Cambios en la Plataforma de Interoperabilidad y dependencias

Durante el análisis realizado para el Módulo de Monitorización se han identificado ciertos cambios que se deben realizar en la actual Plataforma de Interoperabilidad. Estos cambios quedan fuera del proyecto del MM y serán llevados a cabo por el equipo de PLATINT.

Trazas bus

Es necesario que se cambien las trazas que genera el Bus de servicio para que ofrezcan la siguiente información:

- Id petición.
- Código de servicio.
- Número de solicitudes de la petición.
- Tiempo en el que la petición llega al conector.
- Tiempo en el que la petición sale del conector hacia el Bus.
- Tiempo en el que la petición llega al bus.
- Tiempo en el que el Bus llama al servicio final.

- Tiempo en el que el Bus recibe la respuesta del servicio final.
- Tiempo en el que el Bus devuelve la respuesta del servicio.
- Tiempo en el que la respuesta llega al conector.
- Tiempo en el que el conector devuelve la respuesta.
- Estado petición.

Trazas tareas programadas

Las trazas de las tareas programadas deberán de guardar información acerca de si la ejecución de la tarea se realizó correctamente o no.

Gestor de avisos

El equipo de PLATINT desarrollará un módulo de Gestión de avisos general para toda la plataforma, que permita el envío de emails y el almacenamiento de listas de emails. El MM utilizará este módulo para enviar avisos cuando salten las reglas.

6.4. Casos de uso

A través de los casos de uso que a continuación se presentan se mostrará la funcionalidad general del MM. Para cubrir la funcionalidad de los casos de uso se utilizarán los distintos requisitos, los cuales son una definición de las funcionalidades en un nivel de detalle mayor.

6.4.1. Actores

Los actores del Módulo de Monitorización son:

- **Monitorización:** representa cualquier proceso de tipo batch que se ejecuta cada cierto tiempo.
- **Usuario:** representa cualquier usuario con permisos para acceder al portlet del MM.

6.4.2. Escenarios

Para describir los diferentes casos de uso haremos una división de estos en escenarios. Por un lado tendremos el escenario de **Procesos Batch** y por el otro el escenario **Consola de monitorización**.

Dentro del escenario de **Procesos Batch** están todos aquellos casos de uso derivados de la ejecución automática de procesos que se encargarán de realizar la monitorización y el análisis proactivo.

El escenario **Consola de monitorización** representará todos los casos de uso relacionados con la interfaz web del Módulo de Monitorización.

6.4.3. Escenario 1: Procesos Batch

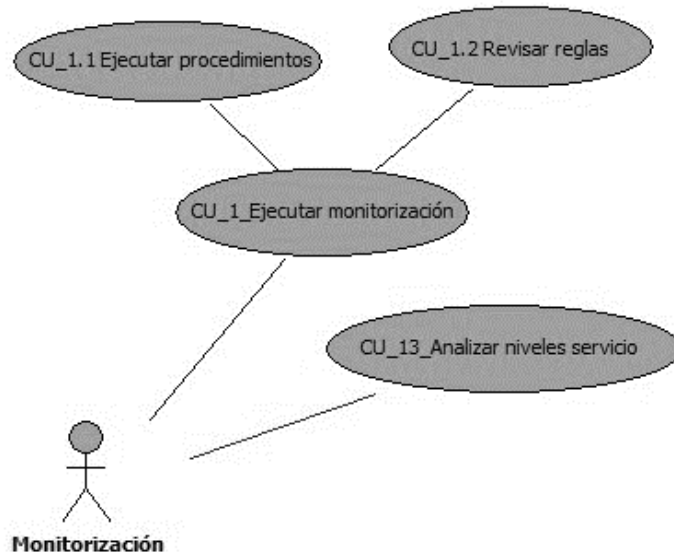


Ilustración 12. Diagrama casos de uso Procesos Batch

Nombre	CU_1 Ejecutar Monitorización
Descripción	La ejecución de la monitorización se basa en un proceso que se ejecuta cada cierto tiempo y que lanza la ejecución de los procedimientos y la revisión de las reglas.
Flujo	<ol style="list-style-type: none"> 1. Se obtienen todos los componentes activos. 2. Se ejecutan los procedimientos (CU 1.1). 3. Se ejecuta la revisión de las reglas (CU 1.2). <p>*Este flujo se ejecuta periódicamente.</p>
Escenarios alternativo	N/A
Reglas negocio	N/A
Condiciones previas	N/A
Condiciones posteriores	Se ejecuta la monitorización de los componentes a través de la ejecución de los procedimientos y la revisión de las reglas.
Actores	Monitorización

Nombre	CU1.1 Ejecutar procedimientos
Descripción	Cada componente monitorizado tiene asignados una serie de procedimientos a través de los cuáles se obtienen los valores de sus indicadores para guardarlos como registros. En el proceso de monitorización se obtienen todos los componentes y se ejecutan sus procedimientos.
Flujo	<ol style="list-style-type: none"> 1. Se obtienen todos los componentes activos. 2. Por cada componente se recuperan sus procedimientos. 3. Se ejecuta cada procedimiento (EA1). 4. Se guardan los registros generados. 5. Se obtienen las reglas a partir del componente y del indicador. 6. Por cada regla se verifican los registros (EA1).
Escenarios alternativo	<p>EA1. Frecuencia procedimiento-componente</p> <p>La frecuencia con la que se ejecuta cada procedimiento para cada componente es configurable, por ello, antes de ejecutar un procedimiento se comprueba si el tiempo que ha pasado desde la última ejecución del procedimiento para el componente es mayor que la frecuencia. Si se cumple lo anterior se ejecuta el procedimiento, sino se pasa al siguiente procedimiento.</p>
Reglas negocio	N/A
Condiciones previas	Debe existir al menos un componente activo en la monitorización
Condiciones posteriores	Se crean los registros con los valores de los indicadores para cada componente
Actores	Monitorización

Nombre	CU1.2 Revisar reglas
Descripción	La revisión de reglas es un procedimiento que comprueba el valor de los registros monitorizados con las reglas de los indicadores. Este procedimiento permite detectar situaciones de error y enviar avisos a los responsables.
Flujo	<ol style="list-style-type: none"> 1. Se reciben los registros a comprobar. 2. Se obtienen las reglas a partir del componente y del indicador. 3. Por cada regla se verifican los registros (EA1).

Escenarios alternativo	EA1. Se cumple la regla En caso de cumplirse la regla se comprueba si ya existe una incidencia que haga referencia a esa regla. Si existe ya una incidencia no se realiza ninguna acción. En caso de que no exista ninguna incidencia se envía un aviso y se crea una incidencia.
Reglas negocio	RN1. Perfil componente servicio Un componente de tipo servicio siempre tiene que tener un perfil de monitorización asignado.
Condiciones previas	Deben existir componentes activos en monitorización.
Condiciones posteriores	Se validan los registros y en caso de que se encuentra algún error se envía un aviso y crea una incidencia.
Actores	Monitorización

Nombre	CU13 Analizar niveles de servicio
Descripción	Cada intervalo de tiempo marcado por el rango de días de los niveles de servicio se procede al cálculo de estos. Los niveles de servicio se calcularán a partir de los registros del MM, se registrarán y se analizarán respecto a los anteriores niveles de servicio para detectar posibles degradaciones de servicio.
Flujo	<ol style="list-style-type: none"> 1. Se obtiene la fecha de inicio de análisis configurada en el MM (EA1). 2. Se recuperan todos los componentes activos y se calculan sus niveles de servicio. 3. Se registran los niveles de servicio. 4. Se analizan los niveles de servicio registrados en función de los anteriores (EA2). 5. Se muestran los detalles de la incidencia, donde se mostrará la incidencia como resuelta.
Escenarios alternativo	EA1. Fecha actual posterior a fecha de inicio del análisis. Si la fecha establecida en la configuración del MM para el inicio del análisis de los niveles de servicio es posterior a la fecha actual se termina el análisis de los niveles de servicio. EA2. Se detecta una degradación del nivel de servicio.

	Si se detecta una degradación en el nivel de servicio se enviará un aviso a los responsables.
Reglas negocio	N/A
Condiciones previas	Deben existir componente para monitorizar, registros y niveles de servicio.
Condiciones posteriores	Se registran los niveles de servicio y se analiza si existe una degradación de servicio, en caso de existir se envía un aviso.
Actores	Monitorización

Tabla 21. Casos de uso escenario Procesos Batch

6.4.4. Escenario 2: Consola de monitorización

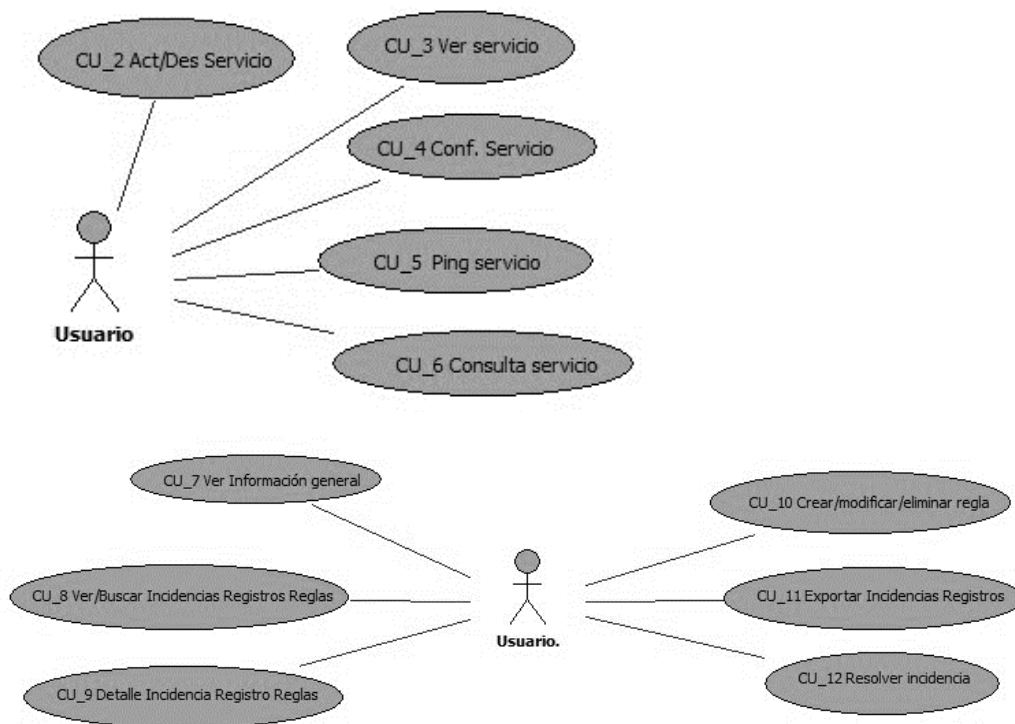


Ilustración 13. Diagrama casos de uso Consola monitorización

Nombre	CU2 Activar / Desactivar Servicio
Descripción	Los servicios de la Plataforma de Interoperabilidad podrán ser monitorizados o no. Para ello cada servicio podrá ser activado o desactivado a través del portlet del MM.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de búsqueda de servicios. 2. El usuario pulsa el botón de “Activar/Desactivar servicio”. 3. El servicio cambia de estado (EA1).
Escenarios alternativo	<p>EA1. Activar perfil por primera vez</p> <p>La primera vez que se activa un servicio se solicitará seleccionar un perfil de monitorización y se le asignarán los procedimientos e indicadores de este. En caso de que ya estuviese activado se le asigna la configuración que ya tenía.</p>
Reglas negocio	<p>RN1. Perfil componente servicio</p> <p>Un componente de tipo servicio siempre tiene que tener un perfil de monitorización asignado.</p>
Condiciones previas	Deben existir servicios en la Plataforma de Interoperabilidad
Condiciones posteriores	El servicio cambia su estado de monitorización
Actores	Usuario

Nombre	CU3 Ver servicio
Descripción	Desde el portlet del MM se podrá ver información de los distintos servicios monitorizados. Entre la información del servicio estará su nombre, su responsable, su estado de monitorización, sus incidencias y sus últimos registros.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de búsqueda de servicios. 2. El usuario pulsa el botón de “Ver detalle servicio”. 3. Se muestra la información del servicio.
Escenarios alternativo	N/A

Reglas negocio	N/A
Condiciones previas	Debe de existir al menos un servicio activo en la monitorización
Condiciones posteriores	Se muestra el detalle de un servicio
Actores	Usuario

Nombre	CU4 Configurar servicio
Descripción	Los componentes de tipo servicio podrán ser configurados. Los parámetros que se podrán configurar son un perfil de monitorización, una plantilla de pruebas, la frecuencia de ejecución de los procedimientos y los umbrales de los indicadores.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de búsqueda de servicios. 2. El usuario pulsa el botón de “Configurar un servicio”. 3. Se muestra la configuración del servicio con los datos actuales. 4. El usuario modifica los datos de configuración (EA1). 5. Se validan los datos introducidos (EA2). 6. Se guarda la configuración del servicio y se muestra la pantalla de detalle de un servicio.
Escenarios alternativo	<p>EA1. Establecer valores defecto perfil</p> <p>Los procedimientos e indicadores que aparecen en cada servicio para ser configurados vienen dados por el perfil que tiene asignado. Si se quiere establecer otros procedimientos e indicadores o establecer los valores por defecto de estos, se deberá seleccionar un perfil y pulsar el botón “Establecer perfil”.</p> <p>EA2. Validación</p> <p>Si los datos introducidos no pasan la validación se volverá a la página de configuración del servicio y se mostrará un mensaje de error.</p>
Reglas negocio	<p>RN1. Validación</p> <p>En la configuración de un servicio todas las frecuencias de los procedimientos tienen que ser positivas y los umbrales de los indicadores deben cumplir que el umbral medio alto sea mayor que el medio bajo.</p> <p>RN2. Plantilla</p>

	Si se configura una plantilla de prueba para utilizar en la consulta al servicio, será responsabilidad del usuario subir una plantilla correcta.
Condiciones previas	Debe de existir un componente de tipo servicio activado
Condiciones posteriores	El servicio queda configurado
Actores	Usuario

Nombre	CU5 Ping servicio
Descripción	Para conocer el estado de los conectores productores y consumidores de servicios, se podrá hacer un ping a estos a través del portlet del MM.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la vista de detalle de un servicio. 2. El usuario pulsa el botón de “Ping de servicio”. 3. Se muestra una pantalla en la que se pueden ver por un lado los consumidores con su estado y por otro los productores también con su estado.
Escenarios alternativo	N/A
Reglas negocio	N/A
Condiciones previas	Debe existir al menos un servicio activado en monitorización.
Condiciones posteriores	Se muestra el estado de los distintos conectores.
Actores	Usuario

Nombre	CU6 Consulta servicio
Descripción	El portlet del MM permite realizar una consulta de prueba a los servicios de la Plataforma de Interoperabilidad. La consulta permitirá al usuario comprobar si el servicio está funcionando correctamente.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la vista de detalle de un servicio. 2. El usuario pulsa el botón de “Consulta de prueba”. 3. Se descarga la respuesta del servicio en un fichero xml.

Escenarios alternativo	N/A
Reglas negocio	RN 1. Plantilla de consulta Los datos a consultar del servicio se introducen a través de una plantilla en la configuración del servicio. Si un servicio no tiene configurada una plantilla no se podrá realizar la consulta de prueba.
Condiciones previas	Existe un servicio monitorizado que tenga datos de prueba y una plantilla de prueba configurada.
Condiciones posteriores	Se descarga un fichero xml con la respuesta del servicio.
Actores	Usuario

Nombre	CU7 Ver información general
Descripción	Como página de inicio del portlet del MM se mostrará una pantalla con información general del Módulo de Monitorización.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la página inicial del portlet del MM. 2. Se muestra información general del MM como son: <ol style="list-style-type: none"> a. Últimas incidencias b. Resumen de los registros de los servicio monitorizados.
Escenarios alternativo	N/A
Reglas negocio	N/A
Condiciones previas	El portlet del Módulo de Monitorización está disponible en el Portal de Interoperabilidad y el usuario tiene permisos para acceder.
Condiciones posteriores	Se muestra la información general del MM.
Actores	Usuario

Nombre	CU8 Ver/Buscar Incidencias, Reglas, Registros
Descripción	Existirán diversas zonas en el portlet del MM a través de las cuales el usuario dispondrá de buscadores de incidencias, registros y reglas.

Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la zona de búsqueda de incidencias, reglas o registros. Se podrá acceder desde diversos sitios: <ol style="list-style-type: none"> a. Incidencias: Desde la pantalla general donde se muestran las últimas incidencias, desde el detalle de un servicio donde se muestran las incidencias del servicio o desde el menú de navegación. b. Reglas. Desde el menú de navegación. c. Registros: Desde el detalle de una incidencia donde se muestran los registros relacionados con la incidencia, desde el detalle de un servicio donde se muestran los últimos registros del servicio o desde el menú de navegación. 2. Se muestran todas las incidencias, reglas o registros ordenados por fecha y paginados. 3. Se podrán filtrar las incidencias, reglas o registros utilizando el panel de búsqueda donde se podrán introducir diversos campos de búsqueda para incidencias, reglas o registros (EA1). 4. Se muestran las incidencias, reglas o registros buscados.
Escenarios alternativo	<p>EA1. Búsqueda sin resultado.</p> <p>Si no se encuentran resultados que coincidan con los campos de búsqueda se muestra un mensaje informativo en el lugar de los resultados.</p>
Reglas negocio	N/A
Condiciones previas	N/A
Condiciones posteriores	Se muestran las incidencias, registros o reglas si existen, en otro caso, se muestra un mensaje informativo.
Actores	Usuario

Nombre	CU9 Detalle Incidencia, Regla
Descripción	A través del portlet del Módulo de Monitorización se podrán ver los datos de una incidencia o de una regla.
Flujo	<ol style="list-style-type: none"> 5. El usuario accede a la zona de búsqueda de incidencias o reglas. 6. El usuario pulsa el botón de “Detalles” de una regla o incidencia. 7. Se muestran los datos de la incidencia o de la regla seleccionada.

Escenarios alternativo	N/A
Reglas negocio	N/A
Condiciones previas	Debe existir al menos una regla o una incidencia.
Condiciones posteriores	Se muestran los datos de la incidencia o regla.
Actores	Usuario

Nombre	CU10.1 Crear regla
Descripción	El portlet del MM permitirá la creación de reglas. Las reglas tendrán un componente, un indicador, un número de repeticiones de un nivel de monitorización, un mensaje y unos destinatarios.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la zona de reglas y pulsa el botón de “Crear”. 2. Se muestran los campos de la regla. 3. El usuario rellena los campos de la regla y pulsa en guardar. 4. Se validan los campos de la regla (EA1). 5. Se guarda la regla. 6. Se muestra la vista de detalle de la regla creada.
Escenarios alternativo	<p>EA1. Validación de los campos incorrecta.</p> <p>Si los campos introducidos no pasan la validación se muestra la página de creación de la regla con los datos rellenos y los mensajes de error.</p>
Reglas negocio	<p>RN1. Validación</p> <p>Para crear una regla es obligatorio introducir:</p> <ul style="list-style-type: none"> • Componente • Indicador • Nivel de monitorización • Máx. repeticiones • Mensaje • Al menos un destinatario o una lista de emails.
Condiciones previas	Deben existir componentes e indicadores en el Módulo de Monitorización

Condiciones posteriores	Se crea la regla
Actores	Usuario

Nombre	CU10.2 Modificar regla
Descripción	El portlet del MM permitirá la modificación de reglas.
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la zona de reglas y pulsa el botón de “Modificar” de una regla (EA1). 2. Se muestran los campos de la regla. 3. El usuario rellena los campos de la regla y pulsa en guardar. 4. Se validan los campos de la regla (EA2). 5. Se guarda la regla. 6. Se muestra la vista de detalle de la regla creada.
Escenarios alternativo	<p>EA1. Modificar desde la vista de detalle.</p> <p>El usuario puede acceder a la vista de detalle de la regla, pulsando el botón de “Ver detalle”, para luego desde esta pulsar el botón de modificar e ir a la vista de modificar.</p> <p>EA2. Validación de los campos incorrecta.</p> <p>Si los campos introducidos no pasan la validación se muestra la página de modificación de la regla con los datos rellenos y los mensajes de error.</p>
Reglas negocio	<p>RN1. Validación</p> <p>Para crear una regla es obligatorio introducir:</p> <ul style="list-style-type: none"> • Componente • Indicador • Nivel de monitorización • Máx. repeticiones • Mensaje • Al menos un destinatario o una lista de emails. <p>RN2. Eliminar referencias a la regla</p> <p>Al modificar una regla se asume que todas las incidencias que fueron creadas por esta pierden validez por lo que se borra la referencia a la regla por parte de la incidencia y se resuelve la incidencia.</p>

Condiciones previas	Deben existir al menos una regla
Condiciones posteriores	Se modifica la regla
Actores	Usuario

Nombre	CU10.3 Eliminar regla
Descripción	El portlet del MM permitirá la eliminación de reglas
Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la zona de reglas. 2. Busca la regla que desea eliminar. 3. Pulsa el botón de “Eliminar regla” (EA1). 4. Se elimina la regla 5. Se guarda la regla. 6. Se muestra la vista de detalle de la regla creada.
Escenarios alternativo	<p>EA1. Eliminar regla desde detalle</p> <p>El usuario puede entrar a ver el detalle de la regla a través del botón de “Ver detalles” y eliminarla desde la vista de detalles.</p>
Reglas negocio	<p>RN1. Eliminar referencias a la regla</p> <p>Al eliminar una regla se asume que todas las incidencias que fueron creadas por esta pierden validez por lo que se borra la referencia a la regla por parte de la incidencia y se resuelve la incidencia.</p>
Condiciones previas	Deben existir al menos una regla
Condiciones posteriores	Se elimina la regla
Actores	Usuario

Nombre	CU11 Exportar Registros, Incidencias
Descripción	Un usuario podrá descargar en formato csv tanto los registros como las incidencias guardadas por el Módulo de Monitorización. Se podrán realizar búsquedas para seleccionar las incidencias o registros que se quieran descargar.

Flujo	<ol style="list-style-type: none"> 1. El usuario accede a la zona de búsqueda de incidencias o registros (EA1). 2. El usuario pulsa el botón de “Exportar”. 3. Se descargan las incidencias o registros.
Escenarios alternativo	<p>EA1. Búsqueda de registros o incidencias.</p> <ol style="list-style-type: none"> 4. El usuario realiza una búsqueda para filtrar las incidencias o registros que desea descargar.
Reglas negocio	N/A
Condiciones previas	Deben existir incidencias o registros
Condiciones posteriores	Se descarga un archivo en formato csv con las incidencias o registros.
Actores	Usuario

Nombre	CU12 Resolver incidencia
Descripción	Cuando una incidencia se considere solucionada se podrá marcar como resuelta a través del portlet del MM.
Flujo	<ol style="list-style-type: none"> 1. Un usuario accede a la pantalla de búsqueda de incidencias. 2. El usuario pulsa el botón de “Resolver” de una incidencia no resuelta (EA1). 3. Se muestran los detalles de la incidencia, donde se mostrará la incidencia como resuelta.
Escenarios alternativo	<p>EA1. Resolver incidencia desde detalle.</p> <ol style="list-style-type: none"> 1. El usuario pulsa el botón de “Ver detalles” de una incidencia no resuelta. 2. Se muestran los detalles de la incidencia. 3. El usuario pulsa el botón de “Resolver” incidencia.
Reglas negocio	N/A
Condiciones previas	Debe existir al menos una incidencia en estado no resuelta.
Condiciones posteriores	La incidencia tiene el estado resuelto.

Actores	Usuario
----------------	---------

Nombre	CU14 Configurar monitorización
Descripción	<p>El portlet del MM contará con una zona en la que se podrán configurar los siguientes parámetros:</p> <ul style="list-style-type: none"> • Perfiles de monitorización: Se podrán definir perfiles de monitorización asignándoles un nombre y una lista de procedimientos. • Rango de análisis: Se podrá indicar la fecha a partir de la cual se tendrán en cuenta los datos monitorizados.
Flujo	<ol style="list-style-type: none"> 1. Un usuario accede a la pantalla de configuración de la consola de administración. 2. Introduce los datos de configuración y los guarda. 3. El sistema valida los datos introducidos. <ol style="list-style-type: none"> a. En caso de error en la configuración se indica (EA1). 4. Se registra la nueva configuración de monitorización. 5. Se muestra la nueva configuración.
Escenarios alternativo	<p>EA1. Los datos introducidos no son válidos.</p> <p>En el caso de que los datos enviados no sean válidos se mostrará un mensaje de error que impedirá llevar a cabo la modificación solicitada (RN1).</p>
Reglas negocio	<p>RN1. Validación</p> <p>Perfil de monitorización: Al crear un perfil se validará que no existe un perfil igual y que como mínimo el perfil tenga un procedimiento asignado. Si se elimina un perfil se comprobará que no esté asignado a ningún componente, en caso de estarlo no se podrá eliminar.</p> <p>Rango de análisis: La fecha a partir de la cual se analizarán los datos de monitorización será posterior a la fecha actual. Además el formato de fecha debe ser DD/MM/AAAA.</p>
Condiciones previas	N/A
Condiciones posteriores	La configuración de monitorización queda registrada.
Actores	Usuario

Tabla 22. Casos de uso Consola de monitorización

6.5. Matriz de trazabilidad

La matriz que a continuación se muestra relaciona los casos de uso con los requisitos funcionales relacionados con cada uno.

		CU_1	CU_1.1	CU_1.2	CU_2	CU_3	CU_4	CU_5	CU_6	CU_7	CU_8	CU_9	CU_10	CU_11	CU_12	CU_13	CU_14
		Ejecutar monitorización	Ejecutar procedimientos	Revisar reglas	Act/Desc servicio	Ver servicio	Conf. servicio	Ping servicio	Consulta servicio	Ver inf. General	Ver/Buscar Incidencias Registros Reglas	Detalle Incidencia Regla	Crear/modificar/eliminar regla	Exportar Incidencias Registros	Resolver Incidencia	Analizar niveles servicio	Configurar monitorización
RF_1	Componentes		x														
RF_2	Procedimientos		x	x												x	
RF_3	Indicadores		x					x									
RF_4	Ping servicio		x														
RF_5	Trazas tareas		x														
RF_6	Trazas servicios		x														
RF_7	Registros															x	
RF_8	Niveles de servicio	x															
RF_9	Batch monitorización	x		x												x	
RF_10	Enviar avisos	x		x													
RF_11	Crear incidencias			x		x					x					x	
RF_12	Ver/Buscar incidencias											x			x		
RF_13	Detalles incidencia														x		
RF_14	Resolver incidencias			x									x				

		CU_1	CU_1.1	CU_1.2	CU_2	CU_3	CU_4	CU_5	CU_6	CU_7	CU_8	CU_9	CU_10	CU_11	CU_12	CU_13	CU_14
		Ejecutar monitorización	Ejecutar procedimientos	Revisar reglas	Act/Desc servicio	Ver servicio	Conf. servicio	Ping servicio	Consulta servicio	Ver inf. General	Ver/Buscar Incidencias Registros Reglas	Detalle Incidencia Regla	Crear/modificar/eliminar regla	Exportar Incidencias Registros	Resolver Incidencia	Analizar niveles servicio	Configurar monitorización
RF_15	Crear/editar/eliminar reglas										x						
RF_16	Ver/Buscar reglas					x					x			x			
RF_17	Ver/Buscar registros				x												
RF_18	Activar/Desactivar servicio				x												
RF_19	Ver servicios					x											
RF_20	Detalles servicio						x		x								
RF_21	Configurar servicio																x
RF_22	Perfiles de monitorización				x		x										
RF_23	Asignar perfil a servicio				x		x		x								
RF_24	Consulta servicio								x								x
RF_25	Editar conf. Monitorización															x	x

Tabla 23. Matriz trazabilidad Casos de uso / Requisito

6.6. Requisitos funcionales

RF1: Componentes

Para realizar la monitorización de la Plataforma de Interoperabilidad, esta se dividirá en Componentes. Tanto el estado de la plataforma como los avisos e incidencias se obtendrá a partir del estado de los componentes.

Existirán dos tipos de componentes, las tareas programadas y los servicios. La monitorización de las tareas programadas no será configurable a través del portlet de monitorización mientras que la de los servicios sí.

Los componentes de tipo servicio podrán estar activos, de manera que se monitoricen o desactivados.

RF2: Procedimientos

El Módulo de Monitorización contará con una serie de procedimientos que calcularán unos indicadores para cada componente que lo tenga asignado.

Los procedimientos para la primera versión del Módulo de monitorización serán:

- Ping servicio
- Trazas servicio
- Trazas tarea

Cada procedimiento tendrá una frecuencia definida que marcará cada cuánto tiempo se ejecuta un procedimiento para calcular los indicadores de un componente.

Además de calcular el valor de los indicadores, los procedimientos deben guardar el valor de estos como Registros de monitorización.

RF3: Indicadores

Los indicadores son parámetros de los componentes que pueden ser medidos para obtener información del estado de estos. Cada indicador será calculado por un procedimiento y tendrá dos umbrales, uno será el bajo-medio y otro el medio-alto.

Los umbrales definirán en qué estado se encuentra un componente para un indicador. Estos posibles estados en los que puede estar un indicador son rojo, amarillo y verde.

RF4: Ping servicio

Uno de los procedimientos para monitorizar los servicios será realizar un ping a todos los webservices tanto productores como consumidores de este. Este procedimiento calcula el indicador Estado de servicio, el cual podrá tomar los valores activo o no activo. Para tomar el valor activo todos los webservices han de estar activos.

Además se podrá invocar a este procedimientos desde el portlet del MM, de manera que se pueda ver es estado de los webservices online. Cuando se llame al procedimiento de Ping de servicio de manera online no se crearán registros.

RF5: Trazas tareas

El procedimiento de trazas será utilizado para monitorizar las tareas programadas y consistirá en leer las trazas generadas para obtener el indicador de Estado de traza, indicador que ofrece información acerca de si la tarea programada está funcionando correctamente o no.

RF6: Trazas servicios

El procedimiento de trazas de servicio será utilizado para obtener los indicadores de las peticiones a los servicios. Estos indicadores serán los distintos tiempos que una petición pasa en el conector, en el bus y en el servicio final.

RF7: Registros

El Módulo de Monitorización contará con un registro en el que se almacenen los valores de los indicadores en todas sus mediciones. El registro no se podrá borrar.

RF8: Niveles de servicio

Los niveles de servicio aportan información sobre cuál ha sido el estado de los indicadores en un rango de tiempo.

Se realizará un análisis de los niveles de servicio que permita avisar a los responsables en caso de que la tendencia de los niveles de servicios sea negativa. Este análisis permitirá detectar posibles fallos futuros.

RF9: Batch monitorización

El Módulo de monitorización contará con un proceso que se lanzará automáticamente para ejecutar los distintos procedimientos y calcular los indicadores de monitorización para los componentes activos. Además de ejecutar los procedimientos se comprobarán los valores de los indicadores con el fin de avisar a los responsables en caso de que se detecten estados incorrectos.

RF10: Enviar avisos

El Módulo de Monitorización debe enviar emails cuando se detecte que un componente se encuentra en un estado incorrecto. En el contenido del email debe estar la información necesaria para que el destinatario pueda identificar el error.

RF11: Crear incidencias

El Módulo de Monitorización registrará los estados incorrectos encontrados creando incidencias. Las incidencias permitirán obtener a través del portlet del MM información de todos los problemas detectados. Todas las incidencias se crearán con estado Sin resolver.

RF12: Ver/Buscar incidencias

Se dispondrá de una zona en el portlet del Módulo de Monitorización donde realizar búsquedas de las incidencias producidas. Se podrán buscar incidencias, resolverlas y ver sus detalles. Las incidencias se podrán buscar por componente, estado y fecha.

RF13: Detalles incidencia

El Módulo de Monitorización permitirá obtener información detallada de una incidencia. Esta información constará al menos de componente, indicador, estado, registros relacionados y mensaje.

RF14: Resolver incidencias

Una incidencia puede tener estado Resuelta o Sin resolver. Aquellas incidencias no resueltas se podrán resolver a través del portlet del Módulo de Monitorización.

RF15: Crear/editar/eliminar reglas

Para enviar avisos y crear incidencias se utilizarán una serie de reglas que permitan contrastar si el valor de un indicador debe de enviar un aviso. Las reglas harán referencia a un componente, un indicador, un estado, un número de repeticiones y un mensaje. De este modo si un componente para determinado indicador se encuentra un número de veces en el estado que marca la regla se enviará un aviso con el mensaje de la regla.

Las reglas podrán crearse, editarse y eliminarse para los componentes de tipo servicio.

RF16: Ver/Buscar reglas

El Módulo de Monitorización contará con una zona de búsqueda de reglas que permitirá un acceso más rápido a estas para editarlas/eliminarlas. Se podrán realizar búsquedas por componente e indicador.

RF17: Ver/Buscar registros

El MM contará con una zona de búsqueda de registros en la que se podrán filtrar estos por id, componente, indicador, nivel de monitorización y fecha.

RF18: Activar/Desactivar servicio

Desde la Consola de monitorización se podrá activar y desactivar la monitorización de los componentes de tipo servicio. Si es la primera vez que se activa un servicio se deberá de asignar un perfil de monitorización.

RF19: Ver servicios

Se dispondrá de una zona en el portlet del MM donde se podrán ver los distintos servicios de la Plataforma de Interoperabilidad. A través de esta pantalla se podrá acceder a activar/desactivar la monitorización de un servicio, ver sus detalles o configurarlo.

RF20: Detalles servicio

En el detalle de un servicio se mostrará información como su nombre, su responsable, su estado de monitorización, sus incidencias y últimos registros.

Desde el detalle de un servicio se podrá hacer ping a este o lanzar una consulta de prueba en caso de que tenga una plantilla de prueba configurada.

RF21: Configurar servicio

En la configuración de un servicio se podrá establecer un perfil de monitorización, modificar la frecuencia con la que se ejecutan los procedimientos que el servicio tiene asignados y modificar los umbrales de los indicadores del servicio.

RF22: Perfiles de monitorización

El Módulo de Monitorización contará con una serie de perfiles de monitorización. Un perfil es un conjunto de procedimientos.

RF23: Asignar perfil a servicio

A través de la asignación de un perfil a un servicio se le establecerán a este todos los procedimientos e indicadores que tenga el perfil con sus valores por defecto. Al activar un servicio por primera vez será obligatorio asignar un perfil.

RF24: Consulta servicio

Para los servicios de la Plataforma de Interoperabilidad que tengan datos de prueba en el entorno de producción se podrá realizar una consulta de prueba que sirva para ver la respuesta en formato xml del servicio.

RF25: Editar conf. Monitorización

En la configuración general de la monitorización se podrá establecer una fecha a partir de la cual se iniciará la monitorización y además se podrán administrar los diferentes perfiles de monitorización.

6.7. Requisitos de diseño

RD1: Estructura modular

Siguiendo la línea del resto de desarrollos de la Plataforma de Interoperabilidad el diseño del módulo de monitorización tendrá que ser modular.

RD2: Integración módulo trazas

El módulo de monitorización se integrará con el módulo de trazas de la actual Plataforma de Interoperabilidad para obtener información sobre los indicadores del sistema.

RD3: Escalable

El proceso de monitorización será independiente del número de componentes y de los procedimientos, de manera que se puedan integrar fácilmente nuevos procedimientos y componentes.

6.8. Requisitos de seguridad

RS1: Conservar registros

Los registros de monitorización no podrán ser modificados ni eliminados.

RS2: Control de acceso

Tanto el acceso como la información que se podrá consultar a través de la Consola de administración serán controlados en función de los permisos del usuario.

7. Diseño

Introducción

Los primeros conceptos sobre los que tratará este capítulo son los referentes a la arquitectura del proyecto. Es conveniente comenzar definiendo la arquitectura con el objetivo de poder ver cuáles son las capas en las que se divide el Módulo de Monitorización y como estas interactúan entre sí.

En el resto del capítulo se definirá el modelo de datos y se detallará el diseño y construcción de cada una de las distintas capas de la arquitectura, así como, los patrones y frameworks utilizados.

Para dar más valor a los diseños, se introducirá en el medio de estos, secciones de diseño detallado, que permitirán clarificar los diseños acercándose un poco más a la implementación. Estas secciones estarán diferenciadas por un recuadro gris, para que aquellos lectores que no quieran entrar al detalle puedan identificarlas fácilmente.

7.1. Arquitectura

La arquitectura que se va a utilizar para el desarrollo del Módulo de Monitorización es una arquitectura por capas.

La arquitectura por capas más utilizada en el desarrollo de sistemas informáticos es la *Arquitectura de Tres Capas*, que se divide en:

- **Capa de Presentación:** responsable de la interacción con el usuario final.
- **Capa de Lógica de negocio:** responsable de cumplir con las reglas de negocio.
- **Capa Almacenamiento:** responsable de las operaciones de persistencia del sistema.

Las principales ventajas de utilizar un diseño en varias capas son:

- **Mejoras en las posibilidades de mantenimiento:** la única dependencia entre las capas es la interfaz que utiliza cada capa superior para comunicarse con la inferior. Esto posibilita que se puedan hacer cambios en las distintas capas sin afectar al resto.
- **Escalabilidad:** las capas pueden distribuirse en diferentes máquinas, lo que ayuda a la escalabilidad.

El diseño de la arquitectura del Módulo de Monitorización se basará en una arquitectura de cuatro capas como se puede ver en la siguiente ilustración:

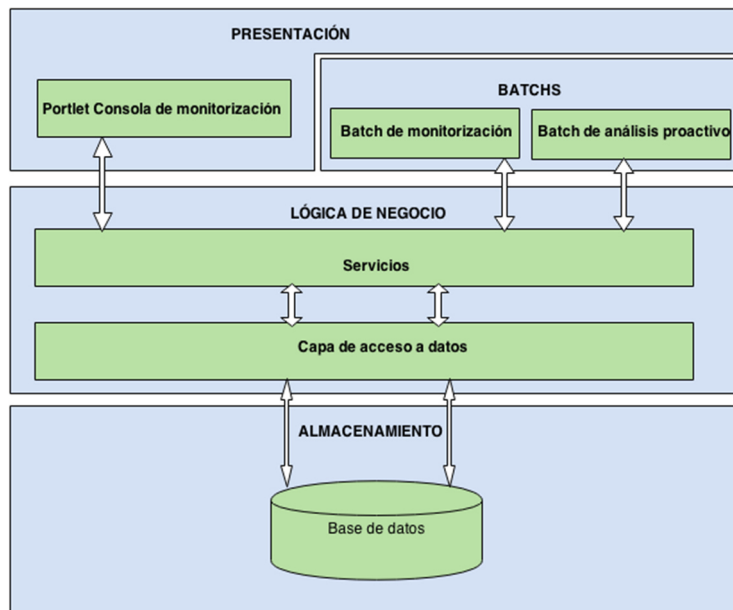


Ilustración 14. Arquitectura Módulo de Monitorización

La arquitectura del Módulo de Monitorización sigue el típico diseño de Tres Capas al que se le ha añadido una capa más, la de lo Batches.

La misión que desarrolla cada una de las capas del Módulo de Monitorización son:

- **Base de datos:** representa el Sistema Gestor de Base de Datos, encargado de la persistencia del modelo de datos de monitorización.
- **Capa de acceso a datos:** encargado del Modelo de Objetos del dominio de monitorización y de ofrecer los servicios necesarios para realizar las operaciones de acceso a datos, que serán invocadas desde la capa de servicios. La ventaja de esta capa es no crear una dependencia entre la Capa de servicios y el SGBD.
- **Servicios:** los servicios son los encargados de cumplir las funcionalidades del Módulo de monitorización junto con las reglas de negocio. Los servicios harán uso de la capa de acceso a datos para cumplir con su objetivo.
- **Batches:** la capa de Batches es la encargada de llevar a cabo la ejecución programada de funcionalidades que ofrece la Capa de servicios. La importancia de esta capa en el Módulo de Monitorización es muy alta debido a que se necesita de una capa que se ejecute de manera automática para poder monitorizar la Plataforma de Interoperabilidad. Esta capa se divide en dos batches:
 - Batch de Monitorización: se encarga de ejecutar la monitorización de cada componente activo en el Módulo de Monitorización. Además es el encargado de comprobar si algún componente se encuentra en un estado de alerta y enviar un aviso.
 - Batch de Análisis Proactivo: responsable de ejecutar el cálculo de los niveles de servicio y el envío de avisos por degradación de estos.
- **Presentación:** responsable de las vistas que se ofrecen al usuario y que le permiten a este la interacción con el Módulo de Monitorización.

7.1.1. Integración arquitectura Plataforma de Interoperabilidad

La arquitectura vista hasta el momento solo cuenta con el diseño del Módulo de Monitorización, pero como ya se ha visto en apartados anteriores, el Módulo de Monitorización se integrará con la Plataforma de Interoperabilidad ya que necesita utilizar ciertos componentes de esta para poder monitorizarla.

La arquitectura de la Plataforma de Interoperabilidad al igual que la del Módulo de Monitorización, está diseñada en capas. Gracias a esto, la integración de ambas arquitecturas resulta sencillo:

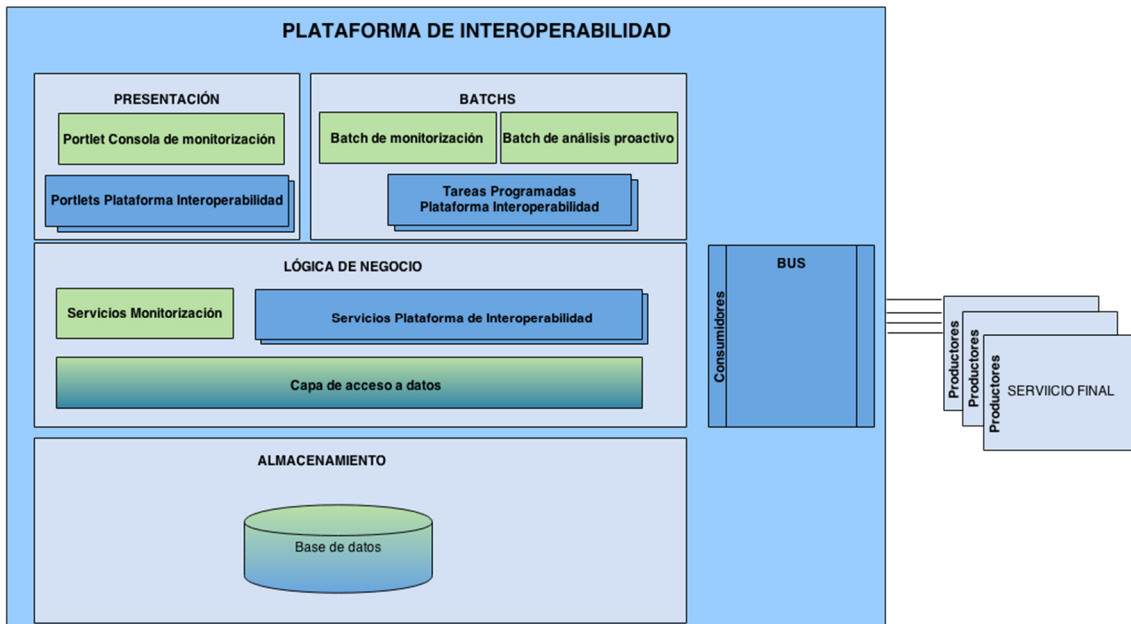


Ilustración 15. Arquitectura Plataforma de Interoperabilidad / Módulo de Monitorización

En la ilustración anterior se puede ver como se integran los componentes del Módulo de Monitorización, de color verde, con los componentes de la Plataforma de Interoperabilidad, de color azul. Además existen dos componentes compartidos, la Capa de acceso a datos y la Base de datos ya que el modelo de datos del Módulo de Monitorización se integrará con el de la Plataforma de Interoperabilidad.

A continuación se presenta la misión de aquellos componentes de la Plataforma de Interoperabilidad que no se presentaron en el anterior apartado:

- **Servicios de Plataforma de Interoperabilidad:** las funcionalidades que ofrece la Plataforma de Interoperabilidad están agrupadas en distintos módulos/servicios:
 - **Auditoría**
 - **Catálogo de producto**
 - **Catálogo de organismos**
 - **Seguridad**
 - **Gestor de avisos**

El módulo de Servicio de Monitorización pasará a formar parte de este conjunto de módulos y podrá hacer uso de las funcionalidades que estos ofrecen para cumplir sus objetivos.

- **Presentación:** como ya se dijo en *3. Definición del Ámbito tecnológico*, el Portal de Interoperabilidad, estructura sus vistas en base a portlets de Liferay. Por ello para el Módulo de Monitorización se creará un Portlet de Consola de Monitorización que formará parte de la capa de presentación junto con el resto de Portlets de la Plataforma de Interoperabilidad que forman el Portal de Interoperabilidad.
- **Tareas programadas:** batchs de la Plataforma de Interoperabilidad que cumplen la misión de ejecutar diversas funcionalidades automáticamente. Los batchs del Módulo de Monitorización formarán parte de la misma capa de Batchs que las Tareas programadas de la Plataforma de Interoperabilidad. Las distintas Tareas programadas son:
 - Eliminación de auditorías
 - Securización de auditorías
 - Volcado de auditorías
 - Solicitud de respuesta
 - Activación/Desactivación programada

La misión concreta de cada Tarea Programada se puede ver en *2.4 Tareas programadas*.

- **BUS:** esta capa se encarga del tratamiento y re-encaminamiento de las peticiones que recibe la Plataforma de Interoperabilidad a través de los consumidores y que redirige a los productores de los servicios finales. Además también realiza el camino inverso una vez recibe la respuesta a la petición por parte del servicio final.

7.2. Modelo de datos

En esta sección se describe el modelo de datos del sistema que representa la información almacenada en la base de datos relacional para dar soporte al Módulo de Monitorización.

El modelo de datos surge a partir del análisis de la información que es necesaria persistir para que el Módulo de Monitorización cumpla con sus objetivos. La representación del modelo de datos se realizará a través de un diagrama entidad relación en el que, para facilitar su lectura, no se explicitarán los atributos de las entidades.

código será clave a la hora de ejecutar la monitorización ya que permitirá la carga dinámica de los procedimientos de monitorización en tiempo de ejecución.

Cada **Indicador** tendrá unos umbrales por defecto y un **Nivel de servicio**. El **Nivel de servicio** se representa como una entidad distinta de **Indicador** aunque mantenga una relación uno a uno con esta. La razón por la que se crea una entidad específica es porque representa un concepto distinto al de **Indicador**. Un **Nivel de servicio** es el resumen del estado de un indicador en un rango de tiempo.

Los procedimientos se agruparan a través de la entidad **Perfil**. Los perfiles son un parámetro configurable del Módulo de Monitorización y como tal mantiene una relación de muchos a uno con la entidad **Configuración**, que da soporte a los datos de configuración del Módulo de Monitorización. La **Configuración** además de tener una serie de perfiles tendrá una fecha que marcará el inicio del análisis proactivo, o lo que es lo mismo, la fecha que marca a partir de qué fecha se tienen en cuenta los datos de monitorización para calcular los niveles de servicio y detectar la degradación de estos.

Los componentes tienen una serie de procedimientos con los que se monitorizan. Para relacionar los componentes con sus procedimientos se utilizará la entidad **Componente Procedimiento**. Al tener una tabla específica para esta relación se podrá establecer una frecuencia específica para un procedimiento dentro de un componente concreto. Además la entidad **Componente Procedimiento** también guarda la fecha en la que se ejecutó el procedimiento para el componente.

Tanto los componentes tarea como sus procedimientos serán dados de alta directamente desde bases de datos y no serán modificables. Los componentes de servicio podrán ser dados de alta desde la consola de monitorización. Para facilitar la asignación de procedimientos a los componentes de tipo servicio se utilizarán los perfiles. Al asignarle un perfil a un componente servicio se crearán los componentes procedimiento para el componente y los procedimientos que tenga el perfil con los valores por defecto.

Un componente tendrá todos aquellos indicadores que se obtengan a partir de los procedimientos que tenga asignados y se reflejarán en la entidad **Componente Indicador**. Al igual que pasa con los procedimientos, los indicadores para los componentes tarea son establecidos directamente en base de datos mientras que para los componentes servicio se hará a través de los perfiles.

La entidad **Componente Indicador** permite establecer umbrales específicos para cada indicador y componente concreto. Además la entidad **Componente Indicador** también tendrá la fecha en la que le fue calculado el último nivel de servicio.

Tanto los valores de los indicadores para los distintos componentes monitorizados como los niveles de servicio se almacenan en la entidad **Registro**. Un registro tendrá un componente, un indicador, una fecha, una clave, un valor y un tipo de registro. El **Tipo Registro** diferenciará entre los registros de indicador y los registros de nivel de servicio.

Para controlar los valores de los indicadores que se monitorizan se utilizarán las **Reglas Indicador**. Una regla tendrá un componente indicador, un número de repeticiones, un nivel de

monitorización y un mensaje. Además cada regla tendrá un **Aviso Monitorización** que representará el aviso que se enviará al cumplirse la regla.

Un **Aviso Monitorización** se relacionará con la entidad **Aviso Email** que pertenece a las entidades de la Plataforma de Interoperabilidad y que representa un aviso que puede ser enviado utilizando los servicios que ofrece el gestor de avisos de la Plataforma de Interoperabilidad.

Cuando se cumple una regla, además de enviar un aviso, se crea una **Incidencia**. Una incidencia tiene una relación con el registro y la regla que la generaron, un mensaje, una fecha y un nivel de monitorización. Las incidencias se crean siempre con estado de no resuelta y pueden ser resueltas a través de la consola de monitorización. Para controlar el estado de la incidencia la entidad se relaciona con la **Estado Incidencia**.

De todas las entidades vistas anteriormente se diferencian un grupo específico de ellas a las que de ahora en adelante llamaremos 'maestros'. Los maestros son entidades estáticas del modelo de datos. Todas las entidades de tipo maestro tienen solamente un atributo, que es nombre y son creadas con la base de datos.

Los maestros del modelo de datos del Módulo de Monitorización son:

- **Estado Incidencia:** que puede ser Resuelta o No resuelta.
- **Estado Componente:** que puede ser Activo o No activo.
- **Nivel Monitorización:** que puede ser Verde, Amarillo o Rojo.
- **Tipo registro:** que puede ser Indicador o Nivel de servicio.

Diseño detallado - Tablas base de datos

El diagrama de la representación en tablas de modelo entidad relación que se acaba de describir es:

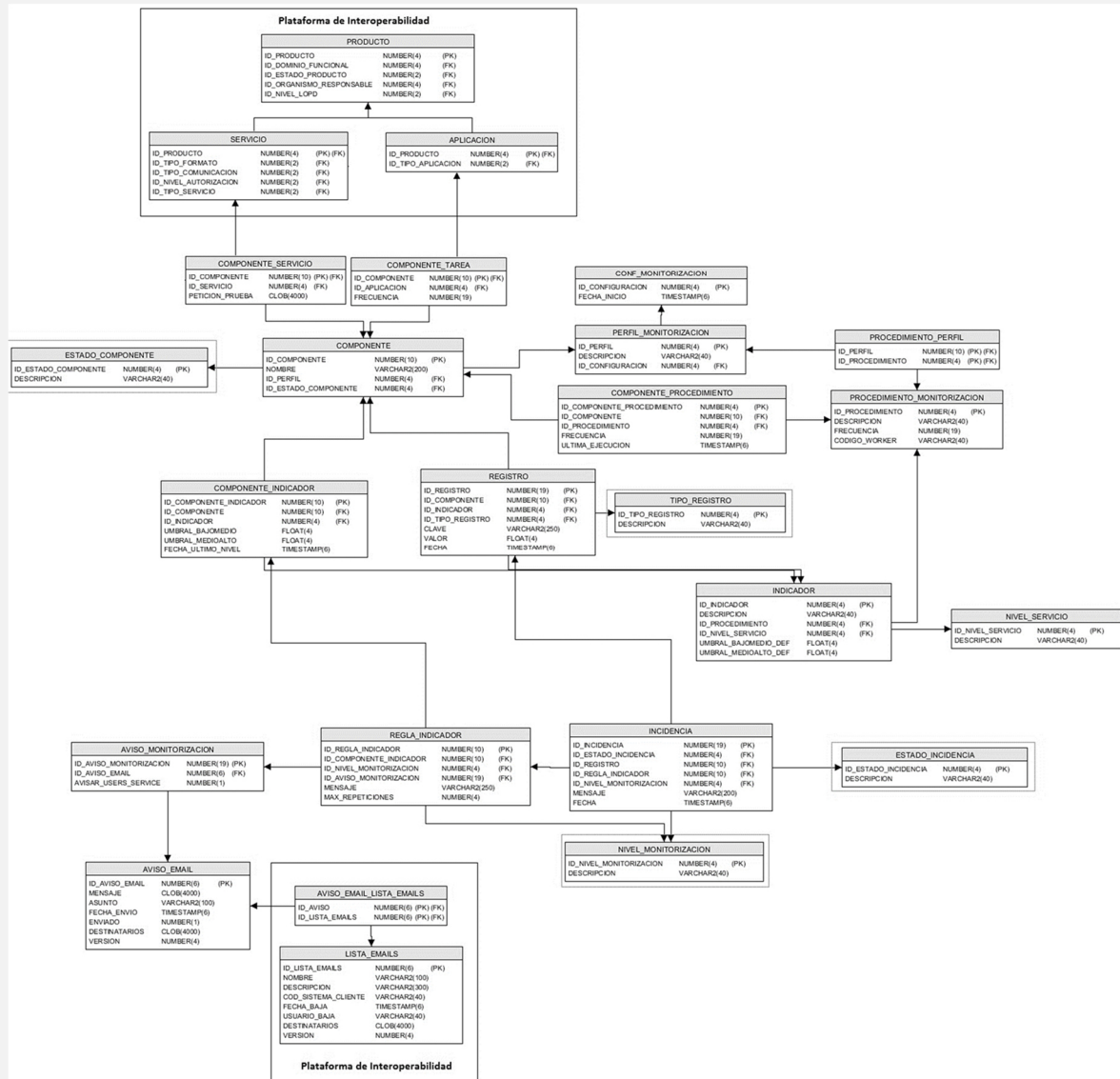


Ilustración 17. Diagrama de tablas base de datos

Adjunto a este documento se encuentra el script de creación de la base de datos.

La base de datos que dará soporte a este modelo de datos se encuentra en un servidor Oracle. El modelo de datos del Módulo de Monitorización compartirá la misma base de datos que la Plataforma de Interoperabilidad, por lo que, en el script de creación solo se crean tablas y no la base de datos en sí.

7.3. Capa de acceso a datos

El objetivo de esta capa es gestionar las operaciones de tratamiento de los datos del sistema almacenados en la base de datos. Esta capa sólo tiene conocimiento de la existencia de la base de datos del sistema.

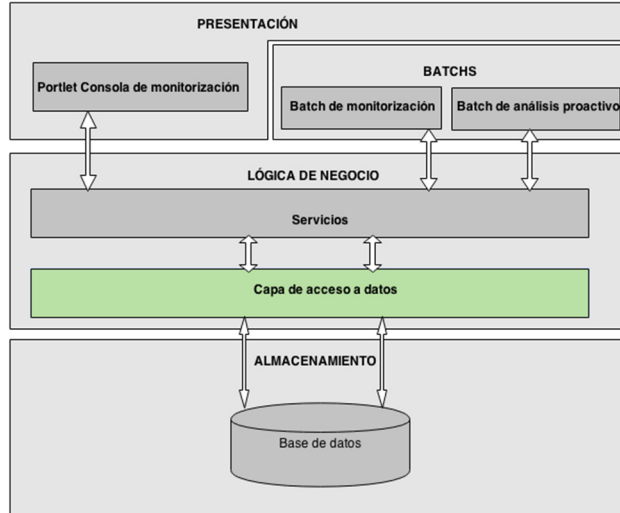


Ilustración 18. Capa de acceso a datos

La capa de acceso a datos es utilizada por la capa de servicios. Para poder cumplir con su objetivo, esta capa cuenta con un conjunto de clases que corresponden a la representación del modelo de datos. Cada clase se corresponde con una entidad o interrelación del modelo de datos. Cada objeto o instancia de una determinada clase se corresponde con una tupla de la entidad correspondiente. La función de esta capa es, por tanto, ofrecer una visión del modelo de datos utilizando para ello objetos de un lenguaje de programación orientado a objetos (en este caso Java). A partir de ahora haremos referencia a estas clases con el nombre de clases persistentes.

Diseño detallado – Creación clases persistente

Al igual que ocurre con las tablas de base de datos, las clases persistentes se agruparán junto con las clases persistentes de la Plataforma de Interoperabilidad. Toda clase persistente hereda o de la clase Entity o de EntityMaestro en función de si mapean una entidad del modelo de datos que sea un maestro o no. Además todas las entidades tiene un atributo id que es de tipo Number.

```
public class Entity<ID extends Number>
implements Serializable {

    protected ID id;

    getters y setters

}
```

```
public abstract class EntityMaestro<ID
extends Number> extends Entity<ID> {

    protected String descripcion;

    getters y setters

}
```

Como ya veremos en el apartado de la capa de servicios las clases persistentes solo serán utilizadas para la comunicación entre la capa de servicios y la capa de acceso a datos, ya que, para el tratamiento de las entidades por parte de los servicios y para su transferencia a la capa de presentación y batches se utilizará otro tipo de clases siguiendo el patrón CTO.

7.3.1. Patrón DAO

Para la implementación de la capa de acceso a datos haremos uso del patrón DAO (Data Access Object). Es un patrón introducido en el desarrollo de aplicaciones J2EE y consisten en definir una capa de acceso a los datos, de tal forma que la aplicación no tenga por qué conocer el destino de los datos, necesitando conocer únicamente las operaciones definidas para cada tipo de dato.

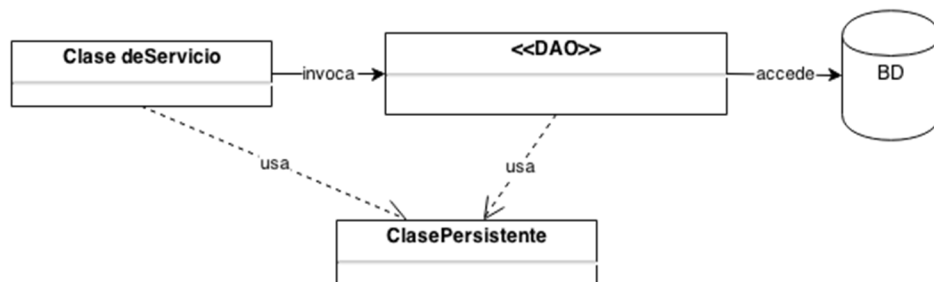


Ilustración 19. Patrón DAO

La forma usual de implementar este patrón, es definir las interfaces DAO que son las que utilizarán las clases que acceden a datos para conocer cuáles son las operaciones que pueden realizar. Siguiendo este patrón lo que se hará será definir un DAO por cada clase persistente.

Cada DAO constará de una interfaz a través de la cual ofrecerá las operaciones de acceso a datos que podrán ser invocadas por la capa de servicios. Además de un interfaz cada DAO tendrá su implementación de las operaciones de acceso a datos.

7.3.2. Framework Hibernate

Para facilitar la implementación de la capa de acceso a datos se utiliza el framework Hibernate. Hibernate es el mapeador objeto-relacional por excelencia de la comunidad Java. Un mapeador objeto-relacional es una herramienta que facilita la conversión entre las entidades almacenadas en una base de datos relacional y las clases persistentes del dominio de la aplicación.

Hibernate es un framework que se integra perfectamente con Spring, un framework que se explicará en la capa de servicios. Con Spring se configurará Hibernate utilizando los beans y la inyección de dependencias para los diferentes objetos requeridos por Hibernate. Además también se configurarán las clases persistentes a las que serán mapeadas las tablas de la base de datos, para ello utilizaremos la configuración a través de mapeos con archivos xml.

Además de utilizar el framework Hibernate se utilizará un framework propio de la empresa y que supone una capa entre el Hibernate y los DAO de la Plataforma de Interoperabilidad. En concreto, se usará una implementación de Hibernate, en la cual se ofrece un DAO genérico de acceso a datos con operaciones básicas ya implementadas (inserts, finders, etc.). Además este DAO genérico es extendido a su vez por BaseDao, un DAO de la Plataforma de Interoperabilidad

para búsqueda genérica por Criteria. En resumen, los DAO's que se creen para el Módulo de Monitorización deberán extender de BaseDao obteniendo de este modo ciertas funcionalidades genéricas.

A continuación se muestran los DAO's genéricos:

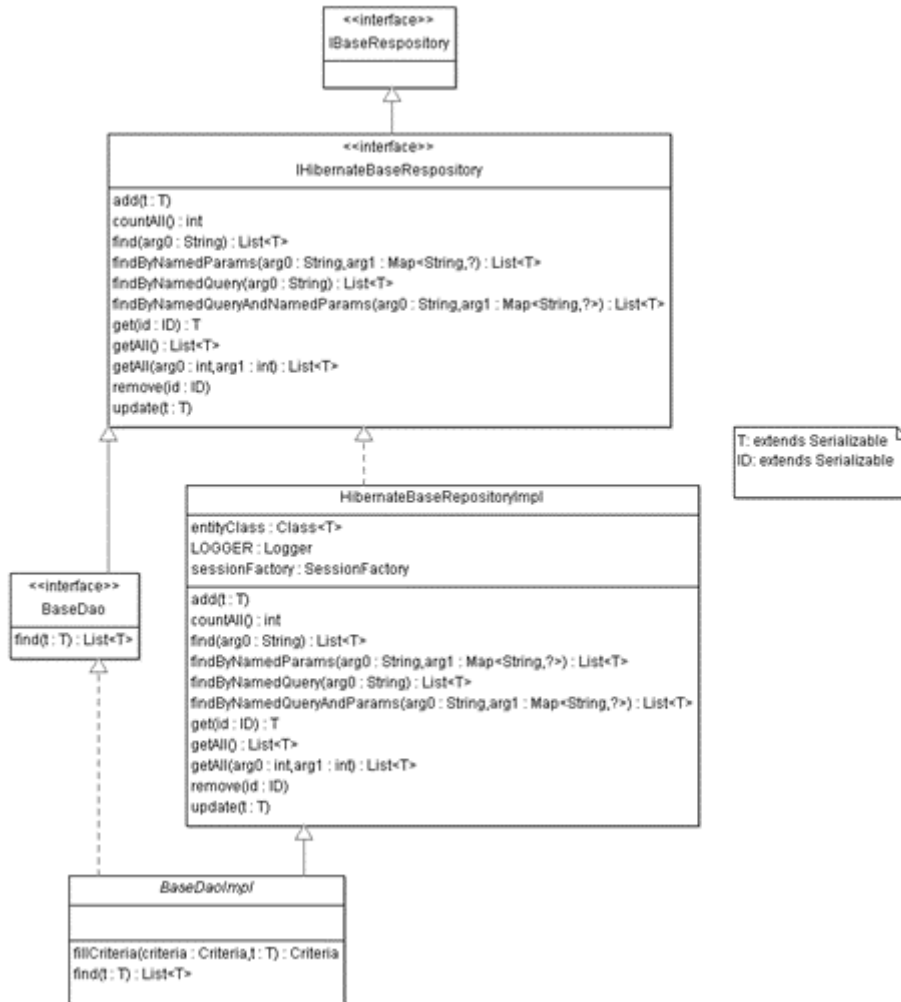


Ilustración 20. DAO's genéricos

Todas las clases que hereden de BaseDaoImpl deberán implementar el método protegido fillCriteria().

Diseño detallado – Creación de un DAO

Para la creación de un DAO, lo primero es tener una clase persistente, en este caso utilizaremos la clase Registro:

```
public class Registro extends Entity<Long> {
    private Componente componente;
    private Indicador indicador;
    private TipoRegistro tipoRegistro;
    private Float valor;
    private String clave;
    private Date fecha;
    constructor vacio, getters, setter, toString, hashCode y equals}

```

Como se puede ver la clase tiene sus atributos y hereda de la clase Entity, pasándole mediante el uso de Genéricos de java el tipo de dato del id de Registro. A continuación se muestra el mapeo de Hibernate correspondiente:

```
<hibernate-mapping><class name="com.amt.entities.registro.Registro" table="REGISTRO">
    <id name="id" type="java.lang.Long"><column name="ID_REGISTRO"/>
    <generator class="native"><param name="sequence">REGISTRO_SEQ</param></generator></id>
    <many-to-one name="componente" column="ID_COMPONENTE" not-null="true"/>
    <many-to-one name="indicador" column="ID_INDICADOR" not-null="true"/>
    <many-to-one name="tipoRegistro" column="ID_TIPO_REGISTRO" not-null="true"/>
    <property name="valor" type="java.lang.Float" column="VALOR"/>
    <property name="clave" type="string" column="CLAVE"/>
    <property name="fecha" type="java.util.Date" column="FECHA"/></class></hibernate-mapping>

```

Lo primero que se hace en el mapeo es establecer la clase persistente y la tabla con la que se mapea. Luego se indica el tipo de dato del id de Registro, la columna en la que se mapea el id y el nombre de la secuencia que registra el id de Registro.

Después se establecen los mapeos de los distintos atributos. Aquellos atributos que son propiedades se les indica el tipo de dato que son y el nombre de la columna. Los atributos que hacen referencia a otras entidades del modelo de datos como son Componente e Indicador solo se le indican el nombre del atributo. Hibernate detectará que son entidades y cargará sus respectivos mapeos.

Por último se crea la interfaz RegistroDao que debe heredar de BaseDao y que es implementada por RegistroDaoImpl que a su vez hereda de BaseDaoImpl. Como ya dijimos anteriormente, RegistroDaoImpl debe implementar el método fillCriteria().

Además RegistroDao añadirá dos funcionalidades más a las que hereda de BaseDao, estas son buscarRegistros y countRegistros que serán utilizadas por los servicios de búsqueda de Registros para crear un buscador de registros en el portlet de la Consola de monitorización.

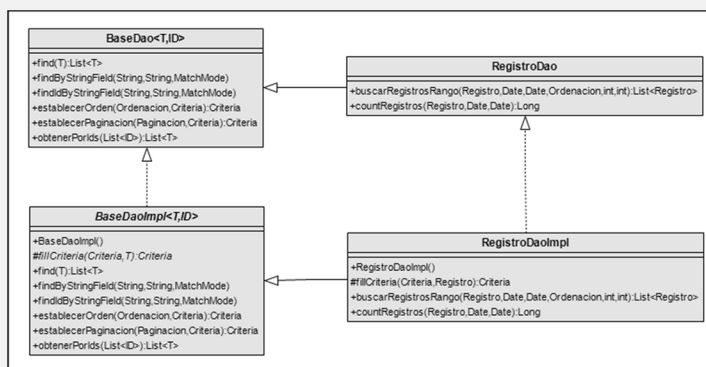


Ilustración 21. Registro DAO

7.3.3. Refinamiento del diseño de acceso a datos

Uno de los problemas que puede suponer utilizar el mapeo para recuperar objetos desde base de datos es que por ejemplo si se llama al método *get(Long id):Registro* de *RegistroDao* el objeto que se nos va a devolver contendrá todos los datos del registro cuyo id le pasemos. Con todos los datos implica también que al tener un atributo *Componente*, lo que va a hacer Hibernate es mapearnos también esa entidad y todas las que tenga dentro. Esto puede desencadenar una secuencia de peticiones a base de datos demasiado grande.

Si por ejemplo, lo único que interesase del registro fuese su valor, podríamos obtenerlo fácilmente con la siguiente query:

```
SELECT valor FROM Registro WHERE id=1;
```

Sin embargo, si lo hacemos con mapeo la petición será menos eficiente ya que primero hará:

```
SELECT * FROM Registro WHERE id=1;
```

Después de lanzar la anterior consulta, a partir del id del componente y del id del indicador lanzará otras queries para obtener las distintas entidades que tiene la clase *Registro*.

Es por esto que en aquellos casos en los que utilizar el mapeo para recuperar datos suponga una gran pérdida de rendimiento respecto a la comodidad que supone recuperar datos utilizando objetos java, lo que se hará será que el DAO implemente un método en el que utilice HQL para resolver la operación de acceso a datos.

HQL es un lenguaje de consulta de Hibernate que se parece a SQL. Sin embargo, comparado con SQL, HQL es completamente orientado a objetos y comprende nociones como herencia, polimorfismo y asociación.

Además el uso de HQL puede facilitar en ocasiones la realización de consultas, sobre todo cuando estas impliquen combinación de entidades que no estén relacionadas directamente entre sí.

Diseño detallado – Consulta HQL

A continuación se muestra un ejemplo de implementación HQL que se utilizará en *ComponenteServicioDao* para recuperar solo los ids y el nombre de los componentes de tipo servicio activos en Monitorización y en el Catálogo de Productos de la PLATINT.

```
public List<SimpleEntity<Integer>> obtenerIdsMonitorizar() {
    StringBuilder sbQuery = new StringBuilder();
    sbQuery.append("SELECT new com.amt.services.base.entities.SimpleEntity(cs.id,cs.nombre) ");
    sbQuery.append("FROM ComponenteServicio as cs ");
    sbQuery.append("INNER JOIN cs.servicio s ");
    sbQuery.append("WHERE cs.estado.id=:ESTADO_COMPONENTE AND s.estadoProducto=:ESTADO_SERVICIO");
    Query query = this.getSession().createQuery(sbQuery.toString())
        .setInteger("ESTADO_COMPONENTE", Constantes.ESTADO_COMPONENTE_ACTIVO)
        .setInteger("ESTADO_SERVICIO", Constantes.ESTADO_PRODUCTO_ACTIVO);
    return query.list();
}
```

Este método devuelve una lista de *SimpleEntity* que es un objeto que solo tiene dos atributos, un id y un nombre. Utilizando este método se evita tener que cargar otras entidades que tenga un *Componente* como pueden ser una lista

7.3.4. Diagrama de clases persistentes

Para finalizar con la capa de acceso a datos se mostrará el diagrama de clases persistentes utilizado para representar el modelo de datos del Módulo de Monitorización. El diagrama de clases persistente es muy similar al diagrama de tablas. Esto se debe al uso del mapeador objeto-relacional.

Una de las principales diferencias es que a nivel de clase persistente una entidad si puede tener un conjunto o una lista de otras entidades. Sin embargo, el modelo relacional y la teoría de normalización de bases de datos impiden que haya listas de datos en las entidades de la base de datos.

En el diagrama no se muestran los métodos de las clases para facilitar su lectura. Todas las clases tendrán un constructor vacío y los métodos getters, setters, toString, hashCode y equals.

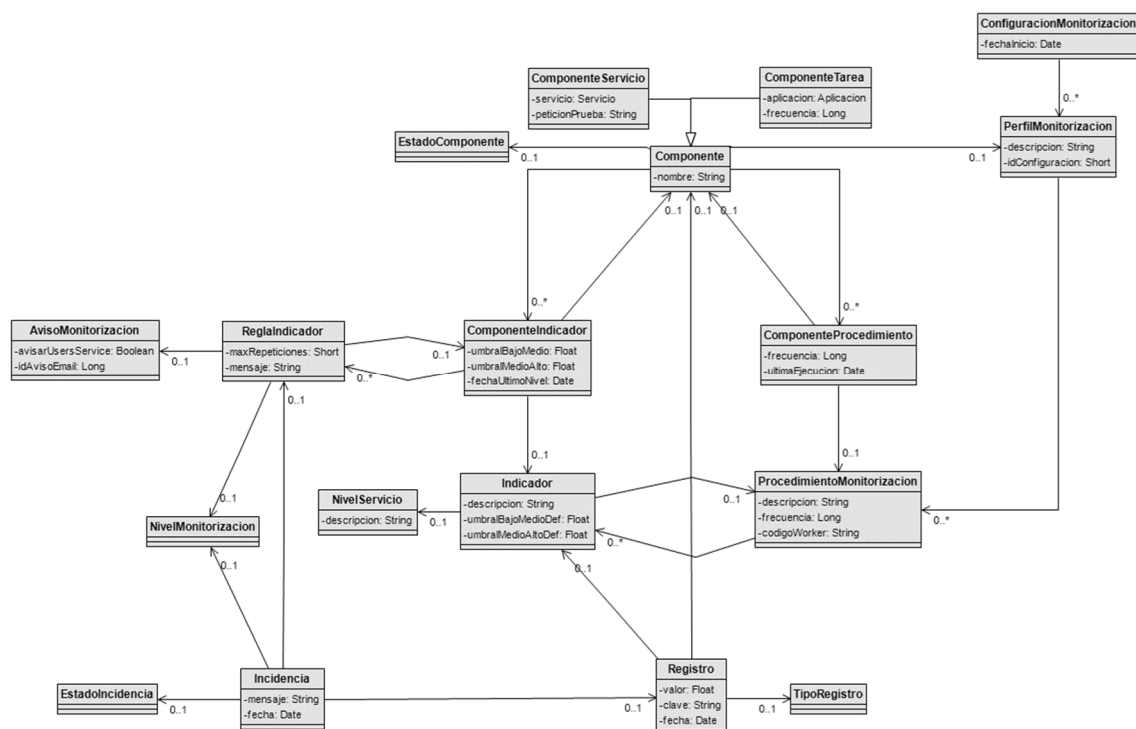


Ilustración 22. Diagrama de clases de persistencia

7.4. Capa de servicios

El propósito de esta capa es aportar la funcionalidad necesaria para que el Módulo de Monitorización cumpla con sus objetivos. Se comenzará definiendo las divisiones de la capa de servicios y explicando el framework Spring y el patrón CTO. Por último se explicará el diseño y la misión de cada uno de los grupos que componen la capa de servicios, así como, los patrones utilizados.

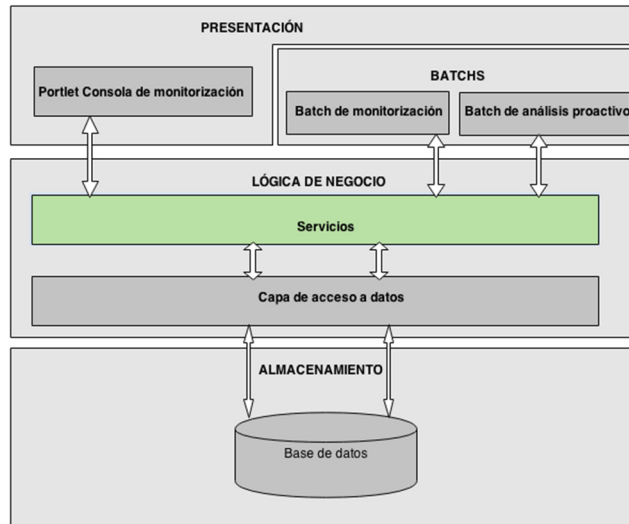


Ilustración 23. Capa de servicios

Dentro de la capa de servicios se distinguen varios grupos/paquetes:

- **Servicios:** representa las funcionalidades que se ofrecen a las capas superiores de presentación y batch. Las funcionalidades ofrecidas son agrupadas en función de la entidad con la que se relacionan. De este modo si se quiere por ejemplo aportar una funcionalidad de búsqueda de Registros el encargado de ofrecerla a las capas superiores será una clase específica que representará el Servicio de Registro. Además existirá algún servicio que no se relaciona con una entidad, ya que, las funcionalidades que ofrece implican el uso de servicios de varias entidades.
- **Procedimientos:** este grupo tiene el objetivo de implementar los distintos procedimientos de monitorización. Los servicios que se ofrecen desde este grupo no serán ofrecidos directamente a las capas superiores sino que se creará una entidad ficticia, Servicio de Monitorización, que formará parte del grupo de Servicios de entidad y que servirá de enlace para que las capas superiores accedan a los procedimientos de monitorización.

7.4.1. Spring Framework

Como ya hemos dicho, la capa de servicios hará uso de los servicios que ofrece la capa de acceso a datos, a sus vez los propios componentes de la capa de servicios pueden hacer uso de componentes de la misma capa y además los componentes de la capas superiores también lo harán. Esto provoca que haya una gran dependencia entre los componentes que forman las distintas capas.

Para solucionar este problema y que cada componente que vaya a utilizar a otro no tenga que preocuparse por su dependencia ni por instanciarlo se utilizará Spring.

Spring es un framework de desarrollo de aplicaciones web que introduce la inyección de dependencias. La inyección de dependencias permite que una clase puede declarar qué objetos necesita sin necesidad de instanciarlos, de eso se ocupará Spring que será el encargado de realizar la instanciación y la destrucción. Esto nos beneficia a la hora de disponer de un código mucho más eficiente y limpio.

Diseño detallado – Inyección de dependencias con anotaciones Spring

Por ejemplo, un servicio que necesite utilizar el DAO de la entidad Componente lo declarará como:

```
@Autowired
private ComponenteDao componenteDao;
```

Como se puede ver el servicio no instancia la implementación del DAO sino que solo hace referencia al interfaz que desea utilizar.

Todos aquellos componentes que se quieran utilizar para ser inyectados como dependencias deben de llevar la anotación oportuna. En el caso de la implementación de los DAO's estas tendrán que llevar la anotación @Repository:

```
@Repository
public class ComponenteDaoImpl {.....}
```

Otros componentes como son los servicios o los procedimientos llevarán la anotación @Service o @Component respectivamente.

Durante la inicialización del contexto de Spring se realiza un escaneo de todos los paquetes buscando las anteriores anotaciones y creando un bean por cada una encontrada. De este modo se crea un contenedor Spring que contendrá instancias de las distintas clases.

Volviendo al ejemplo del servicio que necesitaba utilizar ComponenteDao, al tener la anotación @Autowired lo que se hará será que se le inyectará un bean de los existentes en el contenedor de Spring que implemente el interfaz con el que tiene dependencia.

7.4.2. Patrón CTO

Como ya se ha adelantado en la descripción de la capa de acceso a datos, la capa de servicios utiliza los servicios ofrecidos por los DAO's. Para ello, la capa de acceso a datos debe utilizar también las clases persistentes.

El uso de las clases persistentes en la capa de servicios solo se hará para la comunicación con la capa de accesos a datos. Antes y después de esta comunicación siempre se producirá una conversión de clase persistente a CTO o viceversa.

La motivación de porque utilizar un modelo de CTO's y no utilizar las clases persistentes son dos:

- Supone una capa de abstracción que permite ocultar aquellos componentes propios del módulo de datos que no se quiera que sean visibles a través de los servicios.
 - Por ejemplo, si tuviésemos una tabla Persona representada por su correspondiente clase y quisiéramos tener un servicio que devolviese los datos de la Persona pero sin mostrar que tiene una propiedad *contraseña*, lo que haríamos sería crear una clase PersonaCto que sería una representación de Persona pero sin la propiedad contraseña. El servicio devolvería la clase PersonaCto.
- Las clases persistentes están condicionadas por el mapeo que se realiza con la base de datos. En ciertas ocasiones un servicio puede tener que devolver información que se encuentra en distintas clases o incluso información que es calculada a partir de los datos de una clase. El patrón CTO permite la creación de una clase personalizada que contenga los datos necesarios que debe devolver un servicio.
 - Por ejemplo, la clase Componente no tiene un conjunto de registros sino que es un Registro quien tiene un Componente. Si se quiere tener un servicio que devuelva los datos de un Componente junto con sus registros lo que se hará será crear una clase ComponenteCto que tendrá las propiedades de Componente, pero que además, contará con un conjunto de Registros.

7.4.2.1. Dozer

Para facilitar la conversión de clases persistentes a CTO's se utilizara la herramienta Dozer.

Dozer es un mapeador entre objetos java, copia recursivamente datos de un objeto a otro. Soporta mapeos de propiedades simples y complejas.

Suponiendo que las clases del objetos origen y destino, tienen los nombres de sus atributos en común, Dozer mapea implícitamente las propiedades, y copia los valores de cada una, del objeto fuente, a una nueva instancia de la clase destino. La copia se hace a través de los getters y setters, de las clases origen y destino, respectivamente.

Si las clases que se desean mapear no comparten el nombre de los atributos se podrán configurar los mapeos utilizando un fichero de configuración xml.

El resultado del mapeo será una nueva instancia de la clase especificada como destino, con los mismos valores en los atributos cuyos nombres sean iguales a los de la clase origen. Si alguno de los atributos mapeados fuesen de tipos distintos, Dozer automáticamente hará la conversión del tipo de dato de la clase origen al tipo de dato de la clase destino.

A continuación se muestra como se utilizaría Dozer en una llamada que se realice desde la capa de presentación a un servicio para obtener un componente por su id:

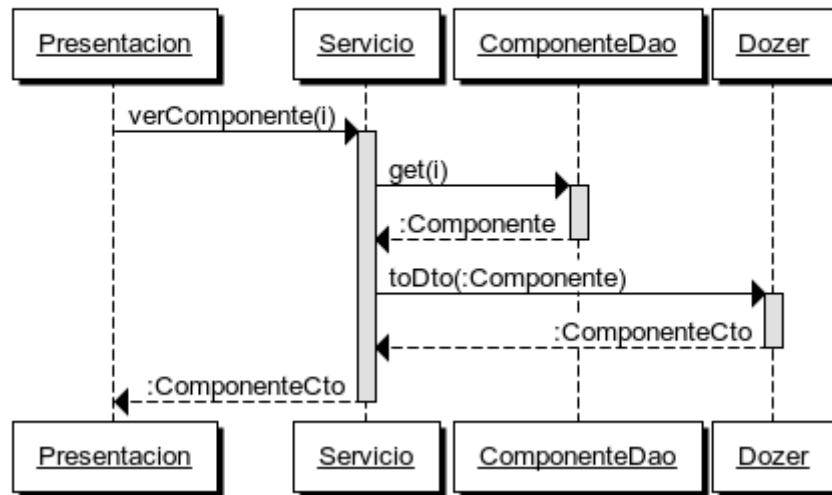


Ilustración 24. Diagrama secuencia Dozer

7.4.3. Servicios

Como ya se adelantaba al inicio de este apartado, la estructura de servicios que se ofrecerá desde el Módulo de Monitorización se basará en crear un servicio por cada entidad.

Se decide crear un servicio por cada entidad siguiendo la estructura modular de servicios de la Plataforma de Interoperabilidad. Esto permite agrupar las funcionalidades que ofrece el Módulo de Monitorización en función de la entidad a la que hacen referencia.

Aunque hasta el momento se ha establecido un servicio por entidad, esto no es del todo cierto. Existen varias entidades para las que no cobra sentido crear un servicio único. Estas entidades son las entidades maestro y aquellas entidades no maestro que solo tengan sentido a la hora de aportar una funcionalidad si se relacionan con otra entidad.

A continuación se explicará el patrón Fachada utilizado para la construcción de los servicios y luego se detallarán los servicios del Módulo de Monitorización.

7.4.3.1. Patrón Fachada

El patrón Fachada proporciona una interfaz unificada para un conjunto de subsistemas. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar.

Se utilizará este patrón en la construcción de los servicios con el fin de:

- Simplificar el acceso a un conjunto de funcionalidades proporcionando una única clase que todos utilizan para comunicarse con dicho conjunto de funcionalidades.
- Reducir la complejidad y minimizar dependencias

Cada servicio constará de una interfaz y una implementación, de manera que todas las funcionalidades que el servicio quiera compartir se publicarán a través de las interfaces. Esto permite desacoplamiento entre los clientes de los servicios (capas superiores y otros servicios) y la implementación de estos.

7.4.3.2. Servicios de Maestros

Las funcionalidades que debe proporcionar el Módulo de Monitorización para las entidades de tipo maestro son dos:

- Recuperar todos los posibles valores de una entidad maestro.
- Recuperar una tupla por su id.

Como estas funcionalidades son comunes para todos los maestros y ninguno tiene funcionalidades a mayores se creará el servicio MaestrosMonitorizacionService.

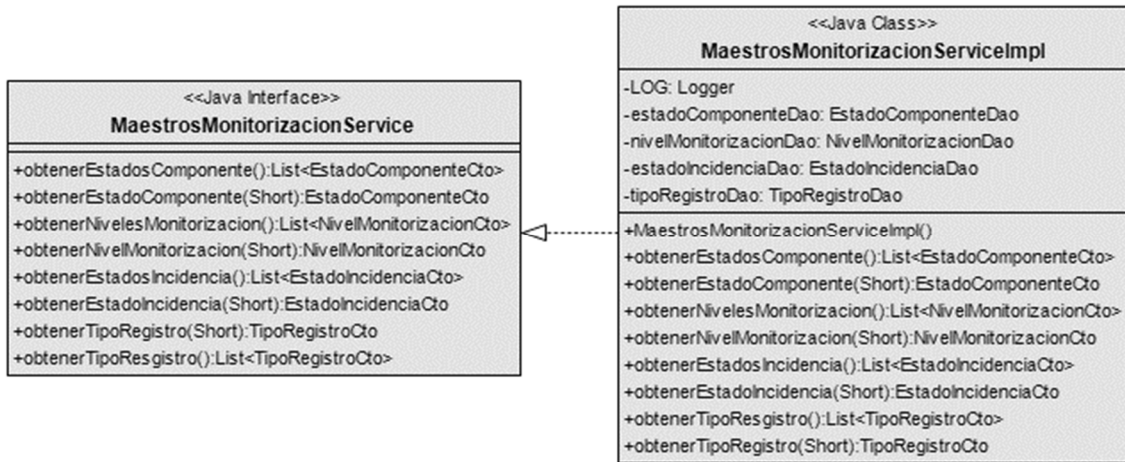


Ilustración 25. Diagrama Servicios Entidades Maestro

Como se puede ver cada maestro del Módulo de Monitorización cuenta con dos métodos, uno que permite recuperar un maestro a partir de su id y otro que recupera una lista con todos los maestros.

Destacar también que como introdujimos al principio del apartado de la capa de servicios, los objetos que devuelven los servicios son de tipo CTO y no devuelven directamente la clase de persistencia.

Además como se dijo en la presentación del framework Spring, la implementación del servicio de los maestros debe llevar la anotación @Component para que pueda ser inyectada como dependencia en los componentes que quieran utilizar el servicio.

7.4.3.3. Servicios de entidad

Siguiendo la estructura modular para la creación de servicios de la Plataforma de Interoperabilidad, se creará un servicio por cada una de las siguientes entidades:

Componente, Componente Tarea, Componente Servicio, Configuración, Incidencia, Registro y Regla Indicador.

El resto de entidades no tendrán un servicio específico ya que sus funcionalidades irán siempre relacionadas con alguna de las anteriores entidades. Por ejemplo, no existirá un servicio para Perfil ya que los perfiles son parte de la entidad Configuración y por tanto su creación, modificación y eliminación se realizará utilizando los servicios de la entidad Configuración.

Además al estructurar los servicios en base a entidades, se pueden reutilizar las interfaces y las implementaciones genéricas para servicios de entidades con las que cuenta la Plataforma de Interoperabilidad.

Este conjunto de interfaces e implementaciones consiste en una jerarquía de entidades de las que deberán heredar los nuevos servicios de entidad que se creen. Dependiendo de la funcionalidad que se quiera aportar al servicio se heredará de un nivel de la jerarquía o de otro.

Aunque en un primer momento resulte complicado la comprensión de este modelo de servicios de entidades, realizar un diseño en base a este modelo supondrá una gran ventaja a la hora de que el Módulo de Monitorización sea más comprensible, más fácil de mantener y evite la duplicación de código.

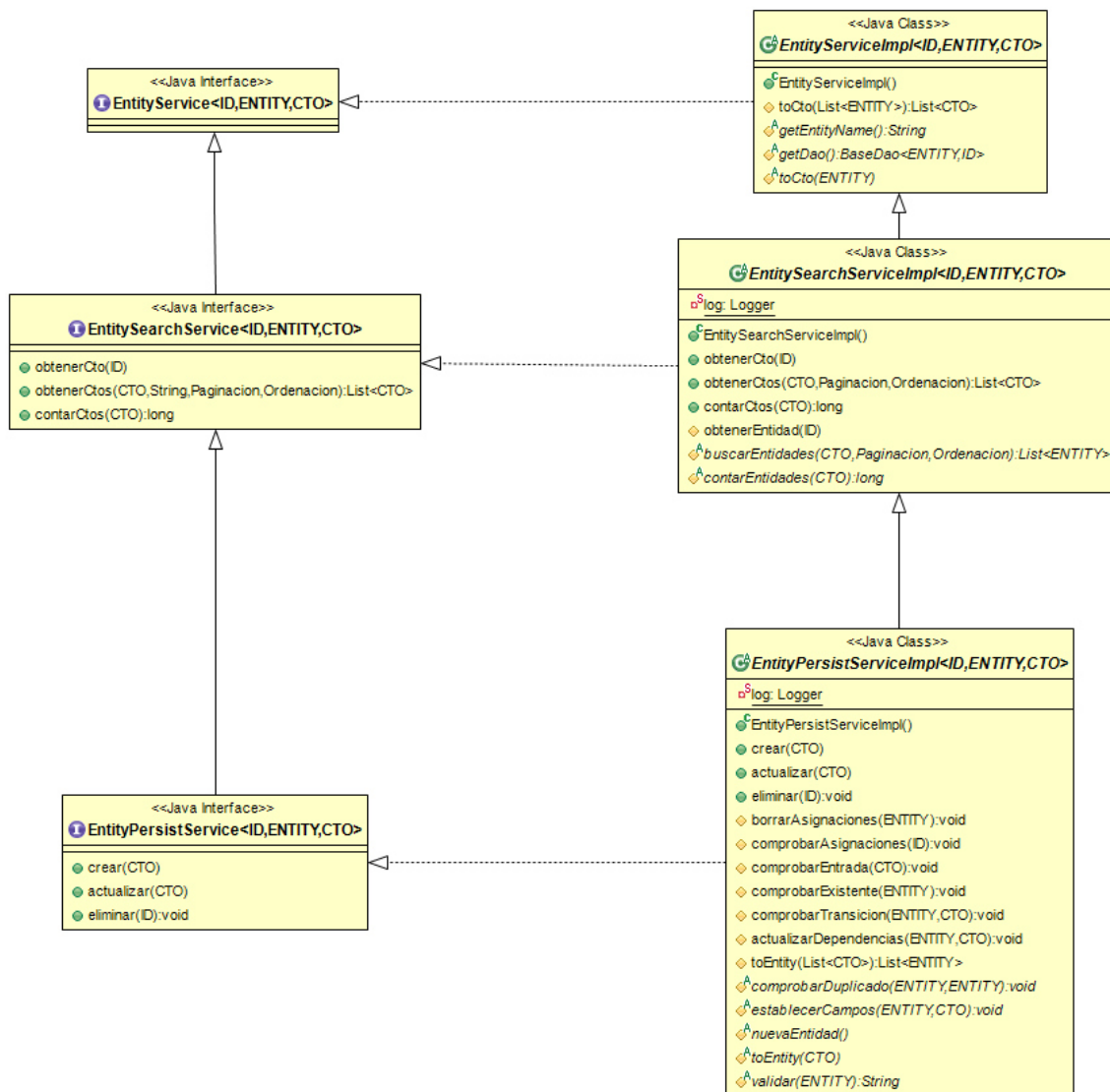


Ilustración 26. Jerarquía de interfaces e implementaciones Servicios Entidad

Para describir la anterior jerarquía lo haremos en función de los distintos niveles de los que podría heredar un servicio de entidad.

Nivel 0: Heredar el interfaz `EntityService` y la implementación `EntityServiceImpl`

Es el nivel más alto de la jerarquía, al heredar de este nivel la implementación que hereda de `EntityServiceImpl` tiene que implementar los siguientes métodos abstractos:

- `getEntityName()`: debe devolver el nombre de la entidad para que pueda ser utilizado para los logs.
- `getDao ()`: debe devolver el DAO de la entidad.
- `toCto(ENTITY)`: recibe una entidad y la convierte a CTO.

Heredar de este nivel no ofrece ningún servicio al resto de componentes, ya que, como se puede ver el interfaz no tiene ningún método. Sin embargo si un servicio hereda de `EntityServiceImpl` e implementa los métodos abstractos vistos, podrá hacer uso de siguiente funcionalidad ya implementada:

- `toCto (List<ENTITY>): List<CTO>`: recibe una lista de entidades y las convierte a CTO.

Nivel 1: Heredar el interfaz `EntitySearchService` y la implementación `EntitySearchServiceImpl`

Un servicio heredará de este nivel cuando quiera ofrecer funcionalidades para buscar o contar entidades. Al heredar de este nivel los métodos abstractos que son necesarios implementar además de los del nivel 0 son:

- `buscarEntidades(CTO, Paginacion, Ordenacion) :List<CTO>`: realiza una búsqueda de la entidad que cumpla con los parámetros del CTO y además ordena y pagina la búsqueda en función de los parámetros del objeto Ordenación y Paginación. Debe devolver una lista de CTO's con el resultado de la búsqueda.
- `countEntidades(CTO):long`: cuenta el número de entidades que cumplen con los parámetros del CTO.

Con los métodos abstractos implementados, el servicio que hereda de este nivel ya ofrece a través de su interfaz los métodos:

- `obtenerCtos(CTO, Paginacion, Ordenacion) :List<CTO>`: realiza una búsqueda de la entidad que cumpla con los parámetros del CTO y además ordena y pagina la búsqueda en función de los parámetros del objeto Ordenación y Paginación. Debe devolver una lista de CTO's con el resultado de la búsqueda.
- `contarCtos(CTO):long` : cuenta las entidades que cumplen los parámetros del CTO.
- `obtenerCto(ID):CTO` : busca una entidad por su id y devuelve un CTO.

Además el servicio que herede de este nivel podrá hacer uso en la implementación de sus funcionalidades del siguiente método:

- `obtenerEntidad(ID):ENTITY` : busca una entidad por su id y devuelve una ENTITY.

Nivel 3: Heredar el interfaz EntityPersistService y la implementación EntityPersistServiceImpl

El nivel 3 es el último nivel y de este nivel deben heredar aquellos servicios que quieran ofrecer funcionalidades para crear, modificar o eliminar entidades.

Al igual que en el resto de métodos existen una serie de métodos abstractos que deben ser implementados:

- *comprobarDuplicado(ENTITY,ENTITY)*: comprueba si dos entidades son iguales, si lo son devuelve una excepción.
- *establecerCampos(ENTITY,CTO)*: establece los campos del CTO a la ENTITY, en este caso no se puede utilizar el conversor Dozer ya que no se quiere que se genere una nueva entidad sino que se quiere modificar la actual.
- *nuevaEntidad()*: instancia un entidad.
- *toEntity(CTO):ENTITY*: convierte un CTO a una ENTITY.
- *validar(ENTITY): String*: valida si los datos de una entidad son correctos para entrar en base de datos. Si no lo son devuelve un String informando de los errores.

Implementando los anteriores métodos abstractos el servicio que hereda de este nivel ofrecería las siguientes funcionalidades:

- *crear(CTO)*: crea una entidad a partir de un CTO.
- *actualizar(CTO)*: actualiza una entidad a partir de un CTO.
- *eliminar(ID)*: elimina una entidad a partir de un ID.

Además este nivel cuenta con una serie de métodos protegidos sin implementar. Estos métodos aunque están sin implementar no se declaran abstracto debido a que dependiendo de la entidad pueden tener sentido o no. Estos métodos son:

- *borrarAsignaciones (ENTITY)*: si la entidad tiene otras entidades que le hacen referencia las elimina.
- *comprobarAsignaciones(ID)*: compruebas si una entidad tiene otras entidades que le hacen referencia para saber si se puede eliminar o no.
- *comprobarEntrada(CTO)*: comprueba que los datos introducidos en el CTO cumplan las lógicas.
- *comprobarExistente(ENTITY)*: comprueba si una entidad ya existe.
- *comprobarTransicion(ENTITY, CTO)*: comprueba si una entidad se puede actualizar a los datos del CTO.
- *actualizarDependencias(ENTITY,CTO)*:actualiza valores de otras entidades que dependen de ENTITY con los valores de CTO.

Todos estos métodos que acabamos de describir son utilizados por los métodos ya implementados de crear, actualizar y eliminar. Si alguno de ellos no es implementado simplemente se ejecutará pero no realizará ninguna acción.

Una vez vistos los distintos niveles de los que pueden heredar los servicios que se van a crear, se muestra un diagrama con los servicios, sus herencias y nuevas funcionalidades. Para facilitar su lectura solo se ha puesto en el diagrama el interfaz de cada servicio y no su implementación.

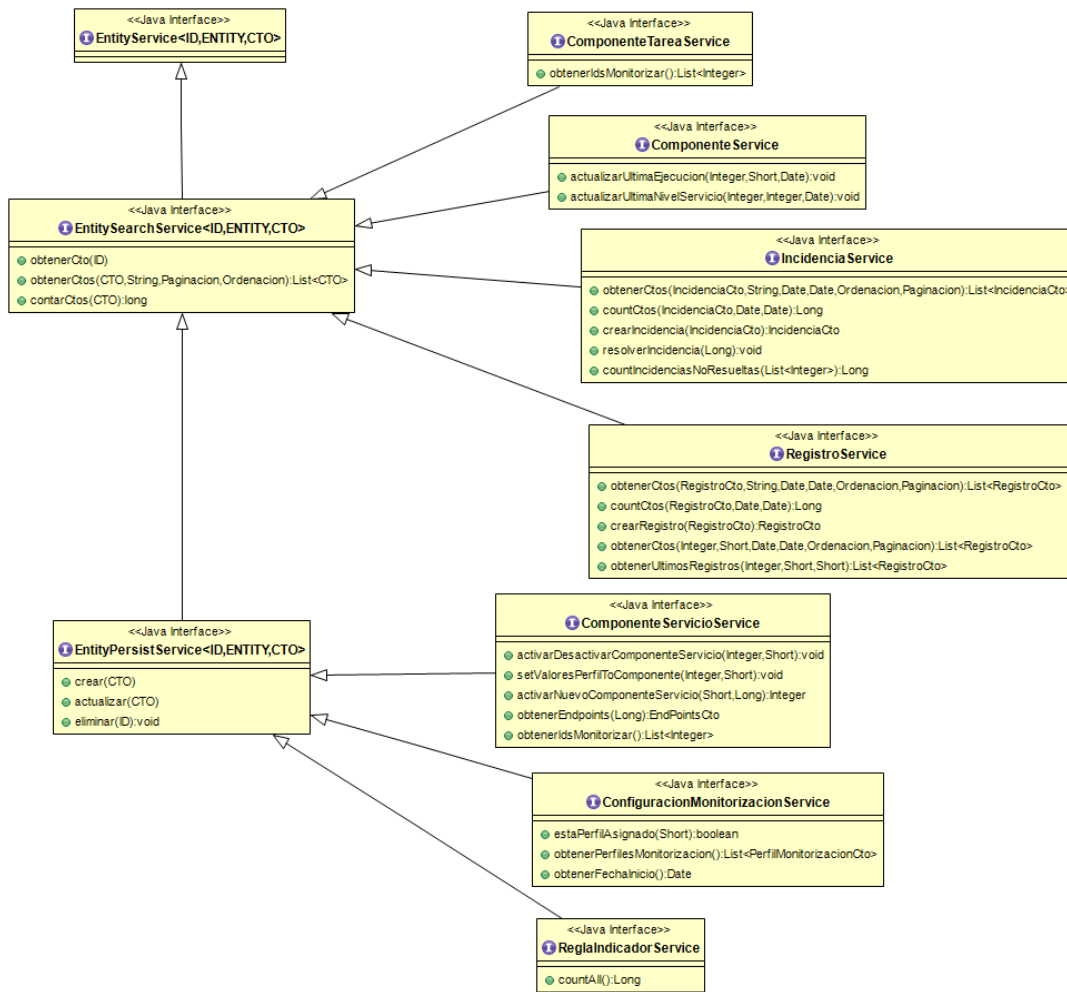


Ilustración 27. Diagrama servicios entidad Módulo de Monitorización

A continuación se detallan las funcionalidades que cada servicio debe añadir:

ComponenteTareaService

obtenerIdsMonitorizar():List<Integer>: devuelve los ids de los componentes tarea que están activos en monitorización y en el Catálogo de Productos de la Plataforma de Interoperabilidad.

ComponenteService

actualizarUltimaEjecucion(Integer,Short,Date): establece la fecha de última ejecución a la entidad ComponenteProcedimiento que tenga el id de componente y el id de procedimiento que se introducen como parámetros.

actualizarUltimoNivelServicio(Integer,Integer,Date): establece la fecha en la que se calculó el último nivel de servicio en la entidad ComponenteIndicador que tenga el id de componente y el id de indicador que se introducen como parámetros.

IncidenciaService

obtenerCtos(IncidenciaCto,Date,Date,Ordenacion,Paginacion): List<IncidenciaCto> : realiza una búsqueda de incidencias que cumplan los parámetros de IncidenciaCto, con la paginación y ordenación indicada y entre las fechas que se pasan como parámetro.

countCtos (IncidenciaCto, Date, Date): Long: cuenta las incidencias que cumplen con los parámetros de IncidenciaCto y que se encuentran entre las fechas indicadas.

crearIncidencia(IncidenciaCto): IncidenciaCto : este método crea una Incidencia con los datos de IncidenciaCto. La incidencia que se crea siempre tiene estado *no resuelta*.

resolverIncidencia(Long): resuelve una incidencia a partir del id de esta.

countIncidenciasNoResueltas(List<Integer>):Long: cuenta cuántos componentes cuyos ids son pasados como parámetro tienen al menos una incidencia sin resolver.

RegistroService

obtenerCtos(RegistroCto,Date,Date,Ordenacion,Paginacion): List<RegistroCto> : realiza una búsqueda de registros que cumplan los parámetros de *RegistroCto*, con la paginación y ordenación indicada y entre las fechas que se pasan como parámetro.

countCtos (RegistroCto, Date, Date): Long: cuenta las incidencias que cumplen con los parámetros de *RegistroCto* y que se encuentran entre las fechas indicadas.

crearRegistro(RegistroCto): RegistroCto: este método crea un registro con los datos de *RegistroCto*.

obtenerUltimosRegistros (Integer, Short, Short): List<RegistroCto>: obtiene una lista de los registros más actuales que tengan el componente e indicador cuyos ids se pasan como parámetro. La cantidad de registros que recupera también es introducida como parámetro.

ComponenteServicioService

activarDesactivarComponenteServicio(Integer,Short): activa o desactiva un componente a partir del id introducido y del id del estado final de este.

setValoresPerfilToComponente(Integer,Short): elimina los ComponenteProcedimiento y los ComponenteIndicador del componente cuyo id es pasado como parámetro y crea los ComponenteProcedimiento y los ComponenteIndicador a partir del id del perfil.

activarNuevoComponenteServicio(Short, Long) : activa un componente servicio por primera vez estableciéndole los valores del perfil pasado como parámetro.

EndPointsCto obtenerEndpoints(Long idServicio): devuelve un objeto *EndPointsCto* que contiene los conectores de los consumidores y productores del servicio cuyo id es pasado como parámetro.

obtenerIdsMonitorizar():List<Integer>: devuelve los ids de los componentes servicio que están activos en monitorización y en el Catálogo de Productos de la Plataforma de Interoperabilidad.

ConfiguraciónMonitorizaciónService

estaPerfilAsignado(Short idPerfil): boolean: comprueba si el perfil cuyo id es pasado como parámetro está asignado a un componente o no.

obtenerPerfilesMonitorizacion():List<PerfilMonitorizacionCto>: devuelve una lista con los perfiles de monitorización.

obtenerFechaInicio():Date: devuelve la fecha de inicio del análisis proactivo.

ReglaIndicadorService

countAll(): Long: cuenta todas las reglas que existen en el Módulo de Monitorización.

7.4.3.4. Otros servicios

Además de los servicios de maestros y los servicios de entidades se crearán otros dos servicios, el servicio de Monitorización y el servicio de Consulta de prueba.

El objetivo del servicio de Monitorización es encapsular las funcionalidades utilizadas para monitorizar un componente y para aplicarle el análisis proactivo. De este modo los batches de las capas superiores cuyo objetivo es ejecutar de manera programada la monitorización y el análisis proactivo solo tendrán que invocar a estos dos métodos pasándole el id del componente a tratar. Además el servicio de Monitorización también ofrece una funcionalidad para obtener un resumen de los datos de Monitorización.

La interfaz y la implementación del servicio de Monitorización son:

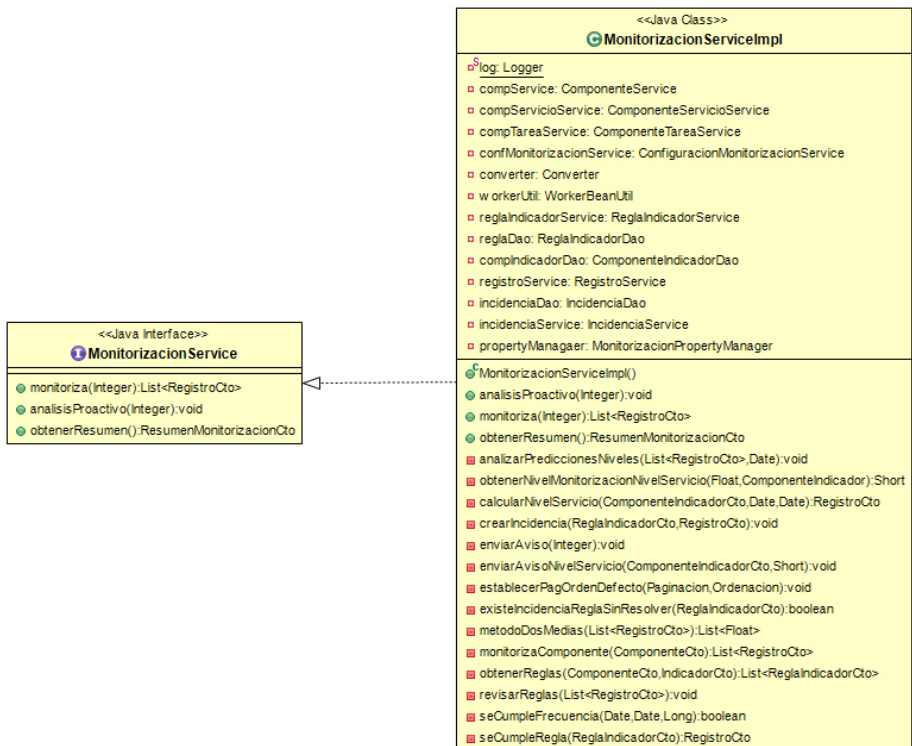


Ilustración 28. Diagrama servicio de Monitorización

En el apartado de explicación de la capa de batchs se analizará más en detalle a través de un diagrama de secuencia la implementación de las funcionalidades de monitorización y análisis proactivo.

Por otro lado se creará también un servicio que englobará la funcionalidad relacionada con realizar una consulta de prueba a un servicio de la Plataforma de Interoperabilidad. La funcionalidad de este servicio podría crearse dentro del servicio de ComponenteServicio, el motivo por el que no se hará es porque se pretende que en un futuro pase a formar parte de un módulo distinto de la Plataforma de Interoperabilidad que se encargue de todo lo relacionado con consultas de prueba.

La funcionalidad que ofrece el servicio de Consulta de Prueba es enviar una petición de prueba a un servicio a partir de la entidad ComponenteServicio y devolver su respuesta.

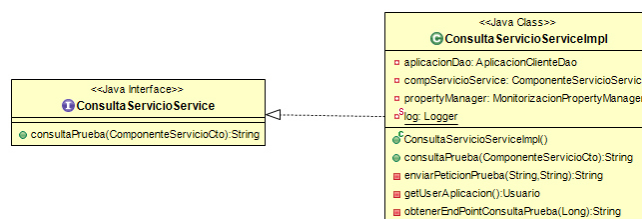


Ilustración 29. Diagrama servicio Consulta de prueba

7.4.4. Procedimientos

Hasta el momento se ha dicho en numerosos apartados que para la monitorización de la Plataforma de Interoperabilidad se utilizarán una serie de Procedimientos. En este apartado se describirá como se construirán estos procedimientos para permitir que se añadan nuevos sin necesidad de tener que modificar ningún código del Módulo de monitorización.

Lo primero que debemos establecer a la hora de hablar de un procedimiento es que este debe calcular al menos un indicador de un componente. El indicador será un parámetro del componente que aporte información sobre el estado de este y que sea medible.

Uno de los procedimientos que se crearán en esta primera versión del Módulo de Monitorización será el Ping de Servicio. Este procedimiento calculará el indicador de Estado de servicio que podrá ser activo o no activo.

Un procedimiento será una función que reciba un id del componente a monitorizar y que devuelva una lista de Registros con los valores de los indicadores que calcula. Por lo tanto el procedimiento de Ping de Servicio recibirá el id de un componente de tipo servicio, a partir de este id hará un ping al servicio y creará un registro donde guarde un 0 si el servicio estaba activo y un 1 si el servicio no estaba activo. Este registro tendrá el id del componente, el id del registro y la fecha.

Como el procedimiento no depende de un componente servicio en concreto se podrá aplicar a varios componentes servicio. La única diferencia que existirá en aplicar el procedimiento a un componente o a otro será que el Registro creado tendrá un componente distinto.

Una vez hemos visto que podemos aplicar un mismo procedimiento a varios componentes distintos, veremos cómo aplicar varios procedimientos a un solo componente.

7.4.4.1. Patrón comando

Uno de los principales objetivos que se busca en el Módulo de Monitorización es conocer el estado de sus componentes. Para ello a cada componente se le aplicarán unos procedimientos que calcularán unos indicadores.

El Módulo de Monitorización arrancará en su primera versión con tres procedimientos de monitorización pero se espera que el número de estos aumente con el fin de poder contar con mayor cantidad de indicadores para controlar el estado de los componentes.

En vistas al aumento del número de procedimientos se pretende diseñar una solución que permita la integración de los nuevos procedimientos sin tener que modificar la implementación del Módulo de Monitorización. Para cumplir este propósito se hará uso del patrón Comando.

El patrón Comando permite definir un interfaz común de manera que se puedan invocar las acciones de manera uniforme y extender el sistema de manera sencilla con nuevas acciones.

Lo que se pretende es que la invocación al método de ejecutar un procedimiento se realice indistintamente del procedimiento que sea. Para ello, se creará un interfaz *Worker* que definirá el método que todos los procedimientos deben implementar y que será el invocado para ejecutar cualquier procedimiento.

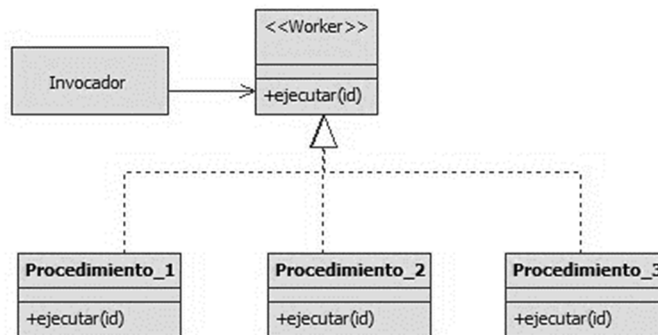


Ilustración 30. Patrón Comando

Con este patrón el invocador puede recuperar siempre un procedimiento como un Worker y llamar al método *ejecutar(id)* independientemente del procedimiento que sea.

A continuación se muestra cual es el diagrama de procedimientos del Módulo de Monitorización y se explica cómo se utilizará Spring para recuperar los distintos procedimientos.

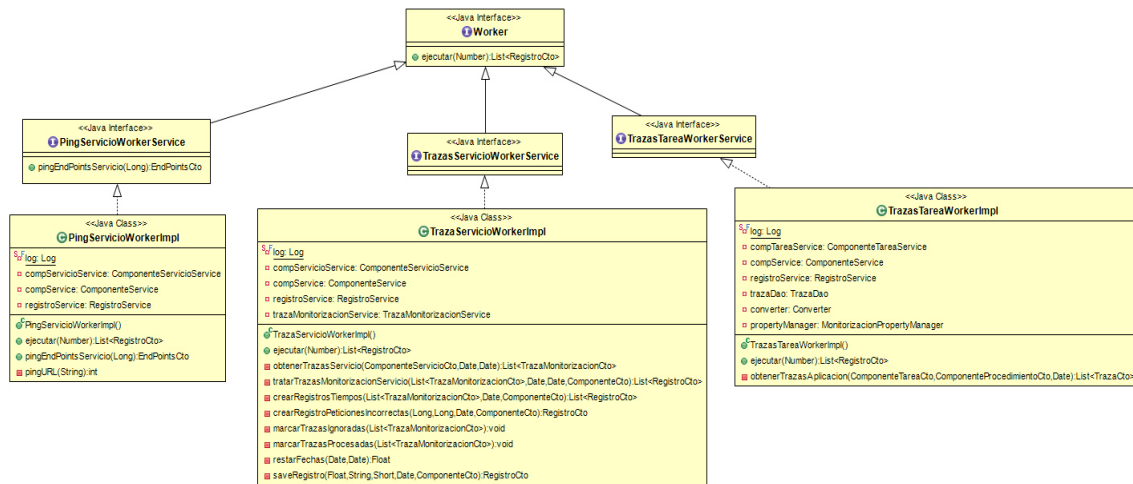


Ilustración 31. Diagrama de clases Procedimientos

Como se puede ver todos los procedimientos heredan de Worker y por lo tanto tienen el método *ejecutar()*. Además se ha introducido una interfaz más entre Worker y la implementación del procedimiento. Esto se utilizará para aquellos casos en los que se quiera reutilizar la implementación del procedimiento para otras funcionalidades. Este es el caso del ping de servicio que se va a utilizar desde la Consola de monitorización y que por eso se publica en la interfaz PingServicio.

A nivel de implementación todos los procedimientos serán ejecutados a partir de su código. Si se quiere introducir un nuevo procedimiento bastará con crearlo en base de datos con el código asignado en la implementación. Además también habrá que crear en base de datos los indicadores del nuevo procedimiento con el fin de que los registros que cree los enlace con el indicador correspondiente.

Una vez esté el procedimiento en base de datos junto con sus indicadores se podrá asignar a un componente y la siguiente vez que este se monitorice se le aplicará el nuevo procedimiento, recuperándolo a partir de su código.

Ejemplo detallado invocación procedimiento

Todas las implementaciones de los procedimientos tendrán la anotación de Spring `@Componente("codigoProcedimiento")`. Al establecer esta anotación se forzará que se cree una instancia de cada procedimiento en el contenedor de beans de Spring. Además al asignarle un código se podrán recuperar estas instancias utilizando el código.

Como ya se ha visto en el modelo de datos, cada componente tiene una serie de procedimientos y cada procedimiento tiene un código que hace referencia al código que se establece en la anotación de Spring. A continuación se muestra como a partir de un componente se ejecutarían sus procedimientos:

```

Componente comp= this.componenteServicioService.get(idComponente);
ApplicationContext context = MyContext.getApplicationContext();
for(ComponenteProcedimiento compProc : comp.getComponentesProcedimiento){
    Worker worker = (Worker) context.getBean(compProc.getProcedimiento().getCodigo());
    List<Registro> registros= worker.ejecutar(comp.getId());
}
  
```

Como podemos ver la ejecución de los procedimientos para monitorizar el componente no depende de la implementación. La única dependencia que existe para que a un componente se le ejecute un procedimiento es que los tenga asignado.

7.5. Batches

En este apartado se detallarán los batches utilizados en el Módulo de Monitorización, ya que constituyen una de las piezas más importantes de la monitorización. Gracias a estos procesos programados se permite automatizar la monitorización de los distintos componentes de la Plataforma de Interoperabilidad.

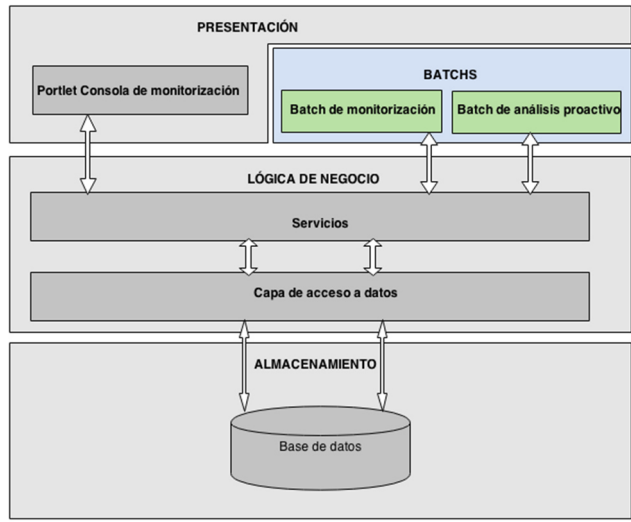


Ilustración 32. Capa de Batches

Para la implementación de los dos procesos batches de los que constará el Módulo de Monitorización se utilizará el framework Spring Batch, del ya explicado framework Spring.

El elemento principal de Spring Batch es el Job. Un Job está compuesto por varios Steps, que se ejecutan de forma secuencial. Todos los Jobs de Spring Batch deben implementar el interfaz JobLauncher. Esta interfaz presenta el método *run*, que recibe un Job y unos parámetros, y ejecutará el Job.

Spring Batch lleva un registro de todas las ejecuciones y parámetros con los que se lanzaron los Jobs. El encargado de guardar estos registros es el JobRepository, que cuenta con una implementación para almacenar esta información en base de datos o memoria.

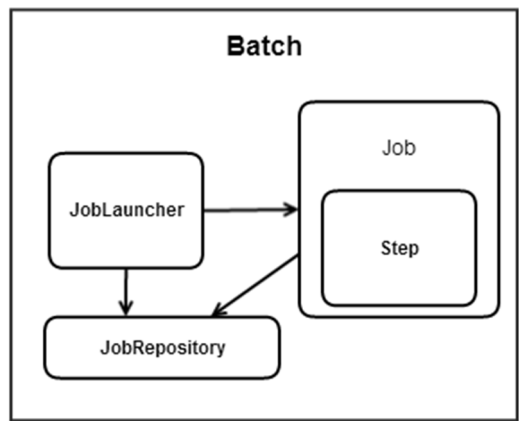


Ilustración 33. Estructura Spring Batch

Para los batchs del Módulo de Monitorización se crearán dos Jobs de un solo Step cada uno. Los Jobs heredarán de SimpleJob que es la implementación más simple de un Job y los Steps heredarán de Tasklet que también es la implementación más simple de un Step.

El Step de cada Job será el encargado de llamar a la Capa de servicios del Módulo de Monitorización para cumplir con los objetivos del batch. Para realizar las llamadas a los servicios de monitorización la clase Step utilizará la inyección de dependencia que ya ha sido explicada en el apartado 7.4.1. *Spring Framework*.

La estructura de cada batch del Módulo de Monitorización será:

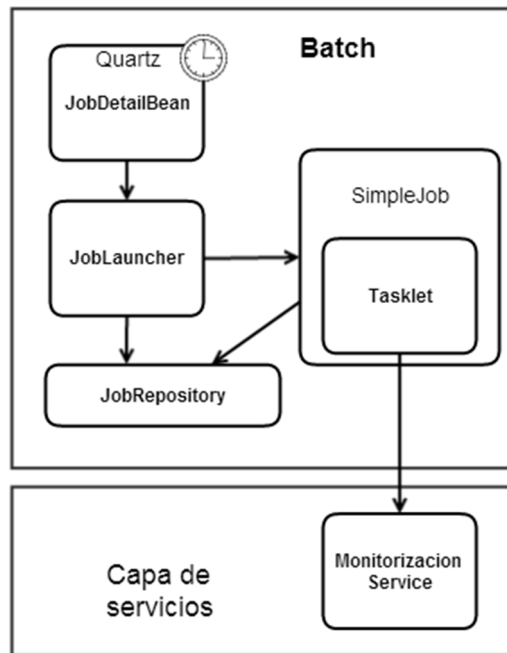


Ilustración 34. Batch del Módulo de Monitorización

Como se puede ver se ha añadido un componente más, el objeto de Quartz JobDetailBean. Quartz es un framework para la planificación y gestión de tareas. Mediante el uso de Quartz se configurará la frecuencia con la que se ejecutará cada batch.

Para establecer la frecuencia de ejecución se utilizará una expresión cron. Una expresión cron es una cadena de texto compuesta por 6 o 7 campos separados por un espacio en blanco que se utiliza para representar instantes o periodos de tiempo.

En los momentos marcados por la expresión cron se utilizará el objeto JobDetailBean para lanzar el Job que ejecutará el Step que cumple con la misión del batch. El objeto JobDetailBean es el encargado de contener la información necesaria para saber que Job de Spring Batch utilizar.

Después de haber visto cómo se crearán los batchs pasaremos a definir la secuencia de pasos que desencadena cada batch para cumplir con su misión. Se mostrarán primero dos diagramas de secuencia, una por cada batch del Módulo de Monitorización y luego se explicará cada uno de estos.

Como ya hemos introducido en apartados anteriores, el objetivo del batch de Monitorización es ejecutar los diversos procedimientos que tiene cada componente para calcular sus indicadores. Además una vez se hayan calculado los indicadores se deben generar los registros y revisar las reglas correspondientes.

Cuando se lanza el batch se recuperan los ids de los componentes de tipo servicio y de tipo tarea que se deban de monitorizar. Estos serán aquellos que estén activos tanto en el Módulo de Monitorización como en el Catálogo de productos.

El siguiente paso es llamar a la función `monitorizar(id)` de `MonitorizacionService` por cada id recuperado. La función `monitorizar` será la encargada de recuperar el componente a monitorizar a partir del id recibido.

Con el componente recuperado se recorrerán todos los procedimientos que tenga asignados. Por cada procedimiento se comprobará cual fue la última vez que fue ejecutado para ese componente y se comparará con la fecha actual. Si la diferencia entre la fecha actual y la fecha de última ejecución es igual o mayor que la frecuencia de ejecución del procedimiento para el componente se ejecutará el procedimiento. Al ejecutarse un procedimiento se actualizará la fecha de última ejecución para ese componente.

Para ejecutar un procedimiento se recuperará del contexto de Spring el bean que lo implementa a partir del código de procedimiento. Como ya hemos visto en el apartado *7.4.4. Procedimientos* los procedimientos se recuperan a través de la interfaz `Worker` y se ejecutan llamando al método `ejecutar(id)` que es común a todos.

Al ejecutar un método este calculará los indicadores para el componente y crearán los registros con los valores de estos. Al terminar la ejecución del procedimiento este devolverá la lista de registros creados.

Una vez hayan sido ejecutados todos los procedimientos del componente se contará con una lista con todos los registros creados.

Por cada registro creado se comprobará si se cumple alguna regla de monitorización. En caso de cumplirse se comprobará si ya existe una incidencia sin resolver debido a esa reglas, si no existe se creará una incidencia y se enviará un email. El destinatario y el mensaje del email se obtendrán a partir de la información que contiene la regla.

Después de revisar todas las reglas finalizará la ejecución del batch que no se volverá a ejecutar hasta que se vuelva a cumplir el siguiente periodo marcado por la expresión cron.

7.5.2. Batch Análisis Proactivo

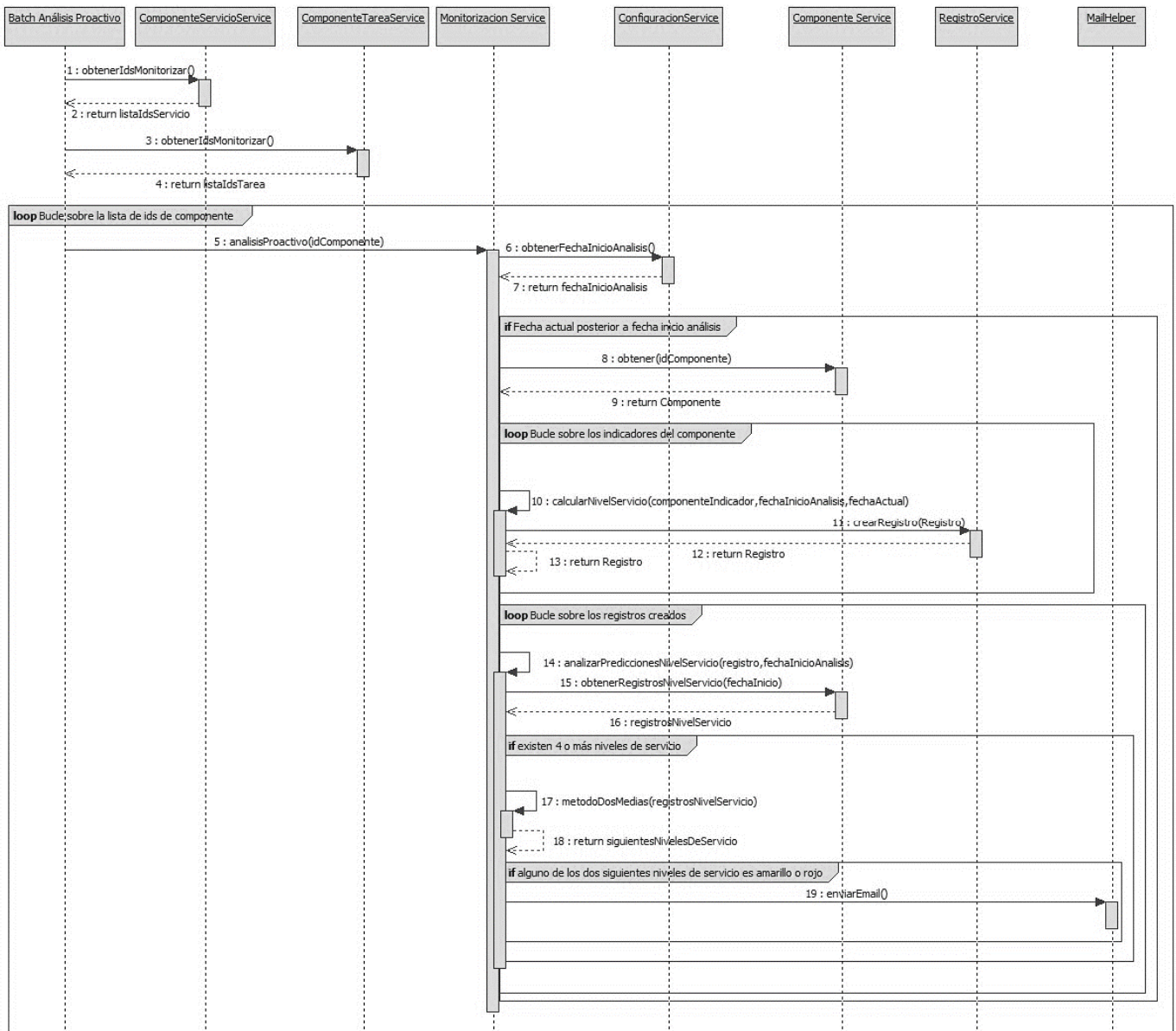


Ilustración 36. Diagrama secuencia Análisis Proactivo

La función del batch de Análisis proactivo será calcular los niveles de servicio de los distintos indicadores y predecir cuáles serán los siguientes con el fin de detectar degradaciones de servicio. Los niveles de servicio son el resumen de los valores de los indicadores en un rango de tiempo. Este tiempo vendrá marcado por el período de ejecución del batch, que en un principio se establecerá cada 7 días.

El batch de Análisis Proactivo comienza recuperando los ids de los componentes activos en la Monitorización y en el Catálogo de Productos. Una vez recuperada la lista de ids a analizar, se recorre esta lista llamando al método de *analisisProactivo(id)* que ofrece MonitorizaciónService.

La primera acción que realizará la función *analisisProactivo(id)* es recuperar la fecha de inicio del análisis establecida en la configuración. Si esta fecha es igual o posterior a la actual realizará el análisis proactivo, de no ser así la función terminará.

Si se cumple la fecha, se recupera el componente al que se le quieren calcular los niveles de servicio. Para ello se llama al método *obtener(id)* de *ComponenteService*. Por cada uno de los indicadores que tenga el componente se llamará a la función *calcularNivelesServicio(componenteIndicador, fechaInicioAnalisis, fechaActual)*.

La función *calcularNivelesServicio* recupera los últimos registros del indicador para el componente que se está analizando. Estos registros son los registros de tipo indicador generados por el batch de Monitorización. Solo se recuperan los registros desde la fecha del último registro que se utilizó para calcular el anterior nivel de servicio de ese mismo componente e indicador. Si nunca se calculó un nivel de servicio, la fecha desde la que se recuperan los registros es la de inicio del análisis proactivo. Además si la fecha del último nivel de servicio calculado es anterior a la fecha de inicio del análisis proactivo se utilizará esta última.

Una vez se recuperan los registros se calcula la media del valor de estos y se guarda un registro de tipo nivel de servicio con el valor de la media. Además se utiliza la fecha del último registro utilizado para hacer la media, como fecha de último nivel de servicio.. Esta fecha será utilizada la siguiente vez que se vuelva a ejecutar el análisis de nivel de servicio para el mismo componente e indicador. El motivo por el que se guarda la fecha es para no calcular los niveles de servicio con registros que ya fueron utilizados para calcular niveles de servicio anteriores.

Después de crear el registro con el nivel de servicio la función *calcularNivelesServicio* devuelve el registro creado. Este registro es añadido a una lista con el resto de los registros generados para los niveles de servicio del componente analizado.

Al finalizar de calcular todos los niveles de servicio de los indicadores del componente analizado se procede a predecir cuáles serán los dos siguientes niveles de servicio para cada uno de los niveles de servicio que se acaban de calcular.

Se recorre la lista con los registros de los niveles de servicio recién creados. Para cada registro se recuperan los anteriores registros de tipo nivel de servicio que hagan referencia al mismo componente e indicador. Se recuperan solo los registros desde la fecha de inicio del análisis proactivos.

En caso de que el número de registros de tipo nivel servicio sea igual o mayor a 4 se llamará al Método de las Dos Medias pasándole los valores de todos estos registros. El Método de las Dos Medias devolverá cuales se predicen que serán los siguientes niveles de servicio. Al final de este apartado se explicará el funcionamiento del Método de las Dos Medias.

Por último, con los valores de los dos siguientes niveles de servicio se calculará, en función de los umbrales del indicador, en qué nivel de monitorización se encuentra cada nivel de servicio. Si alguno de los siguientes niveles de servicio predichos se encuentra en un nivel de monitorización distinto del Verde se enviará un aviso alertando de la degradación del servicio.

7.5.2.1. Método de las Dos Medias

El Método de las Dos Medias permitirá predecir cuáles serán los siguientes niveles de servicio en función de la tendencia de los niveles de servicio con los que se cuenta.

El Método de las Dos Medias se basa en dividir la muestra de valores de nivel de servicio que recibe en dos mitades, y calcular la media de los datos de cada mitad. A partir de este momento se cuenta con dos puntos, uno que representa la media de la primera mitad de la muestra y otro que representa la media de la segunda mitad de la muestra.

Los siguientes niveles de servicio se calcularán a partir de la función que representa la recta que pasa por las dos medias.

A continuación se muestra un ejemplo para explicar cómo funciona el Método de las Dos Medias. Se trata de predecir los dos siguientes niveles de servicio del indicador Tiempo medio de petición. En este ejemplo el componente es indiferente ya que solo influiría para conocer los umbrales y así poder decir en qué nivel de monitorización (verde, amarillo, rojo) se encuentra cada nivel de servicio predicho:

Los últimos 4 registros de tipo nivel servicio para el indicador Tiempo medio de petición fueron:

Número Nivel Servicio	Valor Nivel Servicio
1	2 segundos
2	5 segundos
3	7 segundos
4	8 segundos

Tabla 24. Niveles de Servicio ejemplo Método dos Medias

Siguiendo el Método de las Dos Medias se divide la muestra en dos mitades y se calcula la media de la primera y la de la segunda.

Los valores que se obtienen son 3.5 segundos y 7.5 segundos respectivamente.

El siguiente paso es calcular la fórmula de la recta que pasa por 3.5 segundo y 7.5 segundos. La función de la recta que pasa por los dos puntos será igual a:

$$f(\text{numNS}) = \frac{\text{mediaNS}_2 - \text{mediaNS}_1}{\text{mediaNumNS}_2 - \text{mediaNumNS}_1} * (\text{numNS} - \text{mediaNumNS}_1) + \text{mediaNS}_1$$

$$f(\text{numNS}) = \frac{7.5 - 3.5}{3.5 - 1.5} * (\text{numNS} - 1.5) + 3.5$$

Sustituyendo el número del siguiente nivel de servicio que será el 5 obtenemos que el valor del nivel de servicio será 10.5. Si sustituimos por el siguiente del siguiente, es decir, por 6, obtenemos que el nivel de servicio será de 12.5.

La predicción del Método de las Dos Medias obtendría para el ejemplo dado que el siguiente nivel de servicio para el indicador de Tiempo medio de petición sería de 10.5 segundos y que el siguiente del siguiente sería de 12.5 segundos.

Si ahora contásemos con un componente concreto que tuviese por ejemplo el umbral medio alto establecido en 20 segundos y el umbral bajo medio en 10 segundos podríamos decir que los dos siguientes niveles predichos se encontrarán en el nivel de monitorización amarillo.

A continuación se muestra una gráfica que representa el ejemplo resuelto.

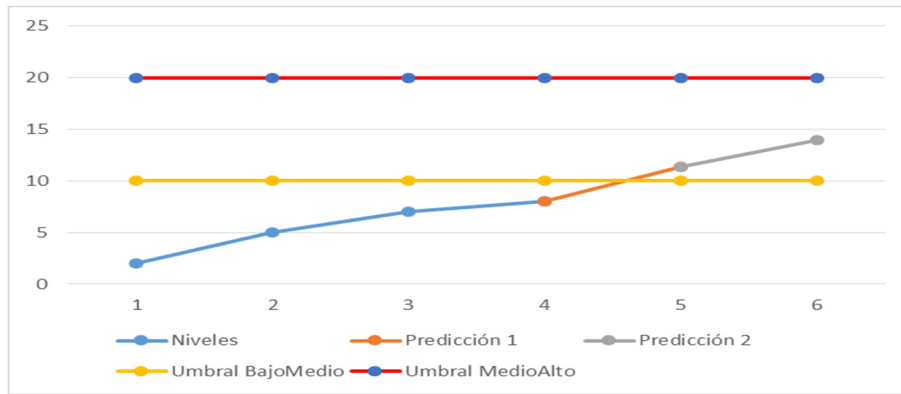


Ilustración 37. Método de las Dos Medias

7.6. Capa de presentación

Para terminar con el apartado de diseño se presentará la capa de presentación. La misión de esta capa es ofrecer una Consola de administración a través de la cual un usuario pueda interactuar con el Módulo de Monitorización.

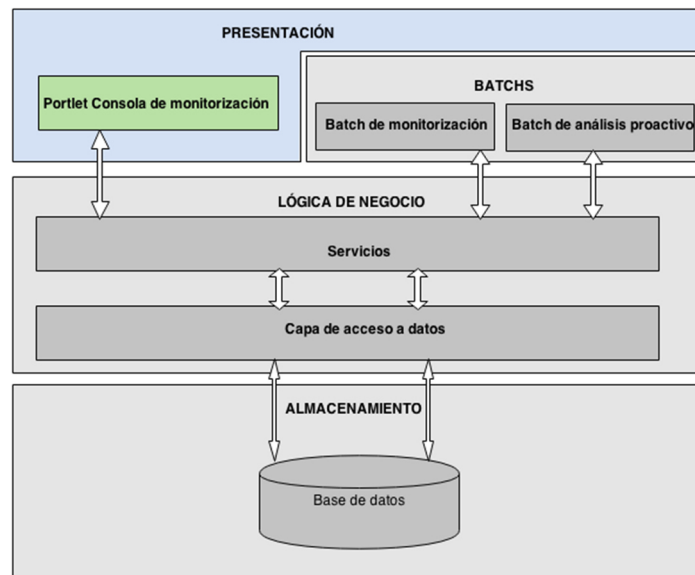


Ilustración 38. Capa de Presentación

La única responsabilidad de la capa de presentación serán las vistas de la Consola de monitorización y coordinar la relación entre las acciones que el usuario hace en las vistas con la lógica de negocio.

7.6.1. Patrón Modelo Vista Controlador

La capa de presentación sigue un diseño multicapa al igual que todo el Módulo de Monitorización. El diseño realizado, también se puede identificar con el patrón Modelo Vista Controlador.

MVC es un patrón de presentación utilizado ampliamente en las aplicaciones Web. Divide la interacción con la interfaz de usuario en 3 roles distintos: el Modelo, la Vista y el Controlador.

El Modelo es la parte de la aplicación que representa información acerca del dominio del sistema. En el caso del Módulo de Monitorización, el Modelo se corresponde con las capas de servicios y acceso a datos. Contiene toda la información y el comportamiento de la aplicación.

La Vista se corresponde con la representación del Modelo en la interfaz de usuario. Sólo se encarga de mostrar la información, no de gestionar los cambios de dicha información, ya que esa es precisamente la tarea del tercer rol del patrón, el Controlador.

El Controlador se encarga de recibir las acciones del usuario, llamar al Modelo y hacer que la Vista se actualice de forma apropiada.

De lo explicado anteriormente se deduce que la capa de presentación será una combinación de la Vista y el Controlador.

7.6.2. Portlet de Consulta de monitorización

Como ya hemos introducido al inicio de esta memoria, el Módulo de Monitorización contará con una Consola de monitorización dentro de la zona privada del Portal de Interoperabilidad.

El Portal de Interoperabilidad está construido utilizando el gestor de contenidos Liferay. Liferay permite la creación de una página web a través de la creación de páginas a las cuáles se les puede añadir contenidos.

Uno de los contenidos que se puede añadir a una página de Liferay son los portlet. Un portlet es un módulo independiente que permite el desarrollo de una aplicación y que después se puede posicionar en cualquier página de Liferay.

El Portal de Interoperabilidad cuenta con una página por cada aplicación. Cada página está constituida por un solo portlet. Para exponer la Consola de monitorización en el Portal de Interoperabilidad lo que se hará será crear un portlet que será desplegado en una nueva página de Liferay llamada Monitorización.

El portlet de Consulta de monitorización será el que asuma las responsabilidades de Vista y Controlador.

7.6.3. Spring MVC Portlet

Para la construcción del portlet se utilizará el framework Spring MVC Portlet. Este framework facilitará la implementación del patrón MVC.

Spring MVC permite la creación de controladores y handlers de las llamadas desde la vista de una forma sencilla mediante anotaciones en las clases java.

Para el portlet de Consulta de monitorización se creará un único controlador que recibirá todas las peticiones de las vistas. Al tener como única puerta de entrada este controlador, la clase que lo representa puede crecer indiscriminadamente. Para evitar esto, y además, facilitar la reutilización de código se crearán una serie de controladores específicos.

El controlador de entrada simplemente tendrá la funcionalidad de redirigir las peticiones hacia los controladores específicos oportunos, que serán los encargados de tratar con los servicios para resolver las peticiones.

La estructura de controladores específicos se organizará a nivel de entidades, al igual que se hizo en los servicios. Este será un punto clave a la hora de reutilizar código y facilitar el mantenimiento de la aplicación, ya que, todas aquellas operaciones comunes a las entidades serán realizadas por unos controladores generales de los que heredarán los específicos.

La opción de crear estos controladores generales es posible gracias al diseño de los servicios, que hace que todas las entidades compartan un determinado número de operaciones. A continuación se muestra un ejemplo para poder entender este concepto.

Todas las entidades tienen un servicio que se llama `obtenerCtos` que recupera de base de datos todos los datos de cada entidad. Si queremos hacer una función en un controlador que recupere todos los registros, lo que tendríamos que hacer sería llamar a la función `obtenerCtos` del servicio de la entidad Registro. Si ahora queremos hacer una función en un controlador que recupere todas las incidencias tendríamos que hacer lo mismo pero con el servicio de la entidad Incidencia.

Con esto, lo que se deduce es que, lo único diferente es el servicio, la llamada es la misma. La forma de crear una única función que sirva para cualquier tipo de entidad es, crear una clase genérica que implemente la función de `obtenerTodos`, que llame a un método abstracto de la propia clase para obtener el servicio que deba utilizar en cada caso. Todas las clases que hereden de esta tendrán la función de `obtenerTodos` ya implementada.

La implementación de esta funcionalidad sería: `this.getService.obtenerCtos()`; Siendo `getService` el método abstracto que deberá implementar cada clase que herede de la clase genérica.

Al igual que en los servicios teníamos distintos niveles de los que podría heredar una entidad, dependiendo de las funcionalidades que se quisieran ofrecer, en los controladores genéricos pasa lo mismo. A continuación se muestra el diagrama de clases de los controladores genéricos de los que podrán heredar los controladores específicos que ayudarán al controlador principal a resolver las peticiones.

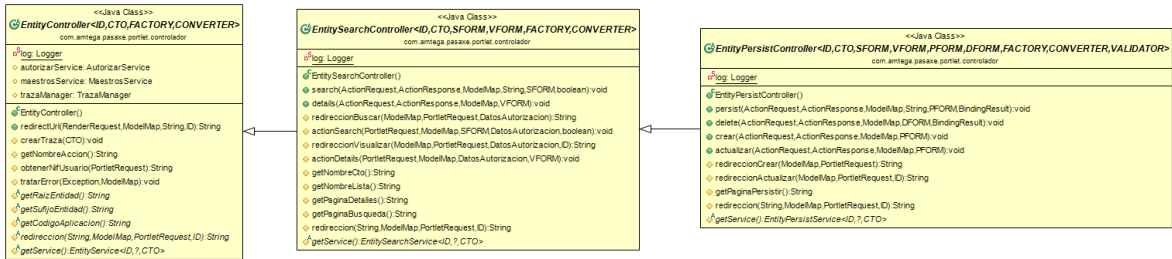


Ilustración 39. Diagrama de clases controladores genéricos

Como se puede ver, todos los métodos abstractos que deberán implementar las clases que hereden de algún elemento de esta jerarquía aportan información concreta de cada entidad. Ejemplos de estos métodos son `getRaizEntidad` y `getSufijoEntidad`, el primero devuelve la ruta de la carpeta donde se guardan las vistas de la entidad y el segundo el nombre final que se le establece a las vistas para identificarlas.

Diseño detallado de un controlador en Spring MVC

La creación de un controlador en Spring MVC se realizará utilizando las anotaciones `@Controller` y `@RequestMapping("VIEW")`. Con la anotación `@Controller` estamos definiendo que es un controlador manejado por Spring y como tal se creará un bean de este al iniciar el contexto de Spring. La anotación `@RequestMapping("VIEW")` es utilizada para indicar que todas las peticiones de la vista deben ser redirigidas a este controlador.

Un controlador está constituido por métodos de dos tipos, los que devuelven un string y los que no devuelven nada. Los primeros son los métodos Render donde el string que devuelven se corresponde con el nombre de la vista a mostrar en el portlet. Los métodos que no devuelven nada son los métodos Action que reciben las peticiones de los usuarios, ejecutan la lógica asociada y establecen cual va a ser el método Render a ser el siguiente en ejecutar, en caso de no establecer ninguno, se ejecutará el por defecto.

Para definir si un método es de un tipo u otro se utilizarán las anotaciones `@Render` y `@Action`. Estas anotaciones pueden llevar parámetros con el fin de que las vistas puedan lanzar peticiones hacia un método concreto.

A continuación se muestra un diagrama en el que se puede observar los métodos que tiene el controlador del portlet de Consulta de monitorización para tratar las peticiones que se hagan sobre la entidad ConfiguraciónMonitorizacion

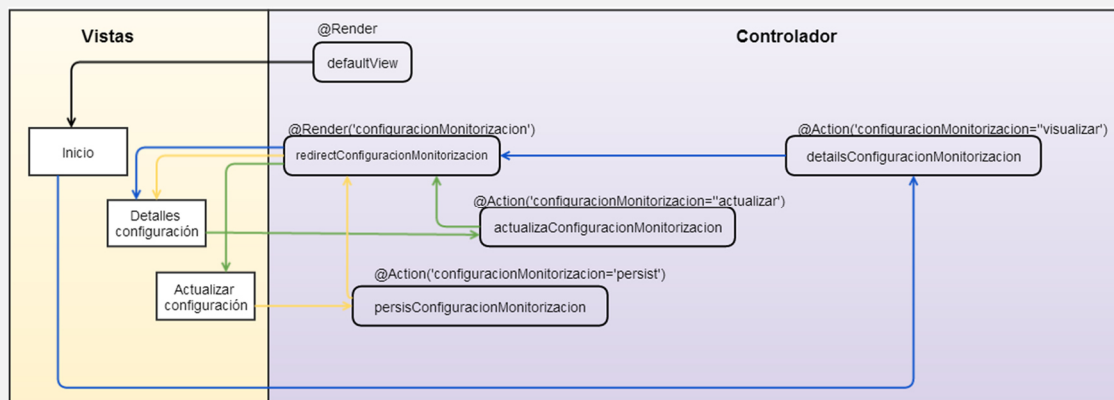


Ilustración 40. Controlador portlet

Como se puede ver el controlador tiene un método por defecto que carga la vista inicial. En la vista inicial existirá un menú a través del cual se podrá acceder a ver la Configuración de la monitorización. Para ello, el botón del menú que lleva a tal vista, lanzará una petición de tipo Action con los parámetros `configuracionMonitorizar=visualizar`.

Esta petición será recogida por el método etiquetado con la anotación correspondiente que cargará los datos de la configuración llamando al servicio correspondiente, los guardará en la request y llamará al método etiquetado con `@Render('configuracionMonitorizacion')`. El método `@Render` de la configuración será el encargado de devolver la vista que se desee mostrar para cada una de las peticiones que hagan los métodos Action de configuración. Con esto lo que se pretende es centralizar en un solo método render la devolución de las vistas de una misma entidad.

Para pasar datos desde la vista al controlador se utiliza un objeto de tipo `ModelMap`. Este objeto es un mapa en el que se pueden guardar otros objetos con un nombre asociado. Utilizando este nombre se pueden recuperar los objetos introducidos en el objeto `ModelMap` a partir de su nombre.

Una vez hemos hablado de cómo construir el controlador, la otra parte que queda es la de las vistas. Estas serán creadas utilizando `jsp`, `Javascript` y `css`. La única restricción acerca de las vistas es que deben seguir la apariencia de la Portal de la Plataforma de Interoperabilidad. Para ello, se dispone de las clases `css` que son utilizadas en el Portal de la Plataforma de Interoperabilidad.

7.6.4. Control de acceso

Una de los requisitos de la Consola de monitorización es no permitir el acceso a aquellos usuarios no autorizados.

La Consola de monitorización es una aplicación de la Plataforma de Interoperabilidad. Esta aplicación se divide en seis partes aplicación. Dividir la aplicación en partes aplicación permite poder controlar el acceso de una forma más específica. Estas partes aplicación son *General, Configuración, Incidencias, Registro, Reglas y Componentes servicio*.

Cada parte aplicación tendrá una serie de operaciones que se puedan realizar sobre ella, estas operaciones podrán ser *consulta, alta, baja y eliminación*.

La Plataforma de Interoperabilidad permite la asignación de roles a usuarios. Cada rol tiene un rango de rol. Cada rango rol tendrá asignados unos tipos de operación para unas partes de aplicación concretas. De esta manera en función del rango rol que tenga un rol se podrá controlar que operaciones puede hacer sobre una parte aplicación un determinado rol.

Además al crear un rol se podrán asignar a nivel de aplicación las operaciones autorizadas. Las posibles operaciones que se puedan asignar a un rol sobre la aplicación estarán condicionadas por el rango rol del rol. De este modo si el rango rol no tiene la operación *consultar* para ninguna parte aplicación de la aplicación, nunca podrá dar permiso de *consultar* a la aplicación.

Utilizando este modelo, el control de acceso a la Consola de monitorización se realizará utilizando los roles de usuario. Si un usuario no tiene un rol con algún permiso para la aplicación de Consulta de monitorización no podrá acceder a esta.

Además también se utilizarán los roles para permitir realizar una operaciones u otras sobre la Consola de monitorización.

A continuación se muestra como vería la parte aplicación de Reglas un usuario con todos los permisos:

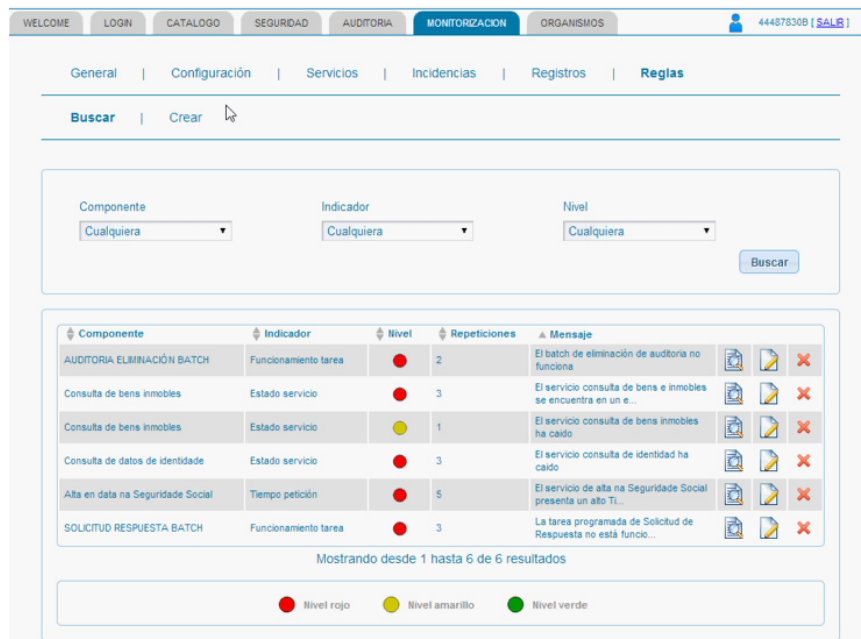


Ilustración 41. Vista Reglas con rol con todos los permisos

La siguiente es la misma parte aplicación de Reglas y el mismo usuario pero con un rol que solo tiene permiso de *consultar*:

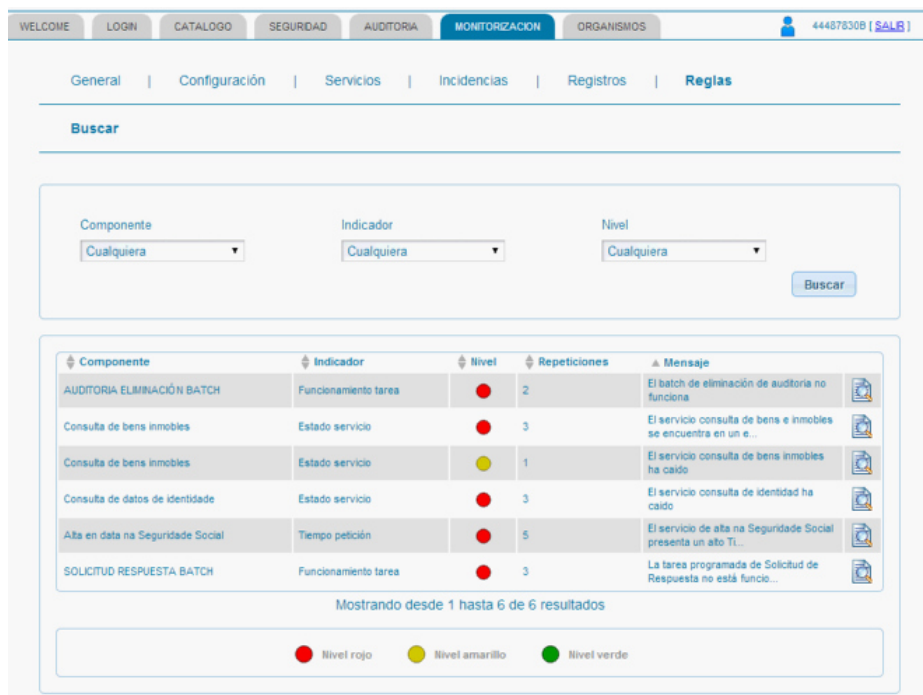


Ilustración 42. Vista Reglas con rol de consulta

Como se puede ver, al usuario que tiene el rol que solo puede consultar, no le aparecen los botones de crear, editar y eliminar.

Para gestionar este control de acceso el controlador pasará siempre a la vista un objeto con las partes aplicación del usuario y sus respectivos permisos. De este modo, en la vistas se podrán mostrar unas opciones u otras en función de las partes aplicación y los permisos.

7.6.5. Organización y diseño de las vistas

El portlet de Consulta de monitorización se dividirá en seis partes. Cada una de estas partes hace referencia a una entidad del modelo de datos del Módulo de Monitorización, a excepción de la primera que es un resumen del resto. Todas ellas son accesibles desde un menú principal que presenta el siguiente aspecto:

General | Configuración | Servicios | Incidencias | Registros | Reglas

Ilustración 43. Menú portlet Consulta monitorización

General

La primera vista será la General. Esta vista presenta un resumen del estado de la monitorización, informando de los componentes que se están monitorizando, el número de incidencias y el número de reglas activas. Además para facilitar la visión del estado de la monitorización se presenta un gráfico que resume el número de componentes con y sin alguna incidencia.

A pesar de existir una zona específica para incidencias se ha decidido mostrar en la pantalla General una tabla con las últimas incidencias. El motivo es que el usuario vea nada más entrar las incidencias.

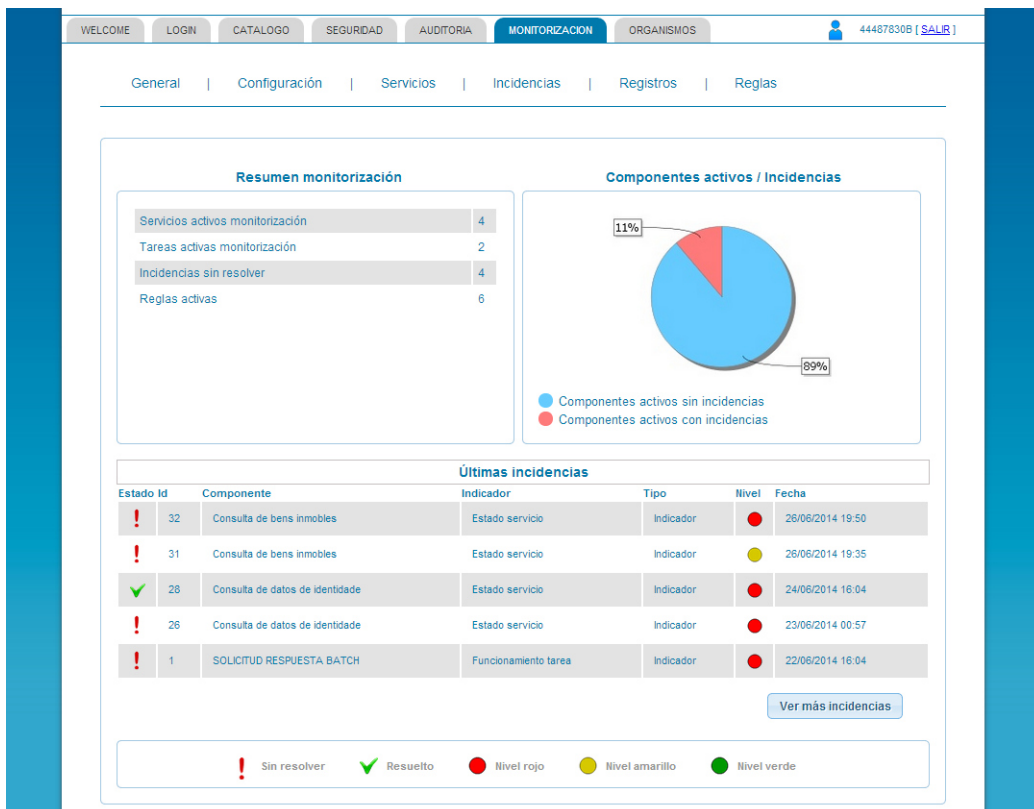


Ilustración 44. Vista General

Configuración

La siguiente parte del portlet será la de Configuración. Esta parte está formada por dos vistas, una de detalle de la configuración y otra para editar la configuración.



Ilustración 45. Vista Detalle configuración

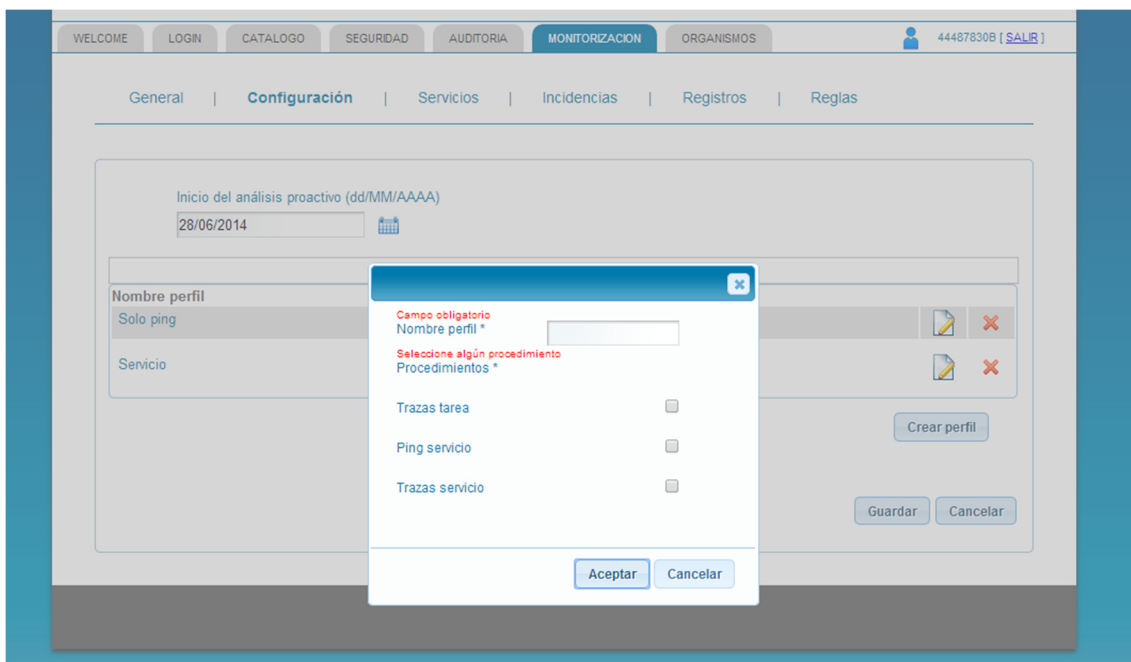


Ilustración 46. Vista Editar configuración.

Como se puede ver, en la vista de editar la Configuración se permite modificar la fecha de inicio del análisis proactivo y crear, editar y eliminar perfiles. En la captura de la vista de configuración se puede apreciar cómo se realizan validaciones sobre los datos introducidos.

Servicios

La tercera zona es la de los componentes de tipo servicio. En esta zona se contará con cuatro vistas. Estas vistas serán la de búsqueda de servicio, detalle de servicio, configuración de servicio y resultado de ping.

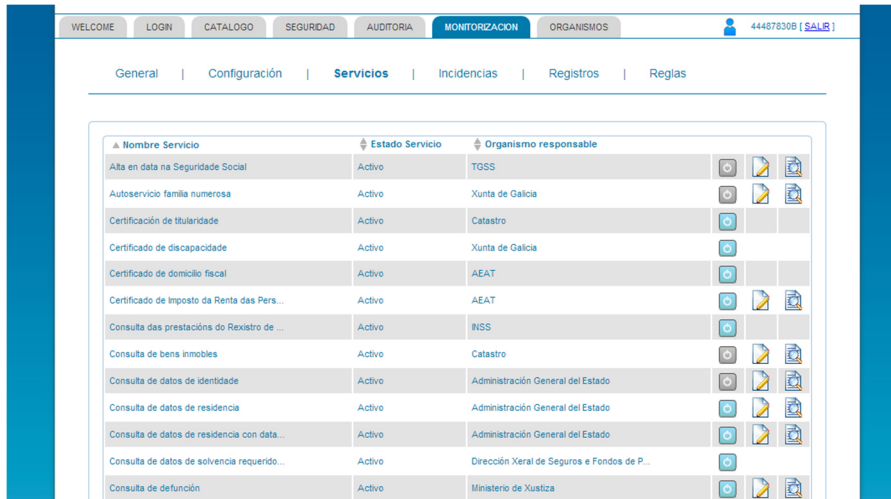


Ilustración 47. Vista Búsqueda servicio

La vista de búsqueda de servicios permite ver los servicios disponibles en el Catálogo de Productos con el estado que tienen en este. Desde esta vista se permite activar y desactivar la monitorización de cada uno de los servicios. Además también podremos acceder desde esta vista a las vistas de detalle y configuración de cada servicio.

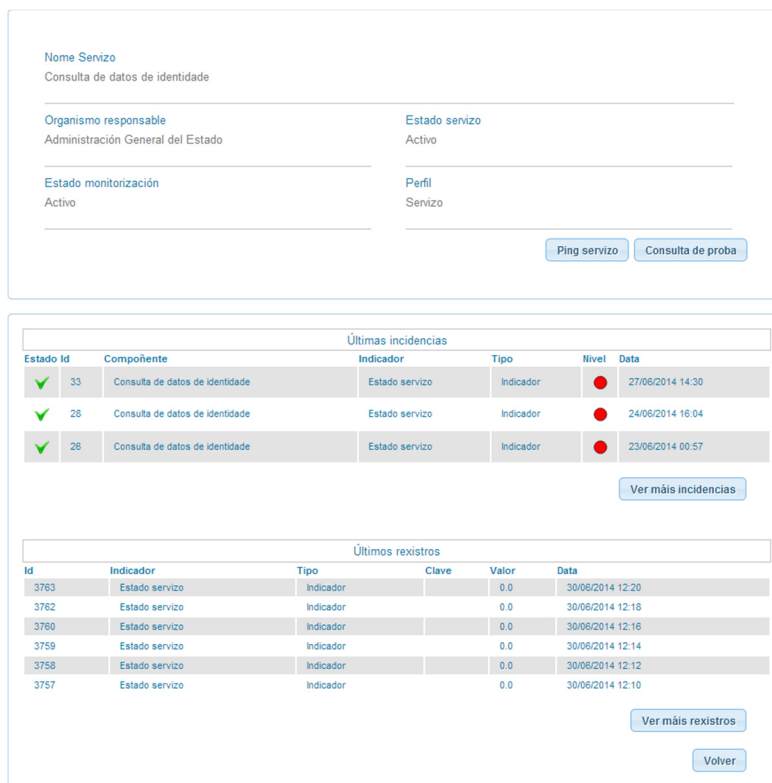


Ilustración 48. Vista de Detalle de un servicio

La vista de detalle de un servicio muestra información acerca del servicio como es su nombre, organismo responsable, estado en el Catálogo de Productos, estado de monitorización y perfil de monitorización asignado.

Esta vista también cuenta con una zona donde se muestran los últimos registros e incidencias relacionadas con el servicio. Además desde la vista de detalle se podrá hacer un ping al servicio que mostrará el estado de todos sus conectores.

Por último, si un servicio tiene configurada una plantilla de prueba, se podrá hacer una consulta de prueba desde la vista de detalle.

Procedimiento	Frecuencia
Trazas servicio	10
Ping servicio	0

Indicador	U.Bajo	U.Alto
Tiempo conector-bus	0.6	0.8
Tiempo bus ida	0.6	0.8
Estado servicio	0.6	0.8
Tiempo bus-conector	0.6	0.8
Tiempo bus vuelta	0.6	0.8
Tiempo conector ida	0.6	0.8
Tiempo petición	0.6	0.8
Tiempo conector vuelta	0.6	0.8
Tiempo servicio final	0.6	0.8
Estado petición	0.6	0.8

Ilustración 49. Vista de Configuración de un servicio

La vista de configuración de un servicio permite editar los parámetros para la monitorización del servicio. Se podrá cambiar el perfil del servicio, subir una plantilla para realizar una consulta de prueba y editar los valores de los procedimientos e indicadores que fueron asignados al servicio a través de su perfil.

De los procedimientos se podrá establecer su frecuencia (minutos) y de los indicadores se podrá modificar los umbrales.

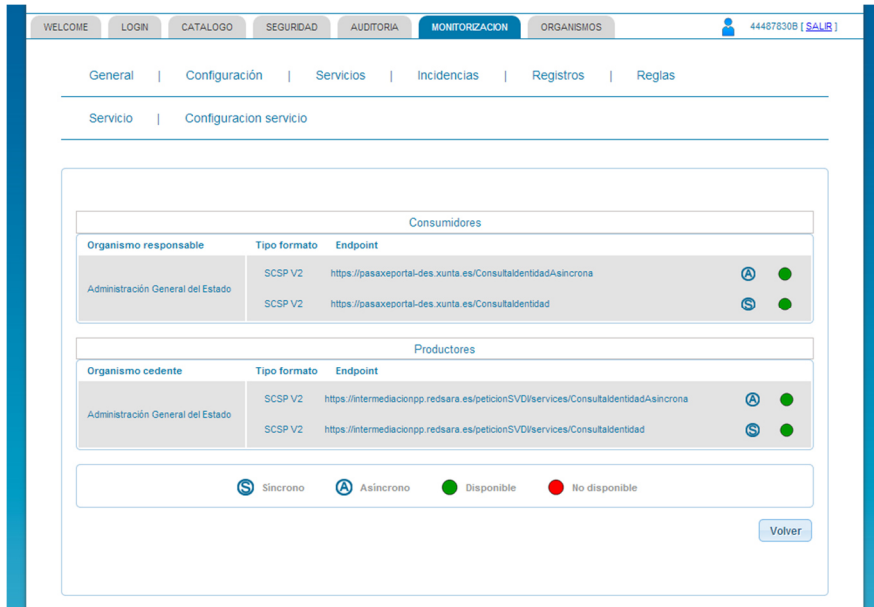


Ilustración 50. Vista Ping a servicio

La vista de ping a servicio muestra los distintos conectores de un servicio clasificados en consumidores y productores. De cada conector se informa el tipo de formato, url, tipo de comunicación y estado.

Registros

La zona de registros del portlet de Consulta de monitorización ofrece un buscador de registros. Utilizando este buscador se podrán realizar filtros sobre los registros de monitorización generados. Se podrán hacer búsqueda por id, tipo de registro, componente, indicador, fecha inicio y fechas fin.

El resultado de las búsquedas es ofrecido en una tabla en la que se muestran los registros con todos sus valores. Esta tabla podrá ser ordenada por cualquiera de sus columnas. Se establecerá un número máximo de resultados que se pueden cargar en la tabla, a partir del cual, se paginarán los resultados.

Además se permite exportar el contenido de la tabla a un fichero csv.

General | Configuración | Servicios | Incidencias | **Registros** | Reglas

Id:

Tipo:

Componente:

Indicador:

Desde:

Hasta:

Id	Componente	Indicador	Tipo	Clave	Valor	Fecha
3310	Consulta de datos de identidad	Estado servicio	Indicador		0.0	26/06/2014 19:05
3307	Consulta de datos de identidad	Estado servicio	Indicador		0.0	26/06/2014 19:00
3306	Consulta de datos de identidad	Estado servicio	Indicador		0.0	26/06/2014 18:55
3305	Consulta de datos de identidad	Estado servicio	Indicador		0.0	26/06/2014 18:50
3292	Consulta de datos de identidad	Estado petición	Indicador		0.0	26/06/2014 18:45
3293	Consulta de datos de identidad	Tiempo petición	Indicador	3	25.0	26/06/2014 18:45
3294	Consulta de datos de identidad	Tiempo conector ida	Indicador	3	2.0	26/06/2014 18:45
3295	Consulta de datos de identidad	Tiempo conector-bus	Indicador	3	0.0	26/06/2014 18:45
3296	Consulta de datos de identidad	Tiempo bus ida	Indicador	3	18.0	26/06/2014 18:45
3297	Consulta de datos de identidad	Tiempo servicio final	Indicador	3	1.0	26/06/2014 18:45
3298	Consulta de datos de identidad	Tiempo bus vuelta	Indicador	3	4.0	26/06/2014 18:45
3299	Consulta de datos de identidad	Tiempo bus-conector	Indicador	3	0.0	26/06/2014 18:45
3300	Consulta de datos de identidad	Tiempo conector vuelta	Indicador	3	0.0	26/06/2014 18:45
3291	Consulta de datos de identidad	Estado servicio	Indicador		0.0	26/06/2014 18:45

Mostrando desde 41 hasta 54 de 54 resultados

Ilustración 51. Vista Registros

Incidencias

Para mostrar la información de las incidencias se utilizarán dos vistas, una de búsqueda y otra de detalle.

WELCOME | LOGIN | CATALOGO | SEGURIDAD | AUDITORIA | **MONITORIZACION** | ORGANISMOS | 444878308 [SALIR]

General | Configuración | Servicios | **Incidencias** | Registros | Reglas

Id:

Nivel:

Tipo:

Componente:

Indicador:

Estado:

Desde:

Hasta:

Estado	Id	Componente	Indicador	Tipo	Nivel	Fecha
✓	28	Consulta de datos de identidad	Estado servicio	Indicador	●	24/06/2014 16:04
!	1	SOLICITUD RESPUESTA BATCH	Funcionamiento tarea	Indicador	●	22/06/2014 16:04
!	26	Consulta de datos de identidad	Estado servicio	Indicador	●	23/06/2014 00:57
!	31	Consulta de bens inmuebles	Estado servicio	Indicador	●	26/06/2014 19:35
!	32	Consulta de bens inmuebles	Estado servicio	Indicador	●	26/06/2014 19:50

Mostrando desde 1 hasta 5 de 5 resultados

! Sin resolver
✓ Resuelto
● Nivel rojo
● Nivel amarillo
● Nivel verde

Ilustración 52. Vista Búsqueda incidencias

La vista de búsqueda de incidencias permite filtrar las incidencias del Módulo de Monitorización. Se podrán hacer búsquedas por id, estado, componente, indicador, tipo, nivel de monitorización y fechas.

El resultado de las búsquedas es mostrado en una tabla que puede ser ordenada por columnas. Se establecerá un número máximo de resultados que se pueden cargar en la tabla, a partir del cual, se paginarán los resultados.

El contenido de la tabla podrá ser exportado a un fichero csv.

Por último, desde esta vista se podrá acceder al detalle de una incidencia.

Últimos rexistros					
Id	Componente	Indicador	Clave	Valor	Data
3716	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 15:00
3710	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 14:30
3705	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:50
3700	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:30
3695	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:15
3690	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:00

Ilustración 53. Vista Detalle incidencia

En el detalle de una incidencia se mostrará su id, componente, indicador, tipo de registro que generó la incidencia, mensaje, estado y fecha. Además se mostrará una lista con los registros relacionados con la incidencia.

Cuando se consideré que una incidencia está resuelta se podrá resolver desde esta vista.

Reglas

La última zona de la Consola de monitorización es la de reglas. Esta zona constará de 3 vistas, una de búsqueda, otra de detalle y la última para crear o editar una regla.

Xeral | Configuración | Servicios | Incidencias | Registros | **Regras**

Buscar | Crear

Componente: Indicador: Nivel:

Componente	Indicador	Nivel	Repeticións	Mensaxe			
AUDITORIA ELIMINACIÓN BATCH	Funcionamento tarefa	●	2	O batch de eliminación de auditoría non funciona			
Consulta de bens inmobles	Estado servizo	●	3	O servizo consulta de bens inmobles encóntrase nun estado inconsistente			
Consulta de bens inmobles	Estado servizo	●	1	O servizo consulta de bens inmobles caiu			
Consulta de datos de identidade	Estado servizo	●	3	O servizo consulta de identidade caiu			
Alta en data na Seguridade Social	Tempo petición	●	5	O servizo de alta na Seguridade Social presenta un alto Tempo de petición			
SOLICITUDE RESPONSA BATCH	Funcionamento tarefa	●	3	A tarefa programada de Solicitud de Resposta non está funcionando correctamente			

Mostrando dende 1 ata 6 de 6 resultados

● Nivel roxo
 ● Nivel amarelo
 ● Nivel verde

Ilustración 54. Vista Búsqueda de reglas

La vista de búsqueda de reglas permite realizar búsquedas sobre las reglas creadas en el Módulo de Monitorización. Las búsquedas se podrán filtrar por componente, indicador y nivel.

El resultado de las búsquedas se mostrará en una tabla que podrá ser ordenada por columnas.

Desde esta vista se podrá ir a crear una incidencia, ver el detalle de una incidencia, editar una incidencia o eliminarla.

General | Configuración | Servicios | Incidencias | Registros | **Reglas**

Buscar | Crear

Componente
Alta en data na Seguridade Social

Indicador: Max. Repeticiónes:

Nivel: Mensaje:

Enviar a usuarios:

Listas Emails

Listas Emails

Destinatarios

Destinatarios

Ilustración 55. Vista de Detalle de regla

En la vista de detalle de una regla se podrán ver todos los datos de una regla. Entre estos datos estarán el componente, el indicador, el nivel de monitorización, número de repeticiones, mensaje, listas de destinatarios y emails de destinatarios.

Ilustración 56. Vista de Creación de una regla

La última vista de la zona de reglas es la vista de creación de una regla. A partir de esta vista se podrán crear nuevas reglas para controlar los avisos del Módulo de Monitorización. Para crear una regla se deberá indicar el componente, indicador, nivel de monitorización, número máximo de repeticiones, mensaje, listas de destinatarios y emails.

Las listas de destinatarios se podrán seleccionar a partir de un combo en el que se mostrarán las listas de emails de la Plataforma de Interoperabilidad. Además por si el destinatario del aviso de la regla no se encuentra en ninguna lista de emails se podrán introducir emails directamente.

8. Pruebas

Introducción

El apartado que se presenta a continuación mostrará un análisis del proceso de pruebas y validación del proyecto *Módulo de Monitorización*. Conviene diferenciar entre varios tipos de pruebas, *Pruebas unitarias*, *Pruebas de integración* y por último *Pruebas de aceptación*.

Las *Pruebas unitarias* junto con las *Pruebas de integración* ayudarán a comprobar durante la fase de desarrollo que se está implementando la funcionalidad correcta mientras que las *Pruebas de aceptación* se utilizarán al final del proyecto para validar que se ha cumplido con los requisitos del proyecto.

8.1. Pruebas unitarias y de integración

El desarrollo del Módulo de Monitorización sigue la secuencia: creación de las entidades del modelo, capa de acceso a datos, capa de servicios y por último vistas y procesos batch.

Siguiendo la dinámica del desarrollo se comienza creando pruebas unitarias con Junit para las clases de las capas más bajas como son las entidades y la capa de acceso a datos. Las pruebas unitarias tendrán el objetivo de validar todas las funcionalidades que aporta cada clase.

Una vez terminado el desarrollo de las capas más bajas, entidades y capa de acceso a datos se cuenta con un conjunto de pruebas Junit que son utilizadas como Pruebas de regresión para comprobar que todo lo implementado hasta el momento funciona correctamente.

Un conjunto de Pruebas de regresión no es más que un conjunto de pruebas que se utiliza para verificar que todo lo implementado anteriormente produce errores. Es importante destacar que la ejecución de las pruebas no asegura la ausencia de errores sino que revela la presencia de estos.

Es muy importante contar con un conjunto de Pruebas de regresión ya que, cualquier pequeño cambio que se pueda producir en el desarrollo futuro, puede producir que código que antes funcionaba y parecía que no tenía relación alguna con los desarrollos siguientes, deje de funcionar.

Hasta este punto, se ha hablado de pruebas para las implementaciones de las capas más bajas. Para probar las capas superiores también se desarrollan pruebas Junit aunque, en este caso, ya podemos hablar de Pruebas de integración más que de Pruebas unitarias.

En el desarrollo de un servicio se pueden estar utilizando varias clases de acceso a datos o incluso funcionalidades de otros servicios. Esto evidencia que, aunque el servicio funcione correctamente, éste depende tanto de una buena implementación, como de que el resto de módulos/unidades que utiliza lo también lo estén. Es por ello que hablamos de Pruebas de integración y no de Pruebas unitarias.

Se mantendrá el uso de Junit para la comprobación de las funcionalidades de las capas superiores ya que, al contar con el conjunto de Pruebas regresivas, si se produce un error podemos asegurar que es debido a la funcionalidad que estamos probando. Lo podemos asegurar porque lanzando las Pruebas de regresión comprobamos que el conjunto de

módulos/unidades de las que depende la funcionalidad a probar están funcionando correctamente.

8.2. Pruebas de aceptación

Uno de los puntos más importantes en la finalización del *Módulo de Monitorización*, es que cumpla con los requisitos establecidos durante el análisis. Para cumplir con este punto se utilizarán las *Pruebas de aceptación*.

Las *Pruebas de aceptación* deben de ser ejecutadas por el cliente pero, en este caso, al encontrarnos en un entorno de Proyecto de empresa/ Trabajo de fin de grado, en el que no se puede esperar a una validación por parte del cliente, lo que se hará será que las *Pruebas de aceptación* serán ejecutadas por el responsable del *Módulo de Monitorización Sergio Nogueiras Fernández*.

A continuación se detallan las *Pruebas de aceptación* y el resultado obtenido tras su ejecución:

P1-> RF1

Descripción

Se dará de alta un servicio en el Catálogo de Productos de la PLATINT y luego se dará de alta en la monitorización a partir del portlet del MM. El perfil que se asignará en la monitorización será cualquiera que tenga el procedimiento de Ping de servicio.

Se dará de alta en el Catálogo de productos la tarea programada de Solicitud de respuesta.

Validación

Se accederá al buscador de registros en el portlet del MM y se comprobará que existen registros para el servicio dado de alta y para la tarea programada de Solicitud de respuesta.

Resultado

Validación correcta.

P2-> RF2, RF3

Descripción

A través de la configuración del portlet del MM se creará un nuevo perfil (perfil_1) con los procedimientos Ping de servicio y Trazas de servicio. Se dará de alta un nuevo servicio y se le asignará el perfil_1. Por último se realizará una consulta al servicio dado de alta a través del portlet de Consulta de la PLATINT.

Validación

Se accederá a la configuración del servicio creado y se comprobará si contiene los procedimientos Ping de servicio y Trazas de servicio con una frecuencia de 10

minutos. Además también se observará si contiene los indicadores Estado servicio, Estado petición, Tiempo petición, Tiempo conector ida, Tiempo conector-bus, Tiempo bus ida, Tiempo servicio final, Tiempo bus vuelta, Tiempo bus-conector y Tiempo conector vuelta.

Se comprobará a través de la zona de registros si existen registros para todos los indicadores asignados al servicio.

Resultado

Validación correcta.

P3-> RF4

Descripción

Se dará de alta un perfil (perfil_1) a través de la configuración del Módulo de monitorización con el procedimiento de Ping de servicio.

Se asignará el perfil_1 al servicio Consulta de identidad y se comprobará que el servicio esté activo tanto en el Catálogo de Productos como en el Módulo de Monitorización. Además se asegurará que todos los conectores del servicio están disponibles.

Se comprueban los registros generados para el servicio de Consulta de identidad.

Se da de baja al menos uno de los conectores del servicio Consulta de identidad.

Se comprueban los registros generados para el servicio de Consulta de identidad.

Validación

En la primera comprobación de los registros deberá existir un registro para el componente Consulta de identidad e indicador Estado servicio con el valor 0.

En la segunda comprobación de los registros deberá existir un registro para el componente Consulta de identidad e indicador Estado servicio con el valor 1.

Resultado

Validación correcta.

P3.1-> RF4

Descripción

Se accede a la pantalla de detalles de un servicio a través de la zona de Servicios del Módulo de Monitorización. Se pulsa el botón de ping servicio.

Validación

Se muestra una pantalla con todos los conectores del servicio divididos en consumidores y productores. De cada conector se informa de su estado y del tipo de comunicación.

Resultado

Validación correcta.

P4-> RF5

Descripción

Se dará de alta la tarea programada de Solicitud de respuesta en el Catálogo de Productos.

Se asegurará que todos los componentes de la PLATINT están disponibles para que la Solicitud de respuesta funcione correctamente.

Se comprobarán los registros generados para Solicitud de respuesta.

Se desactivará el bus de servicios y se volverán a comprobar los registros generados para Solicitud de respuesta.

Validación

En la primera comprobación de los registros generados para componente *Solicitud de respuesta* e indicador *Funcionamiento tarea* el valor será 0.

En la segunda comprobación el registro tendrá el valor 1.

Resultado

Validación correcta.

P5-> RF6

Descripción

Se dará de alta un perfil (*perfil_1*) a través de la configuración del *Módulo de monitorización* con el procedimiento de *Trazas de servicio*.

Se asignará el *perfil_1* al servicio *Consulta de identidad* y se comprobará que el servicio esté activo tanto en el Catálogo de Productos como en el Módulo de Monitorización.

Se realizará una petición a través del portlet de Consulta al servicio de *Consulta de Identidad*.

Se esperará 10 minutos y se comprobarán los registros generados para el servicio de *Consulta de identidad*.

Validación

En la comprobación de los registros deberán existir registros para el componente *Consulta de identidad* para todos los indicadores de tiempo que genera el procedimiento de *Trazas de servicio*. En caso de no existir indicadores de tiempo deberá existir al menos el indicador de *Estado petición* y el resultado será igualmente correcto.

Resultado

Validación correcta.

P6-> RF7, RF17

Descripción

Se accederá al portlet de Módulo de monitorización a la zona de Registros.

Se asegurará que exista al menos un componente activo en la monitorización y en el Catálogo de Productos.

Se realizará al menos una búsqueda por cada parámetro de búsqueda de los Registros.

Validación

Existirán registros creados por el *Módulo de monitorización* y no existirá posibilidad de eliminar ninguno de ellos.

Resultado

Validación correcta.

P7-> RF8

Descripción

Para realizar esta prueba será necesario crear unos datos ficticios a través del uso de Junit ya que, para el análisis de los niveles de servicio es necesario contar con datos de al menos un mes de monitorización.

Se crearán 3 niveles de servicio para el servicio Consulta de Identidad y el indicador Estado de servicio. Los valores de los niveles de servicio serán 0.99, 0.93 y 0.92.

Se forzará el cálculo de los niveles de servicio para Consulta de Identidad y el indicador Estado de servicio a través de Junit. El nuevo cálculo debe tener el valor 0.90. Se asegurará que los umbrales del indicador Estado de servicio para Consulta de Identidad tengan los valores 0.8 y 0.9.

Validación

Se podrán ver los niveles de servicio calculados a través de la zona de Registros del *Módulo de Monitorización*. Estos registros aparecerán con tipo de registro *Nivel de servicio*.

En la zona de incidencias deberá existir una incidencia de nivel de servicio para Consulta de Identidad indicando que el nivel de servicio *Estado de servicio* sufre una degradación.

Resultado

Validación correcta.

P8-> RF9, RF10, RF11, RF14

Descripción

Se asegurará que el servicio Consulta de Identidad está activo en la monitorización y en el Catálogo de Productos. Además este servicio tendrá asignado el procedimiento de Ping de servicio.

Se creará una regla para el componente Consulta de Identidad y el indicador Estado de servicio con los valores nivel de monitorización rojo, máximo de repeticiones 3, un mensaje y un email.

Se observarán los Registros.

Se dará de baja un conector del servicio de Consulta de identidad que nos hemos asegurado que estaba activo en las dos partes y que tenía asignado el Ping de servicio.

Se observarán las Incidencias.

Se resolverá la incidencia que informa que el servicio de Consulta de servicio está caído.

Validación

En la observación de los Registros se podrá ver cómo se van creando nuevos registros cada cierto tiempo para todos los componentes activos.

A partir del momento en el que se desactive un conector para un servicio que se esté monitorizando se observará que los registros creados informan de que el indicador *Estado de servicio* toma el valor 1.

Después de que se creen 3 registros con valor 1 para *Consulta de Identidad* e indicador *Estado de servicio* se recibirá un email a la dirección configurada con el mensaje configurado en la regla.

Además en la zona de incidencias existirá una incidencia para *Consulta de identidad* e indicador *Estado de servicio* con estado sin resolver y nivel de monitorización rojo.

Al resolver la incidencia esta cambiará de estado a resuelta.

Resultado

Validación correcta.

P9-> RF12, RF13

Descripción

Se crearán una serie de incidencias a través del uso de Junit.

Se accederá a la zona de Incidencias del portlet del Módulo de Monitorización y se realizarán al menos una búsqueda por cada posible parámetro de búsqueda de las incidencias.

Validación

Las búsquedas de incidencias devuelven los resultados esperados.

Resultado

Validación correcta.

P10-> RF15

Descripción

Desde la zona de Reglas del Módulo de Monitorización se creará una regla.

Se verán los detalles de la regla.

Se modificará el nivel de monitorización de la regla y el número de repeticiones.

Se volverán a ver los detalles de la regla.

Se eliminará la regla.

Validación

La primera vez que se vean los detalles de la regla aparecerán los mismos datos que se introdujeron en su creación.

La segunda vez que se ven los detalles de la regla aparecerán los mismos datos que en la creación pero con otro nivel de monitorización y otro número de repeticiones, los que se establecieron en la modificación.

Una vez eliminada la regla debe desaparecer.

Resultado

Validación correcta.

P11-> RF16

Descripción

Se crearán un conjunto de reglas a través de la zona de Reglas.

Se realizará al menos una búsqueda por cada parámetro de búsqueda de las Reglas.

Validación

Las diferentes búsquedas de reglas devuelven los resultados esperados.

Resultado

Validación correcta.

P12-> RF18

Descripción

Desde la zona de Servicios del *Módulo de monitorización* se activará un servicio que aún no haya sido activado previamente y se le asignará un perfil.

Se accederá a la configuración del servicio activado.

Se desactivará el servicio y se volverá a activar.

Validación

La primera vez que se activa el servicio se solicita la selección de un perfil. En la configuración del servicio aparecerán los procedimientos e indicadores del perfil asignado.

Al desactivar el servicio este cambia de estado y al volverlo activar cambia también de estado pero esta vez sin solicitar un perfil.

Resultado

Validación correcta.

P13-> RF19

Descripción

Se accede a la zona de Servicios del *Módulo de Monitorización*.

Validación

Se muestran todos los servicios disponibles en el *Catálogo de servicios* menos aquellos que han sido eliminados. De cada servicio se muestra su nombre, estado en el catálogo, responsable y estado en monitorización. En función del estado de cada servicio se podrá activar, desactivar, ver detalles o ir a la configuración.

Resultado

Validación correcta.

P14-> RF20

Descripción

Se accede a la zona de Servicios del *Módulo de Monitorización*.

Se elige un servicio que esté activo en la monitorización y se pulsa en Detalles del servicio.

Validación

En los detalles del servicio se debe mostrar el nombre del servicio, el responsable, el estado de monitorización, el estado en el *Catálogo de servicio*, los últimos registros y las últimas incidencias.

Además se podrá acceder desde el detalle a realizar un ping al servicio y a su configuración. En caso de que el servicio tenga una plantilla de consulta de prueba configurada también se podrá realizar una consulta desde la vista de detalles.

Resultado

Validación correcta.

P15-> RF21

Descripción

Se accede a la zona de Servicios del *Módulo de Monitorización*.

Se elige un servicio que esté activo en la monitorización y se pulsa en Configuración del servicio.

Se sube una plantilla de consulta de prueba si no existe y si existe se cambia.

Se cambia la frecuencia de los procedimientos y los umbrales de los indicadores

Validación

Se vuelve a acceder a la configuración del servicio y se debe de mostrar la configuración con los últimos datos que se han establecido.

Resultado

Validación correcta.

P16-> RF22, RF25

Descripción

Se accede a la zona de *Configuración* del *Módulo de Monitorización*.

Se modifica la fecha de inicio del análisis proactivo.

Se crea un perfil de monitorización con nombre *Perfil_1* y que tenga todos los procedimientos disponibles.

Se muestra los detalles de la configuración.

Se modifica el perfil creado quitándole un procedimiento.

Se muestran los detalles de la configuración.

Se elimina el perfil creado.

Validación

Tras la primera modificación se deben de mostrar los detalles de la configuración con los nuevos datos establecidos, entre los que aparecerán el nuevo perfil y la nueva fecha de inicio del análisis proactivo. Al modificar el perfil y quitarle un procedimiento este ya no se verá la siguiente vez que se vean los detalles de la configuración.

Por último al eliminar el perfil este ya no aparecerá en los detalles de la configuración.

Resultado

Validación correcta.

P17-> RF23

Descripción

Se accede a la pantalla de configuración de un servicio a través de la zona de *Servicios* del *Módulo de Monitorización*.

Se elige un perfil distinto al que tiene el servicio.

Se establece el perfil.

Validación

Los procedimientos e indicadores del servicio cambian por los procedimientos y servicios que tiene el nuevo perfil del servicio.

Resultado

Validación correcta.

P18-> RF24

Descripción

Se accede a la pantalla de detalles de un servicio a través de la zona de servicios del *Módulo de Monitorización*. El servicio al que se accede a ver sus detalles debe ser uno que tenga una plantilla de consulta de prueba configurada.

Se pulsa el botón de consulta de prueba.

Validación

Se descarga un fichero xml con la respuesta del servicio.

Resultado

Validación correcta.

9. Conclusiones y mejoras

En este apartado expondremos las conclusiones obtenidas del desarrollo de este proyecto. Dichas conclusiones versarán sobre el resultado final del mismo y ciertas mejoras que se podrían aplicar al Módulo de Monitorización.

En lo referente al resultado alcanzado, lo principal es evaluar si se han cumplido los objetivos marcados.

Respecto al objetivo de realizar un seguimiento de los componentes de la Plataforma de Interoperabilidad se puede concluir que se ha alcanzado con éxito. El Módulo de Monitorización realiza una vigilancia continua de los diferentes componentes de la Plataforma de Interoperabilidad, de modo que es posible conocer su estado en cada momento. Además los distintos estados de los componentes son registrados y en caso de que alguno de ellos suponga un estado de alerta el Módulo de Monitorización envía un aviso a los responsables automáticamente, cumpliendo así otro de los objetivos.

El cálculo de los niveles de servicio era otro de los objetivos del Módulo de Monitorización. El Módulo de Monitorización realiza este cálculo a partir de los registros generados por la monitorización. Además también analiza la tendencia de estos y avisa en caso de que esta sea de degradación.

De los objetivos referentes al interfaz web, a través de la cual se permite la interacción del usuario con el Módulo de Monitorización, podemos decir que se han alcanzado todos. La Consola de monitorización permite:

- Visualizar y exportar los registros y las incidencias generadas por la monitorización.
- Configurar la fecha de inicio del análisis proactivo y los perfiles de monitorización.
- Configurar una plantilla de prueba para un servicio y realizar una consulta al servicio utilizando dicha plantilla.
- Ver el estado en tiempo real de los consumidores y productores de un servicio.
- Controlar el acceso a la Consola de monitorización en función del rol que tenga cada usuario. Además, no sólo controla el acceso sino que permite restringirlo a ciertas partes y ciertos tipos de operación.

Como acabamos de ver, los objetivos del proyecto han sido cumplidos y por lo tanto el resultado alcanzado es satisfactorio. Pero, además de haber cumplido los objetivos del proyecto cabe destacar que:

- La solución alcanzada ha sido diseñada en función del resto de módulos de la Plataforma de Interoperabilidad, permitiendo la fácil comprensión del módulo por parte de futuros desarrolladores que vayan a realizar tareas de mantenimiento y evolutivos sobre el Módulo de Monitorización.
- El diseño utilizado permite incorporar nuevos procedimientos sin tener que modificar ninguna línea de código del Módulo de Monitorización. Esto es muy importante debido a que el crecimiento del Módulo de Monitorización se basará en la incorporación de nuevos procedimientos que calculen nuevos indicadores, de forma que los datos obtenidos para conocer el estado de la Plataforma de Interoperabilidad sean cada vez mayores.

Aunque los objetivos del proyecto han sido alcanzados, durante el desarrollo de este se han ido detectando ciertos aspectos que se podrían mejorar. Estos puntos son presentados a continuación:

- El Método de las dos medias utilizado para la predicción de los niveles de servicios no es muy preciso. El motivo es que se basa en la media aritmética y por lo tanto los datos atípicos tienen una gran influencia. Consecuentemente, las predicciones de los niveles de servicio también estarán muy influenciadas por los valores atípicos y pueden alejarse de la realidad. La solución a este problema sería utilizar otro algoritmo que tuviese en cuenta la dispersión de los datos a la hora de realizar las predicciones.
Cambiar el Método de las dos medias por otro algoritmo no supondría un gran esfuerzo, ya que el Método de las dos medias es independiente del resto del Módulo de Monitorización y por lo tanto, simplemente supondría cambiar la función que lo implementa.
- Realizar un análisis de los registros que genera la monitorización con el fin de ajustar los umbrales de los indicadores y las frecuencias para los procedimientos a unos valores más precisos.
- Crear un sistema de notificaciones en el Portal de Interoperabilidad. Esto permitirá al Módulo de Monitorización generar notificaciones de modo que, cuando un usuario inicie sesión en el Portal de Interoperabilidad, vea una notificación que le avise de que hay algún problema en la monitorización sin tener que acceder a la Consola de monitorización.
- Integrar el Módulo de Monitorización con una herramienta de reporting como puede ser Pentaho. Esto permitirá realizar un estudio más detallado de los datos obtenidos por la monitorización.
- Incorporar procedimientos que calculen nuevos indicadores, con el fin de contar con más datos para conocer el estado de la Plataforma de Interoperabilidad. De los posibles indicadores que se pueden incorporar se podrían destacar aquellos que aporten información acerca de la infraestructura. Ejemplos de estos indicadores podrían ser el porcentaje de uso de memoria o la velocidad de cpu.

Referencias

ESPINOSA, D. *Estimación de la tendencia de una serie temporal*. Recuperado el 19/05, 2014, de http://davidespinoza.es/joomla/index.php?option=com_content&view=article&id=339:estimacion-de-la-tendencia-de-una-serie-temporal&catid=80: analisis-externo

EVERIS SPAIN SL. (2014). *Documentación Interna Proyecto PLATINT*. A Coruña: Everis.

GARIT, F. *Spring MVC Portlets*. Recuperado el 05/02, 2014, de <http://liferay.wikispaces.com/file/view/Liferay%2BSpring%2BExtJs.pdf>

HIBERNATE. Recuperado el 24/02, 2014, de <http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/>

JFREECHART. Recuperado el 01/03, 2014, de <http://www.jfree.org/jfreechart/api/javadoc/index.html>

NAGIOS CORE DOCUMENTATION. Recuperado el 18/02, 2014, de <http://nagios.sourceforge.net/docs/nagioscore/4/en/toc.html>

ORACLE WEBLOGIC SERVER ONLINE DOCUMENTATION LIBRARY 11GRELEASE 1 (10.3.6). Recuperado el 18/02, 2014, de http://docs.oracle.com/cd/E23943_01/wls.htm

RICHARD SEZOV, J. (2011). *Liferay in Action.-The Liferay official guide to Liferay Portal Development*. Manning. Greenwich: Manning.

Apéndice A: Manual de despliegue

A continuación se presenta el manual de despliegue que ayudará a desplegar el Módulo de Monitorización en los diferentes entornos del cliente.

Para desplegar el Módulo de Monitorización será necesario realizar los siguientes pasos:

1. Crear aplicación de Monitorización

Se crea a través del Catálogo de Productos de la Plataforma de Interoperabilidad una aplicación para Monitorización. Su código será TNA_014 y representará al portlet de Monitorización.

2. Crear partes aplicación, tipo operación y rango rol

Lanzar el script 2014-06-12_3.sql. Este script crea las Parte_aplicación, Parte_aplicación_tipo_operación, Parte_aplicación_rango_rol (SuperAdmin) y Parte_aplicación_rango_rol (Admin).

3. Seguridad

A los roles que tengan rango rol SuperAdmin o Admin se le deben asignar a través del portlet de Seguridad las autorizaciones oportunas sobre la aplicación de Monitorización.

4. Creación base de datos

Ejecutar el script de creación de la base de datos del Módulo de Monitorización (2014-06-12_1.sql). Este script crea las tablas del modelo de datos del Módulo de Monitorización y las entidades maestro.

5. Dar de alta los componentes de tipo tarea

Ejecutar el script 2014-06-12_2.sql. Antes de ejecutar el script revisar si los códigos de las tareas programadas que se quieren dar de alta son los correctos.

6. Librerías Monitorización-model y Common-model

Se deben subir al servidor de aplicaciones las librerías de monitorización-model y common-model. La primera contiene la capa de servicios y la segunda la capa de acceso a datos.

7. Portlet Consulta de monitorización

Para subir el portlet de Consulta de monitorización se empaquetará en un war y se desplegará como una aplicación más en el servidor de aplicaciones. Es importante que el war solo lleve las librerías que no son compartidas con ninguna otra aplicación. Estas librerías son:

```
displaytag-1.2.jar  
displaytag-portlet-1.2.jar  
jcommon-1.0.21.jar  
jfreechart-1.0.17.jar  
jstl-1.2.jar  
log4j.jar  
spring-webmvc-3.1.2.RELEASE.jar  
spring-webmvc-portlet-3.1.2.RELEASE.jar  
util-bridges.jar  
util-java.jar  
util-taglib.jar
```


Apéndice B: Manual de usuario

El manual que a continuación se presenta ayudará a los usuarios del Módulo de Monitorización a saber qué acciones pueden realizar en la Consola de monitorización y cómo.

1. ACCESO AL MÓDULO DE MONITORIZACIÓN

Con el fin de acercar la funcionalidad de la monitorización al usuario, se dispone de un módulo accesible para los usuarios autorizados en la sección de Administración del portal. Para poder acceder al citado módulo, debemos clicar en la pestaña “MONITORIZACIÓN” del menú superior del Portal de Interoperabilidad.



Ilustración 57. Acceso al módulo de monitorización

El módulo cuenta con 5 secciones, si queremos acceder a cada una de ellas sólo tenemos que clicar sobre el correspondiente enlace.

2. GENERAL

Esta sección se mostrara automáticamente al acceder al módulo de monitorización. Es útil como resumen del estado general de la monitorización.

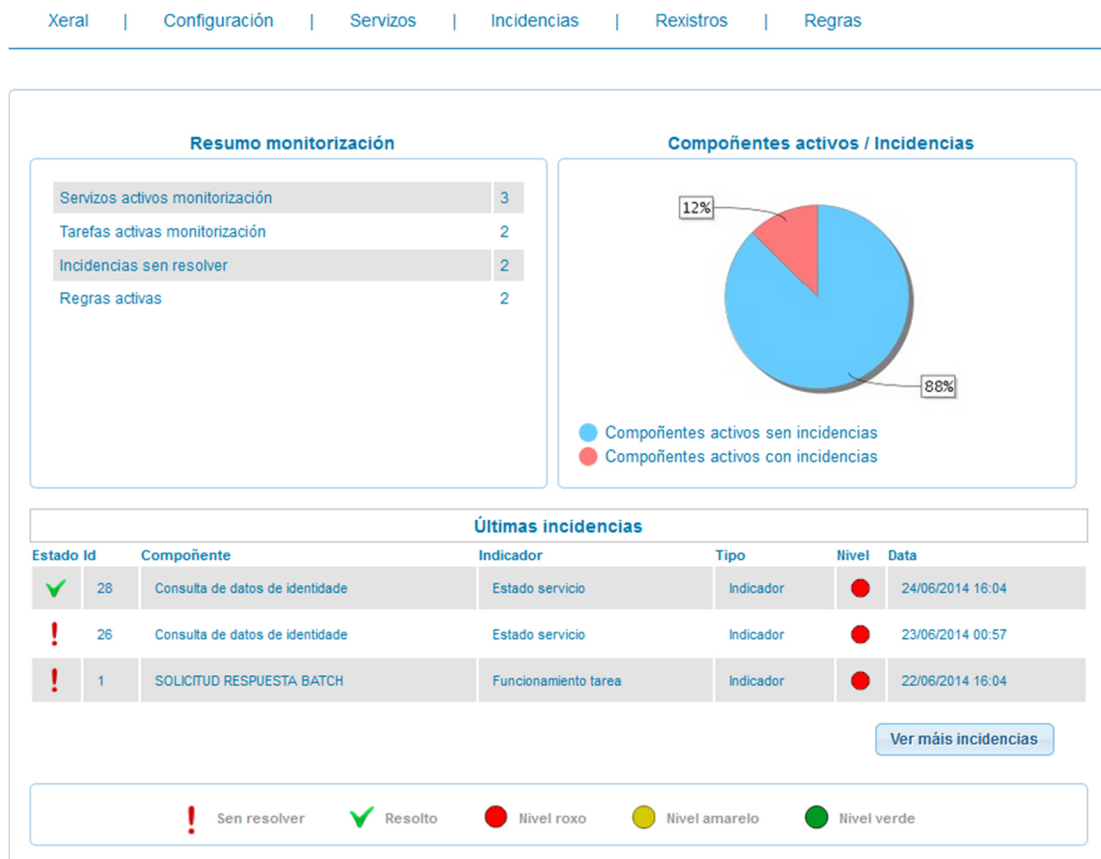


Ilustración 58. Sección General

En la parte superior izquierda de la pantalla podremos ver un resumen general de la monitorización, donde se muestra el número de servicios y tareas que están siendo monitorizados actualmente, junto con el número de incidencias sin resolver y las reglas que hay activas.

A la derecha se muestra un gráfico con el porcentaje de componentes que poseen al menos una incidencia sin resolver.

Ya en la tabla inferior podremos ver un listado de las últimas incidencias que se detectaron en la monitorización, en las que se nos detallará el estado de la misma, el componente, el indicador, la fecha y el nivel de criticidad.

Últimas incidencias						
Estado	Id	Componente	Indicador	Tipo	Nivel	Data
!	32	Consulta de bens inmuebles	Estado servicio	Indicador	●	26/06/2014 19:50
!	31	Consulta de bens inmuebles	Estado servicio	Indicador	●	26/06/2014 19:35
✓	28	Consulta de datos de identidad	Estado servicio	Indicador	●	24/06/2014 16:04
!	26	Consulta de datos de identidad	Estado servicio	Indicador	●	23/06/2014 00:57
!	1	SOLICITUD RESPUESTA BATCH	Funcionamiento tarea	Indicador	●	22/06/2014 16:04

[Ver más incidencias](#)

! Sen resolver ✓ Resolto ● Nivel roxo ● Nivel amarelo ● Nivel verde

Ilustración 59. Últimas incidencias

Si queremos ver más incidencias o gestionar alguna de ellas debemos clicar sobre el botón “Ver más incidencias”, lo que nos llevará directamente a la sección de “Incidencias”.

3. CONFIGURACIÓN

En la parte superior de esta sección se muestra la fecha de inicio del análisis proactivo. En la parte inferior se muestran los perfiles de monitorización que engloban procedimientos para poder asignarlos a los componentes. Si queremos cambiar cualquiera de las configuraciones podemos pulsar sobre “Editar”.

Xeral | **Configuración** | Servizos | Incidencias | Rexistros | Regras

Inicio do análisis proactivo
21/07/2014 00:00

Perfis de monitorización

Nome perfil	Procedementos
Servizo	Ping servicio Trazas servicio
Só ping	Ping servicio

[Editar](#)

Ilustración 60. Sección de configuración

Pulsando el botón “*Editar*” accederemos a la pantalla de edición.

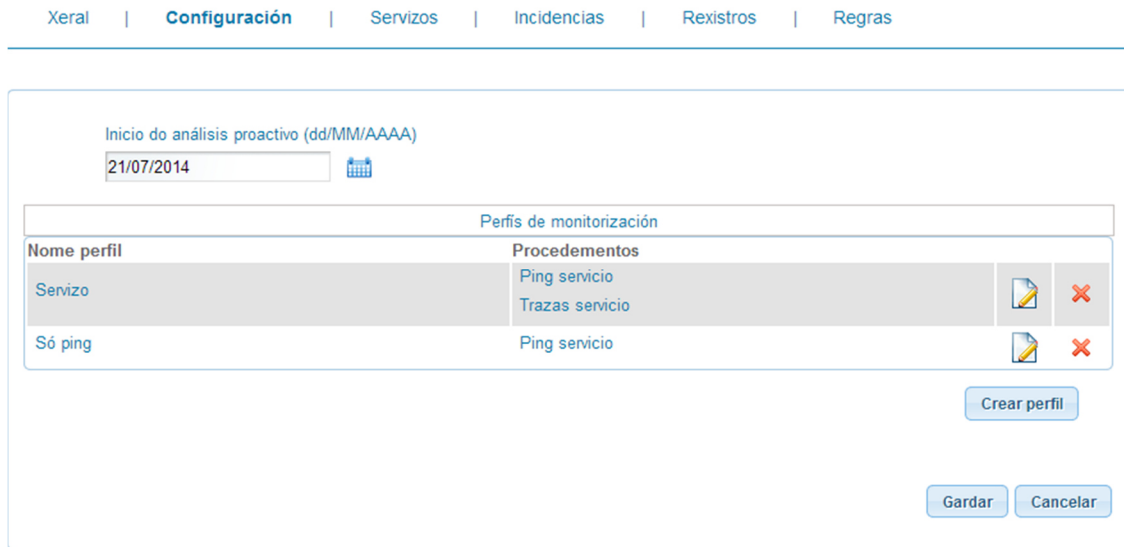


Ilustración 61. Edición de configuración

Desde esta pantalla podemos configurar en cualquier momento la fecha desde la cual se realizará el análisis proactivo de los registros obtenidos en la monitorización. Para poder crear los indicadores de “Nivel de servicio” el sistema analizará los registros a partir de la fecha que establezcamos aquí.

También podremos realizar la gestión de los perfiles de monitorización. Un perfil englobará uno o varios procedimientos. Los procedimientos que tenga incluidos un perfil, dependerán del tipo de componente al que vaya ser destinado / asignado.

3.1 FECHA DE INICIO DEL ANÁLISIS PROACTIVO

Para poder cambiar la fecha de inicio del análisis proactivo nada solamente es necesario, desde el menú de edición, teclear otra fecha en el cuadro de texto o escogerla mediante el desplegable.



Ilustración 62. Editar fecha inicio análisis proactivo

Una vez introducida la fecha, debemos pulsar “*Guardar*”.

3.2 CREAR UN NUEVO PERFIL

Para crear un nuevo perfil dentro del menú de edición, debemos pulsar “Crear perfil”.

Nome perfil	Procedementos		
Servizo	Ping servicio		
	Trazas servicio		
Só ping	Ping servicio		

Ilustración 63. Acceder a crear perfil

Se abrirá una ventana modal en la que deberemos indicar el nombre del nuevo perfil y escoger al menos un procedimiento.

Nome perfil *

Procedementos *

Trazas tarea

Ping servicio

Trazas servicio

Ilustración 64. Aceptar y guardar cambios perfil

Tras cubrir los datos hay que clicar en el botón “Aceptar” y deberemos pulsar “Guardar” también en la pantalla de edición para aplicar los cambios.

3.3 MODIFICAR UN PERFIL

Para modificar un perfil de monitorización dentro del menú de edición, debemos localizar el perfil que queramos modificar y pulsar en el botón izquierdo.

Perfis de monitorización			
Nome perfil	Procedementos		
Servizo	Ping servicio		
	Trazas servicio		
Só ping	Ping servicio		
Servizos	Trazas servicio		

Ilustración 65. Acceso a editar un perfil

En la ventana modal modificamos los datos.

Ilustración 66. Aceptar y guardar cambios

Tras cubrir datos hay que hacer clic primero en “Aceptar” y luego en “Guardar”.

3.4 ELIMINAR UN PERFIL

Para eliminar un perfil basta con clicar sobre el botón “Eliminar” y posteriormente “Guardar” cambios.

Ilustración 67. Eliminar un perfil

4. SERVICIOS

En esta sección podremos configurar que servicios queremos que se vayan a monitorizar. Es importante aclarar que sólo se monitorizarán aquellos servicios que figuren como “activos” en el catálogo de productos.

Para poder monitorizar un servicio, en esta sección deberemos asignarle un perfil.

4.1 ACTIVAR Y DESACTIVAR LA MONITORIZACIÓN DE UN SERVICIO

Para poder activar la monitorización de un servicio debemos pulsar el botón “Activar”.

▲ Nome Servizo	Estado servizo	Organismo responsable			
Alta en data na Seguridade Social	Activo	TGSS			
Autoservicio familia numerosa	Activo	Xunta de Galicia			
Certificación de titularidade	Activo	Catastro			
Certificado de discapacidade	Activo	Xunta de Galicia			

Ilustración 68. Botón de activación del servicio

Si el servicio ya fuera configurado alguna vez y simplemente se encontraba desactivado, el proceso de activación tendrá finalizado aquí, ya que restaurará la configuración anterior.

En caso de que el servicio no hubiese sido configurado previamente, entonces, obligará a seleccionar uno de los perfiles de monitorización existentes.

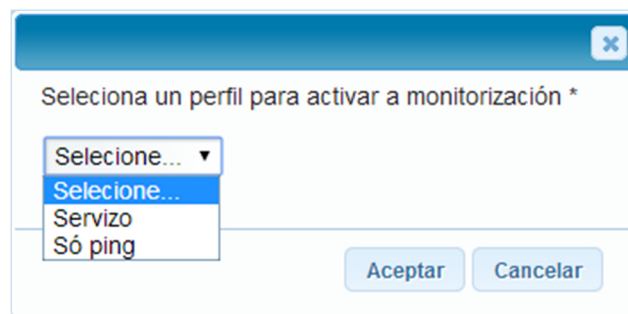


Ilustración 69. Selección perfil

Una vez seleccionado el perfil pulsamos “Aceptar”. Una vez realizado, la monitorización para ese servicio estará ya activa y comprobaremos que el botón habrá cambiado a color gris.

▲ Nome Servizo	Estado servizo	Organismo responsable			
Alta en data na Seguridade Social	Activo	TGSS			
Autoservicio familia numerosa	Activo	Xunta de Galicia			
Certificación de titularidade	Activo	Catastro			
Certificado de discapacidade	Activo	Xunta de Galicia			

Ilustración 70. Servicio activado

Para desactivar la monitorización del servicio, basta con pulsar sobre el mismo botón, y tras hacerlo, volverá a color azul. Aunque se desactive la monitorización de un servicio, la configuración del mismo se mantendrá guardada.

4.2 EDITAR MONITORIZACIÓN DE UN SERVICIO

Podemos cambiar la configuración de monitorización de un servicio que ya fuese configurado previamente. Para hacerlo debemos pulsar el botón de “Editar”.

▲ Nome Servicio	◆ Estado servicio	◆ Organismo responsable			
Alta en data na Seguridade Social	Activo	TGSS			
Autoservicio familia numerosa	Activo	Xunta de Galicia			
Certificación de titularidade	Activo	Catastro			
Certificado de discapacidade	Activo	Xunta de Galicia			

Ilustración 71. Acceso a editar configuración

Al pulsar el botón de “Editar” accederemos a la pantalla de edición.

Xeral | Configuración | **Servizos** | Incidencias | Rexistros | Regras

Servizo | **Configuración servicio**

Certificación de titularidade

Perfil

Servizo ▼ Establecer perfil

Datos de proba

Xa existe plantilla de probas. Editar Eliminar

Procedementos

Procedemento	Frecuencia
Trazas servicio	10
Ping servicio	10

Indicadores

Indicador	U.Baixo	U.Alto
Tempo bus volta	0.6	0.8
Tempo servizo final	0.6	0.8
Tempo petición	2.0	4.0
Tempo conector volta	0.6	0.8
Estado servizo	0.1	0.2
Tempo bus-conector	0.6	0.8
Estado petición	0.15	0.3
Tempo conector-bus	0.6	0.8
Tempo conector ida	0.6	0.8
Tempo bus ida	0.6	0.8

Gardar Cancelar

Ilustración 72. Pantalla de edición de la configuración de un servicio

En esta pantalla podremos realizar varias acciones.

4.2.1 ASIGNAR OTRO PERFIL DE MONITORIZACIÓN

En la esquina superior izquierda podemos cambiar a otro perfil de monitorización.

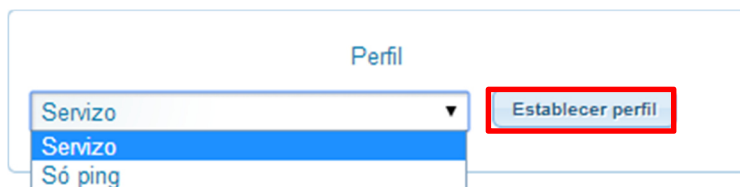


Ilustración 73. Asignar otro perfil

Sólo tenemos que seleccionar el perfil que queramos y pulsar “Establecer perfil”.

4.2.2 AÑADIR PLANTILLA DE PRUEBA

En la esquina superior derecha, podemos añadir un fichero XML con una petición que contenga datos de prueba para lanzar en el entorno de Producción. Esto permite a los usuarios del módulo realizar una petición de prueba al servicio para comprobar su correcto funcionamiento.

Para poder añadir un fichero debemos pulsar en el botón “Seleccionar archivo”.

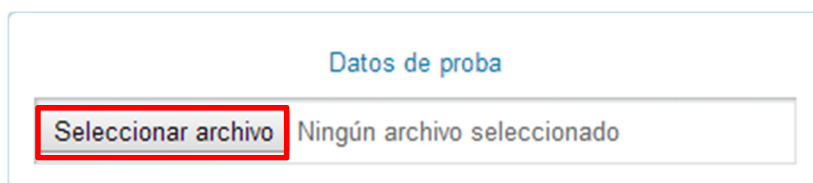


Ilustración 74. Añadir plantilla de prueba

Buscamos el fichero en cuestión y pulsamos “Abrir”.

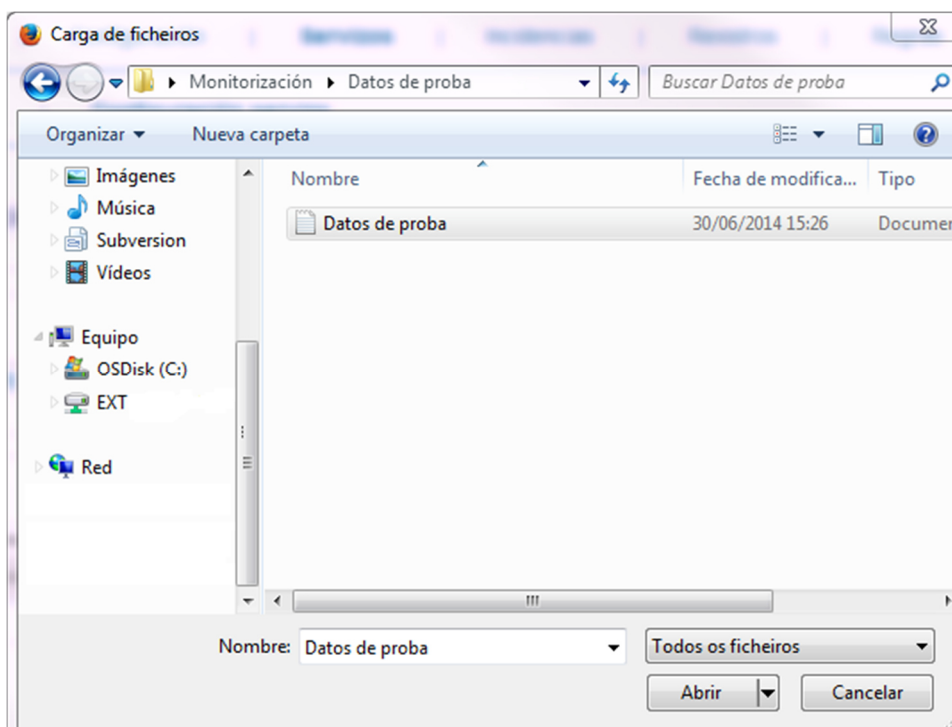


Ilustración 75. Subir fichero

Finalmente habrá que clicar en “Guardar” para que queden registrados los datos de prueba.

Una vez que el servicio ya tiene datos de prueba registrados, en esta sección aparecerán las opciones “Editar” y “Eliminar”.



Ilustración 76. Editar o eliminar plantilla de prueba

Pulsando sobre el botón editar se nos dará la opción de configurar otro fichero en lugar del actual. Eliminar vuelve a dejar el apartado en blanco. Después de usar cualquiera de los dos botones, habrá que clicar en “Guardar” para que se apliquen los cambios.

4.2.3 FRECUENCIA DE LOS PROCEDIMIENTOS

En la parte inferior izquierda de la pantalla de edición de monitorización del servicio, podemos variar la frecuencia en minutos con la que queremos que se ejecuten los procedimientos que tiene el servicio asignado.

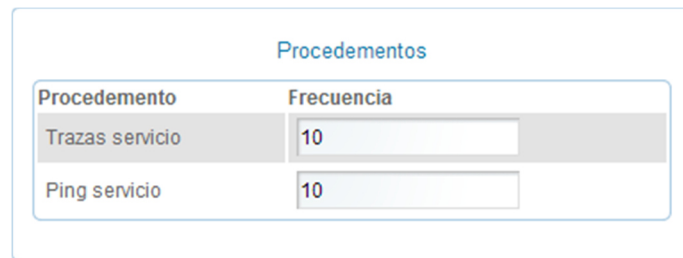


Ilustración 77. Frecuencia de ejecución de los procedimientos

Para hacerlo, tendremos que sustituir el valor por lo que queramos y pulsar en “Guardar”.

4.2.4 REGULAR LOS INDICADORES

Por último en la parte inferior derecha, tenemos la posibilidad de regular los indicadores de que calculan los procedimientos que el servicio tiene asignados, modificando los valores por defecto que trae para adecuarlos al servicio en cuestión. De este modo podremos variar los umbrales en los que se generarán los registros de color verde, amarillo o rojo.

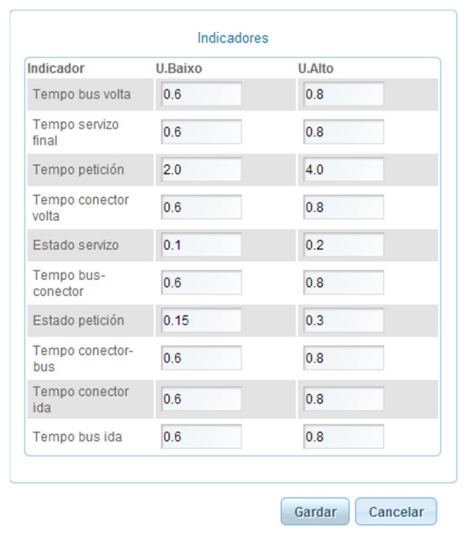


Ilustración 78. Regular indicadores

Según los valores que introduzcamos, cuando el sistema analice los registros que producen los procedimientos del servicio indicará que:

- Si el registro no llega al U.Bajo, el estado es verde.
- Si el registro se sitúa entre el U.Bajo y el U.Alto, el estado es amarillo.
- Si el registro es superior al U.Alto el estado es rojo.

Como en el resto de apartados de la edición del servicio, una vez realizados los cambios, debemos pulsar en “Guardar” para aplicar los cambios.

4.3 VER DETALLES MONITORIZACIÓN SERVICIO

Para ver la configuración de monitorización de un servicio que ya esté configurado, tenemos que clicar en el tercer botón empezando por la izquierda.

▲ Nome Servicio	Estado servicio	Organismo responsable			
Alta en data na Seguridade Social	Activo	TGSS			
Autoservicio familia numerosa	Activo	Xunta de Galicia			
Certificación de titularidade	Activo	Catastro			

Ilustración 79. Acceder a detalles servicio

Tras pulsar en el botón accedemos a la pantalla de detalles.

Nome Servicio
Consulta de datos de identidad

Organismo responsable
Administración General del Estado

Estado servicio
Activo

Estado monitorización
Activo

Perfil
Servizo

Últimas incidencias

Estado	Id	Compoñente	Indicador	Tipo	Nivel	Data
✓	33	Consulta de datos de identidade	Estado servicio	Indicador	●	27/06/2014 14:30
✓	26	Consulta de datos de identidade	Estado servicio	Indicador	●	24/06/2014 16:04
✓	26	Consulta de datos de identidade	Estado servicio	Indicador	●	23/06/2014 00:57

Últimos rexistros

Id	Indicador	Tipo	Clave	Valor	Data
3763	Estado servicio	Indicador		0.0	30/06/2014 12:20
3762	Estado servicio	Indicador		0.0	30/06/2014 12:18
3760	Estado servicio	Indicador		0.0	30/06/2014 12:16
3759	Estado servicio	Indicador		0.0	30/06/2014 12:14
3758	Estado servicio	Indicador		0.0	30/06/2014 12:12
3757	Estado servicio	Indicador		0.0	30/06/2014 12:10

Ilustración 80. Detalles servicio

En la parte superior de la pantalla de detalles se muestran todos los parámetros que tenemos configurados del servicio. Además de eso, podremos ver en la sección del medio las últimas incidencias del servicio y en la parte inferior los últimos registros del servicio.

Si pulsamos sobre “Ver más incidencias” o sobre “Ver más registros”, nos llevará a las pantallas de Incidencias o Registros respectivamente.

Además de poder visualizar la configuración de monitorización de los servicios desde esta pantalla también podremos realizar otras tareas, que permiten verificar en tiempo real la disponibilidad de un servicio.

4.4 PING SERVICIO

Si clicamos en el botón “Ping servicio”, iremos a una pantalla donde se nos indicarán los consumidores y productores de los que dispone el servicio, informando si éstos están respondiendo o no.

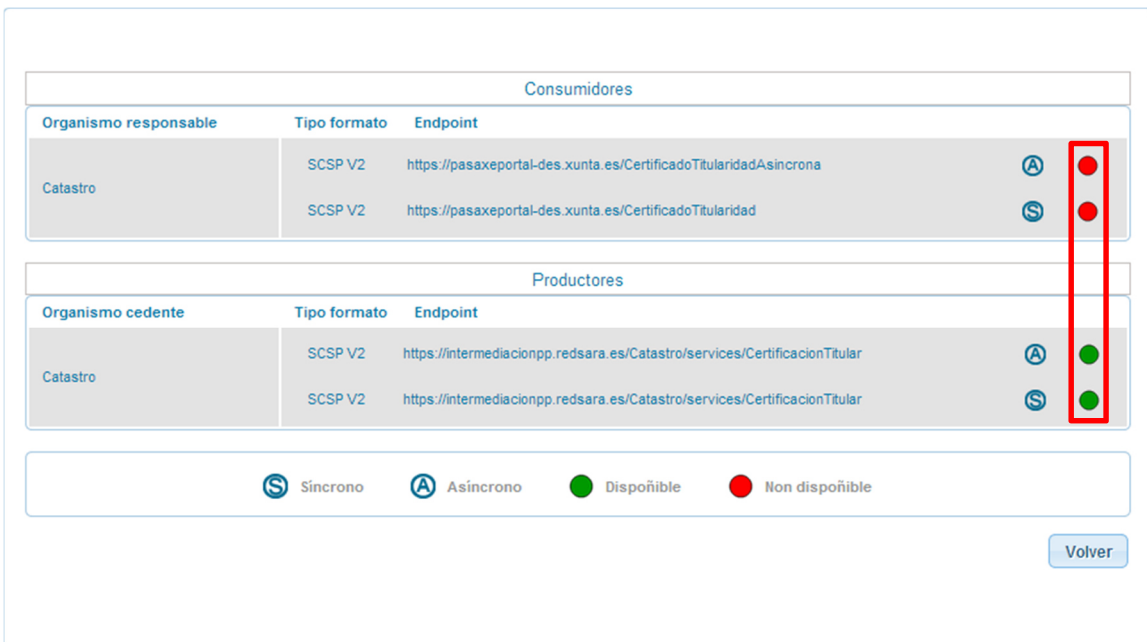


Ilustración 81. Pantalla ping servicio

A la derecha de cada uno de los consumidores y productores, se informa con color verde si responde correctamente y con rojo en caso contrario.

4.5 CONSULTA DE PRUEBA

Este botón y funcionalidad sólo se mostrará en el caso de que el servicio tenga disponibles datos de prueba para el entorno de producción y tenga una plantilla de prueba configurada. Pulsando el botón, se realizará automáticamente la petición registrada, la cual nos puede informar en tiempo real de si el servicio está funcionando de modo correcto o no.

La respuesta que obtendremos al pulsar el botón “Consulta de prueba” será un fichero XML que podremos descargar para comprobar cuál fue el resultado de la petición realizada.

```

▼<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  ▼<soapenv:Body>
    ▼<soapenv:Fault>
      <faultcode xmlns:axis2ns767="http://schemas.xmlsoap.org/soap/envelope/">axis2ns767:Server</faultcode>
      ▼<faultstring>
        0242 Error Genérico devuelto por el BackOffice. Error Genérico devuelto por el BackOffice
      </faultstring>
      ▼<detail>
        ▼<Atributos xmlns="http://www.map.es/scsp/esquemas/V2/soapfaultatributos">
          <IdPetición>7490074900</IdPetición>
          <TimeStamp>2014-06-30T17:16:04.987+02:00</TimeStamp>
          <NumElementos>1</NumElementos>
          ▼<Estado>
            <CodigoEstado>0242</CodigoEstado>
            <CodigoEstadoSecundario/>
            ▼<LiteralError>
              Error Genérico devuelto por el BackOffice. Error Genérico devuelto por el BackOffice
            </LiteralError>
            <TiempoEstimadoRespuesta>0</TiempoEstimadoRespuesta>
          </Estado>
          <CodCertificado>CDISFW501</CodCertificado>
        </Atributos>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

Ilustración 82. Respuesta a una petición de prueba

5. REGISTROS

En esta sección podemos consultar de forma detallada y en tiempo real, los registros de monitorización creados por cada componente e indicador.

Xeral | Configuración | Servicios | Incidencias | **Registros** | Regras

Id <input type="text"/>	Tipo Calquera ▼	Componente Calquera ▼	
Indicador Calquera ▼	Dende <input type="text"/>	Ata <input type="text"/>	<input type="button" value="Buscar"/> <input type="button" value="Limpiar"/>

Id	Componente	Indicador	Tipo	Clave	Valor	Data
3763	Consulta de datos de identidad	Estado servicio	Indicador		0.0	30/06/2014 12:20
3762	Consulta de datos de identidad	Estado servicio	Indicador		0.0	30/06/2014 12:18
3761	Certificación de titularidad	Estado servicio	Indicador		1.0	30/06/2014 12:16
3760	Consulta de datos de identidad	Estado servicio	Indicador		0.0	30/06/2014 12:16
3759	Consulta de datos de identidad	Estado servicio	Indicador		0.0	30/06/2014 12:14
3758	Consulta de datos de identidad	Estado servicio	Indicador		0.0	30/06/2014 12:12
3757	Consulta de datos de identidad	Estado servicio	Indicador		0.0	30/06/2014 12:10

Ilustración 83. Registros

Irán creándose nuevos registros según la frecuencia que tengan establecida en la configuración del componente. Cada dos minutos el sistema analiza que procedimientos tienen que ser ejecutados.

En la sección de registros podemos realizar búsquedas por distintos filtros para analizar más detalladamente y con más claridad los datos monitorizados. Por ejemplo podemos filtrar por componente “Consulta de datos de identidad”, indicador “Tiempo petición” en el día 16 de junio de 2014.

Id	Compoñente	Indicador	Tipo	Clave	Valor	Data
1275	Consulta de datos de identidad	Tiempo petición	Indicador		1.0	16/06/2014 16:34
1224	Consulta de datos de identidad	Tiempo petición	Indicador		7.0	16/06/2014 15:56
1207	Consulta de datos de identidad	Tiempo petición	Indicador		0.11	16/06/2014 15:50
1193	Consulta de datos de identidad	Tiempo petición	Indicador		0.88	16/06/2014 15:46
1182	Consulta de datos de identidad	Tiempo petición	Indicador		0.88	16/06/2014 15:44
1168	Consulta de datos de identidad	Tiempo petición	Indicador		0.88	16/06/2014 15:40
1085	Consulta de datos de identidad	Tiempo petición	Indicador		2.6	16/06/2014 15:30
1062	Consulta de datos de identidad	Tiempo petición	Indicador		2.6	16/06/2014 14:56

Mostrando dende 1 ata 8 de 8 resultados

Ilustración 84. Ejemplo búsqueda de registros

Por último, tenemos la posibilidad de exportar la búsqueda realizada a un fichero CSV, pulsando en el botón “Exportar CSV”.

6. REGLAS

Desde esta sección podemos crear reglas para los componentes y los indicadores, según las cuales, de cumplirse una repetición de registros con un determinado nivel, se envíen automáticamente avisos a los destinatarios que configuremos y se registren nuevas incidencias.

Un ejemplo de regla, podría ser que se avisase al equipo de mantenimiento de un servicio, mediante correo electrónico, en el caso de que el mismo registrase los 6 últimos indicadores amarillos en el indicador de “Estado servicio”.

Las reglas dejarán de enviar avisos y crear incidencias, mientras exista una incidencia abierta sin resolver para el mismo componente y nivel de indicador.

6.1 Crear una regla

Para crear una regla debemos ir a la sección “Crear” dentro de “Reglas” en el menú superior.

Xeral | Configuración | Servicios | Incidencias | Registros | **Reglas**

Buscar | **Crear**

Componente *
Selecione...

Indicador *
Selecione...

Nivel *
Selecione...

Max. Repeticións *
[]

Mensaxe *
[]

Listas Emails *
Selecione...

Listas Emails
Nome lista emails

Destinatarios *
[]

Destinatarios
Emails destinatarios

Ilustración 85. Creación de una regla

Para crear una regla, deberemos especificar por lo menos los siguientes datos:

- Componente
- Indicador
- Nivel
- Max. Repeticiones
- Mensaje
- Listas Emails o Destinatarios

Con la opción “Listas Emails” podemos seleccionar listas de correo de varios correos electrónicos que tengamos configuradas mientras que con la opción “Destinatarios” podremos incluir correos electrónicos de manera individual.

Listas Emails *

Selecione... Añadir lista

Selecione...
 eAdmin

Nome lista emails

Destinatarios *

administrador@admin.es Añadir destinatario

Destinatarios

Emails destinatarios

Gardar Cancelar

Ilustración 86. Destinatarios del aviso

Una vez cubiertos los datos necesarios pulsamos “Guardar” y ya estará la regla creada.

6.2 EDITAR UNA REGLA

Podemos editar una regla que ya esté creada, para modificar cualquier parámetro de la misma, como puede ser el mensaje de aviso, o añadir algún destinatario. Para hacerlo debemos ir a la sección “Buscar”.

Xeral | Configuración | Servicios | Incidencias | Rexistros | **Regras**

Buscar | Crear

Compoñente: Calquera | Indicador: Calquera | Nivel: Calquera Buscar

Compoñente	Indicador	Nivel	Repeticións	Mensaxe			
AUDITORIA ELIMINACIÓN BATCH	Funcionamento tarefa	●	2	O batch de eliminación de auditoría non funciona			
Consulta de bens inmobles	Estado servizo	●	3	O servizo consulta de bens inmobles encóntrase nun estado inconsistente			
Consulta de bens inmobles	Estado servizo	●	1	O servizo consulta de bens inmobles caiu			
Consulta de datos de identidade	Estado servizo	●	3	O servizo consulta de identidade caiu			
Alta en data na Seguridade Social	Tempo petición	●	5	O servizo de alta na Seguridade Social presenta un alto Tempo de petición			
SOLICITUDE RESPONSA BATCH	Funcionamento tarefa	●	3	A tarefa programada de Solicitud de Resposta non está funcionando correctamente			

Mostrando dende 1 ata 6 de 6 resultados

● Nivel roxo
 ● Nivel amarelo
 ● Nivel verde

Ilustración 87. Acceso a editar una regla

Localizada la regla pulsamos en “Editar” para poder modificar todos los parámetros de la regla.

Compoñente *
 Consulta de bens inmobles

Indicador *
 Estado servizo

Nivel *
 Vermello

Max. Repeticións *
 3

Mensaxe *
 O servizo consulta de bens inmobles encóntrase nun estado inconsistente.

Enviar a usuarios

Listas Emails *
 Seleccione... Añadir lista

Listas Emails

Nome lista emails

Destinatarios *
 Añadir destinatario

Destinatarios

Emails destinatarios
 sergio@gmail.com X

Gardar Cancelar

Ilustración 88. Editar una regla

Una vez realizados todos los cambios, hay que pulsar “Guardar”.

6.3 Eliminar una regla

Podemos eliminar cualquiera de las reglas creadas. De este modo dejarán de emitir avisos y registrar incidencias. Para hacerlo debemos ir a la sección buscar y pulsar el botón de “Eliminar”.

Compoñente	Indicador	Nivel	Repeticións	Mensaxe			
Consulta de bens inmobles	Estado servizo	●	1	O servizo consulta de bens inmobles caiu			
Consulta de bens inmobles	Estado servizo	●	3	O servizo consulta de bens inmobles encóntrase nun estado in...			

Ilustración 89. Eliminar regla

7. INCIDENCIAS

Como se explica en el apartado anterior, mediante las reglas además de avisos se generarán incidencias que quedarán registradas en esta sección, y que podrán ser atendidas por el administrador.

Xeral | Configuración | Servicios | **Incidentes** | Registros | Regras

Id

Nivel

Tipo

Compoñente

Indicador

Estado

Dende

Ata

Estado	Id	Compoñente	Indicador	Tipo	Nivel	Data	
✓	33	Consulta de datos de identidade	Estado servizo	Indicador	●	27/06/2014 14:30	
!	32	Consulta de bens inmobles	Estado servizo	Indicador	●	26/06/2014 19:50	
!	31	Consulta de bens inmobles	Estado servizo	Indicador	●	26/06/2014 19:35	
✓	28	Consulta de datos de identidade	Estado servizo	Indicador	●	24/06/2014 16:04	
✓	26	Consulta de datos de identidade	Estado servizo	Indicador	●	23/06/2014 00:57	
!	1	SOLICITUD RESPUESTA BATCH	Funcionamento tarefa	Indicador	●	22/06/2014 16:04	

Mostrando dende 1 ata 6 de 6 resultados

Sen resolver

Resolto

Nivel roxo

Nivel amarelo

Nivel verde

Ilustración 90. Ver incidencias

En el listado de incidencias que se nos presentará se dará completa información acerca de las incidencias.

- Estado de la incidencia: Sin resolver o resuelta.
- Componente en el que se detecto la incidencia.
- Indicador que generó la incidencia.
- Tipo: Puede ser incidencia de tipo Indicador (Establecido con las reglas) o de Nivel de servicio (predicción de degradación de servicio).
- Nivel de criticidad detectado.
- Fecha en la que se registró la incidencia.

Se permite también la búsqueda de incidencias por cualquiera de estos filtros.

En cualquier momento podemos exportar a un fichero CSV la consulta realizada.

7.1 Ver detalles de una incidencia

Si queremos ver la información de una incidencia de modo más detallado, tendremos que pulsar en el botón situado a la derecha de la misma.

Estado	Id	Compoñente	Indicador	Tipo	Nivel	Data	
✓	33	Consulta de datos de identidad	Estado servizo	Indicador	●	27/06/2014 14:30	
!	32	Consulta de bens inmuebles	Estado servizo	Indicador	●	26/06/2014 19:50	
!	31	Consulta de bens inmuebles	Estado servizo	Indicador	●	26/06/2014 19:35	
✓	28	Consulta de datos de identidad	Estado servizo	Indicador	●	24/06/2014 16:04	
✓	26	Consulta de datos de identidad	Estado servizo	Indicador	●	23/06/2014 00:57	
!	1	SOLICITUD RESPUESTA BATCH	Funcionamento tarefa	Indicador	●	22/06/2014 16:04	

Ilustración 91. Ver detalles incidencia

Clicando en el botón accedemos a los detalles de la incidencia.

Id
32

Compoñente
Consulta de bens inmuebles

Indicador
Estado servizo

Tipo
Indicador

Mensaxe
O servizo consulta de bens inmuebles está indispoñíbel nestes momentos

Estado
Sen Resolver

Data
26/06/2014 19:50

Últimos rexistros

Id	Compoñente	Indicador	Clave	Valor	Data
3716	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 15:00
3710	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 14:30
3705	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:50
3700	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:30
3695	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:15
3690	Consulta de bens inmuebles	Estado servizo		1.0	27/06/2014 12:00

[Ver máis rexistros](#)

[Volver](#) [Resolver incidencia](#)

Ilustración 92. Detalles incidencia

Desde los detalles de incidencia podremos ver más información adicional acerca de la incidencia. En la parte inferior podemos ver los 6 últimos registros del componente e indicador que generaron la incidencia, junto con los valores y la fecha y hora concretas en que se produjeron.

7.2 Resolver una incidencia

Desde los detalles de una incidencia, además de consultar la información, si esta está sin resolver, también tendremos acceso al botón “Resolver incidencia”.

3695	Consulta de bens inmuebles	Estado servizo	1.0	27/06/2014 12:15
3690	Consulta de bens inmuebles	Estado servizo	1.0	27/06/2014 12:00

Ver más rexistros

Volver Resolver incidencia

Ilustración 93. Resolver incidencia

El usuario podrá utilizar este botón cuando estime oportuno, por ejemplo si observa que el indicador que dio lugar a que se cumpliese la regla / incidencia, devuelve ya valores verdes (positivos). Pulsando este botón, la incidencia pasará a estado “Resuelta” y la regla podrá volver a generar avisos e incidencias en caso de que más tarde se vuelva a cumplir, ya que como ya se mencionó en la sección anterior, mientras exista una incidencia sin resolver para un mismo componente, indicador y nivel de monitorización, no se generarán nuevos avisos ni incidencias.

Al pulsar el botón “Resolver incidencia” tendremos que confirmar si queremos resolver.

¿Está seguro de que desexa marcar esta incidencia como resolta?

Aceptar Cancelar

Ilustración 94. Confirmar resolver incidencia

Al clicar en “Aceptar” la incidencia pasará a estado “Resuelta”, sin embargo quedará registrada y podrá ser consultada sin problemas.

Apéndice C: *Material presentado*

En el soporte digital adjunto a este documento se encuentra:

- Código fuente del Módulo de Monitorización.
- Scripts de creación de las tablas de base de datos del Módulo de Monitorización.
- Memoria en formato pdf.
- Planificación final del proyecto.
- Historial de riesgos.
- Videos del Módulo de Monitorización. Se adjuntan estos videos debido a que el código fuente adjuntado solo puede ser ejecutado en el entorno del cliente por motivos de dependencias, permisos y derechos. Además se adjunta el archivo *GuionVideos.rtf* donde se explica el contenido de cada uno de los vídeos.

