

INGENIERÍA INFORMÁTICA

MEMORIA DEL PROYECTO:

6152 : Solución de Data Quality en entornos analíticos

Memoria del Proyecto Fin de Carrera
de Ingeniería Informática
realizado por Alejandro Morales Núñez
y dirigido por Andreu Pérez Martínez
Bellaterra, 14 de Septiembre de 2015

El firmante, Andreu Pérez Martínez

Tutor/a de proyectos de la Escuela de Ingeniería de la UAB,

CERTIFICA:

Que el trabajo al que corresponde esta memoria ha estado realizado bajo su dirección por Alejandro Morales Núñez.

Y por tal que conste la firma presente.

Firmado: Andreu Pérez Martínez

Bellaterra, 14 de Septiembre de 2015

El firmante, Marc Ruiz Figueras
de la empresa, Indra Sistemas

CERTIFICA:

Que el trabajo al que corresponde esta memoria ha estado realizado en la empresa bajo la supervisión mediante convenio firmado con la Universidad Autònoma de Barcelona.

Así mismo, la empresa tiene conocimiento y da el visto bueno al contenido que se detalla en esta memoria.

Firmado:

Bellaterra, 14 de Septiembre de 2015

Agradecimientos

Llegados a este punto me gustaría dar las gracias a todas las personas que directa o indirectamente han formado parte de este proyecto, ya fuese aportando ideas, resolviendo problemas, leyendo o corrigiendo la memoria o simplemente dando apoyo en los momentos más pesados y duros durante toda la realización del proyecto.

Agradezco especialmente a mi tutor del proyecto Andreu Pérez Martínez su implicación, apoyo, paciencia, dirección del proyecto, optimismo y ganas por sacar el proyecto adelante.

Así mismo, agradezco a todos mis compañeros, jefes y desde un entonces amigos, por el apoyo dado en el proyecto tanto anímicamente como técnicamente y sobre todo, por ayudar a enfocar el proyecto o ampliar la visión durante su realización.

A todos los profesores i compañeros de la UAB por su ayuda, formación, enseñanzas, curiosidades y buenos momentos compartidos.

Finalmente, agradecer a mis padres y hermano toda su implicación y apoyo de seguir adelante en los momentos más tempranos de esta etapa, no solo durante la realización del proyecto si no durante estos 5 años y medio de carrera, por brindarme la posibilidad de estudiar una carrera y motivarme día a día por conseguir mis objetivos.

A todos ellos/as... ¡GRACIAS!

Índice de contenidos

1.	INTRODUCCIÓN	1
1.1.	Origen del proyecto	1
1.2.	Objetivos del proyecto.....	1
1.3.	Alcance del proyecto.....	2
1.4.	Motivaciones	3
1.5.	Estructura de la memoria.....	3
1.6.	Indra Sistemas S.A.....	4
1.7.	Metodología de desarrollo.....	5
2.	ESTUDIO DE VIABILIDAD	7
2.1.	Situación inicial	7
2.2.	Estado del arte.....	7
2.2.1.	Gartner	9
2.2.2.	Fortalezas y debilidades, comparativa de 3 empresas	14
2.3.	Estudio de viabilidad del proyecto	15
2.3.1.	Situación inicial	15
2.3.2.	Viabilidad legal.....	15
2.3.3.	Viabilidad técnica/económico-financiera.....	15
2.4.	Planificación inicial.....	17
2.5.	Planificación real.....	19
2.5.1.	Reuniones de seguimiento en Universidad	19
2.5.2.	Reuniones de seguimiento en empresa	19
2.5.3.	Planificación temporal real	21
3.	RECURSOS	23
3.1.	Recursos humanos.....	23

3.2.	Recursos hardware	23
3.3.	Recursos software.....	23
3.3.1.	Lenguajes de programación	23
3.4.	Base de datos.....	24
3.5.	Spring Framework.....	25
3.5.1.	Módulos de Spring.....	25
3.5.2.	Gestión de la web	28
3.5.	Maven.....	31
3.6.	Servidor Tomcat.....	32
3.7.	Eclipse.....	33
3.8.	Dropbox.....	34
4.	iQUALITY	35
4.1.	Interfaz web.....	39
4.2.	Perfiles de usuario	39
4.2.1.	Perfil Validación	40
4.2.2.	Perfil Funcional	41
4.2.3.	Perfil Administrador.....	41
4.3.	Capturas de pantalla de la aplicación	41
5.	CONCLUSIONES	45
5.1.	Conclusiones del proyecto	45
5.2.	Conclusiones personales.....	46
6.	Referencias.....	47
7.	Anexo	49
8.	Glosario	67

Índice de figuras

Ilustración 1: Metodología de desarrollo.....	6
Ilustración 2: iQuality	8
Ilustración 3: Cuadrante de Gartner	11
Ilustración 4: Tabla planificación inicial	17
Ilustración 5: Diagrama de Gantt planificación inicial.....	18
Ilustración 6: Método Scrum	19
Ilustración 7: Planificación real.....	21
Ilustración 8: Modelo MVC.....	28
Ilustración 9: Comunicación servlet.....	30
Ilustración 10: Flujo servlet	31
Ilustración 11: Flujo iQuality.....	35
Ilustración 12: Módulo Control Procesos de Carga.....	37
Ilustración 13: Módulo Diccionario de Conceptos	37
Ilustración 14: Módulo Gestión de Escenarios.....	38
Ilustración 15: Módulo Administración	38
Ilustración 16: Diseño interfaz.....	39
Ilustración 17: Perfiles de Usuario	40
Ilustración 18: Log In	41
Ilustración 19: Log In fail	42
Ilustración 20: iQuality tablas ejemplo	43
Ilustración 21: iQuality Main Page.....	43
Ilustración 22: iQuality form ejemplo	44

1. INTRODUCCIÓN

1.1. Origen del proyecto

La idea de este proyecto surge de la necesidad por parte de empresa de tener un producto/herramienta más optimizado, específico y escalable para los campos de BI/BA (Business Intelligence/Business Analytics) , que se pueda vender y a la vez que haga una funcionalidad interna en sus proyectos.

1.2. Objetivos del proyecto

Con la intención de garantizar y optimizar la calidad de los datos almacenados en sistemas analíticos y para poder ser un producto implantado en diferentes sistemas, se necesita llevar a cabo una serie de mejoras entre las que incluyen:

- Solución portable y escalable
- Solución eficiente
- Estudio de tecnologías a disposición para su utilización en el desarrollo
- Validación técnica de los datos origen.
- Certificación de la información de negocio
- Control del proceso de carga
- Replicabilidad
- Diccionario de conceptos de negocio
- Metadata del dato y trazabilidad física

El objetivo de este proyecto es desarrollar una herramienta en Java¹, con una estructura base que nos la define la actual aplicación que hay desarrollada en Apex² y está siendo usada en algunos proyectos de Indra. Como resultado final, conseguiremos una herramienta “gestionadora” de datos, con información proveniente de diferentes orígenes que nos ayudará a hacer un posterior análisis de ellos y sacar conclusiones

inteligentes para la generación de informes, toma de decisiones, etc ; con una velocidad óptima, fácil integración y portabilidad.

Por parte de la empresa se ha propuesto implementar la herramienta en un lenguaje más escalable como Java¹. La intención de ello es mejorar la eficiencia y crear un modelo tipo el cual pueda ser migrado a otros proyectos de forma sencilla y con la realización de cambios mínimos.

En los siguientes puntos veremos una estructuración generalizada de cómo se distribuirán las fases de este proyecto y en que se basará.

1.3. Alcance del proyecto

El alcance de este proyecto está muy sujeto a qué producto final busca Indra, debido a que el departamento donde se hará la implantación es una pequeña área de todo lo que llega a abarcar la división de BI³ y tenemos que mirar que esta implantación sea lo más viable y escalable posible a otros proyectos. Debido al tiempo, recursos e hitos que se requieren, estudiaremos un número limitado de empresas de la competencia que implantan en el mercado un software de características similares.

Por otro lado, la implantación de este sistema en otro proyecto es un proceso medianamente rápido pero complejo a la hora de instaurar la herramienta en otros proyectos, ya que cada uno tiene sus especificaciones. Aunque la empresa disponga de los recursos económicos necesarios para esta implantación, siempre se intentará que el coste sea lo más reducido posible.

Entre los beneficios que obtendríamos al tener una herramienta estándar en la planta de Business Intelligence está el motivo de que sería una herramienta propia y que en un futuro se podría llegar a exportar/vender. Actualmente, nos ofrece 'gasto 0' por ser una creación interna, es decir, desarrollada por un trabajador "corriente" con uso del hardware proporcionado por defecto en la empresa y dentro del salario habitual de un desarrollador en la empresa.

1.4. Motivaciones

Por parte personal, igual que comenté en su momento, a la hora de hacer la oferta al proyecto, supone una motivación extra para mí el hecho de crear algo útil, refiriéndome como útil a algo que puede tener un uso más allá de una práctica de laboratorio. Además, a la vez que me desarrollo en el campo profesional, trabajar sobre un producto que me puede dar un conocimiento más global de cómo Indra estructura y trata los datos.

La motivación por parte de la empresa surge de la necesidad de adaptar las herramientas actuales a las nuevas tecnologías que ofrecen una mayor integración, agilidad, estandarización y portabilidad entre los diferentes sistemas y aplicaciones. Como se ha dicho previamente, la solución está desarrollada con Apex² de Oracle⁴ y se desea migrar a una plataforma más abierta y estándar que permita en un futuro la posibilidad de ejecutarla en entornos móviles si fuese el caso.

En un futuro, si la implantación de esta nueva herramienta se llega a realizar, en un principio se puede creer que puede generar un prescindido de personal pero lo que realmente ofrece esta herramienta es una mejora de la velocidad de navegación, un mayor adaptabilidad y un uso más intuitivo juntamente con el lavado gráfico que se hace de ella.

1.5. Estructura de la memoria

La memoria del proyecto iQuality consta de 7 puntos claramente diferenciados.

Primeramente, hay una parte de introducción donde se definen los objetivos del proyecto, se hace un breve resumen introductorio al proyecto, se exponen las motivaciones personales y cómo se llevará a cabo el desarrollo del propio proyecto.

Seguidamente, hay un estudio de viabilidad en el que se detalla la situación a tratar y se detalla cómo será el proyecto y cómo será su funcionamiento. Además, se define a qué tipo de usuario va dirigido este proyecto, se marcan unos objetivos y se pone al lector en situación con un pequeño estudio del estado del arte.

A continuación se encuentra el apartado de recursos en que se hace una descripción más técnica de las tecnologías empleadas en la realización del proyecto, reuniones con los tutores y software/hardware empleado para el desarrollo.

En el siguiente apartado, se presenta el nuevo iQuality. Cómo funciona, su navegabilidad, se indicarán futuras implementaciones o posibles mejoras que aún no se llevarán a cabo en los próximos meses y por último, qué datos son usados, de qué procedimientos o funciones PL/SQL⁵ manejan los datos así como las tablas que los componen.

En el quinto apartado es donde se expone la adaptabilidad del proyecto con unas comparaciones entre la antigua y la nueva herramienta.

El sexto, se exponen las conclusiones del proyecto, intentando echar la vista atrás y valorar cuáles de los objetivos marcados en la especificación del proyecto que se han podido llevar a cabo y cuáles no, haciendo autocrítica del recorrido y desarrollo del proyecto. También se hace una valoración a nivel personal del nivel de satisfacción del proyecto y se valoran las posibles mejoras o ampliaciones a las que podría quedar sujeto el proyecto.

Como penúltimo apartado, tenemos donde aparecen las referencias bibliográficas, en su mayoría webs, que se han utilizado como documentación o ayuda durante el proyecto.

El octavo y último apartado son los apartados de Anexo de imágenes y glosario técnico, respectivamente, empleado en el desarrollo de la memoria.

1.6. Indra Sistemas S.A.

Indra es una empresa multinacional española que ofrece servicios de consultoría sobre transporte, defensa, energía, telecomunicaciones, servicios financieros; así como servicios al sector público. Es la segunda compañía europea por capitalización bursátil de su sector y la segunda empresa española que más invierte en I+D.

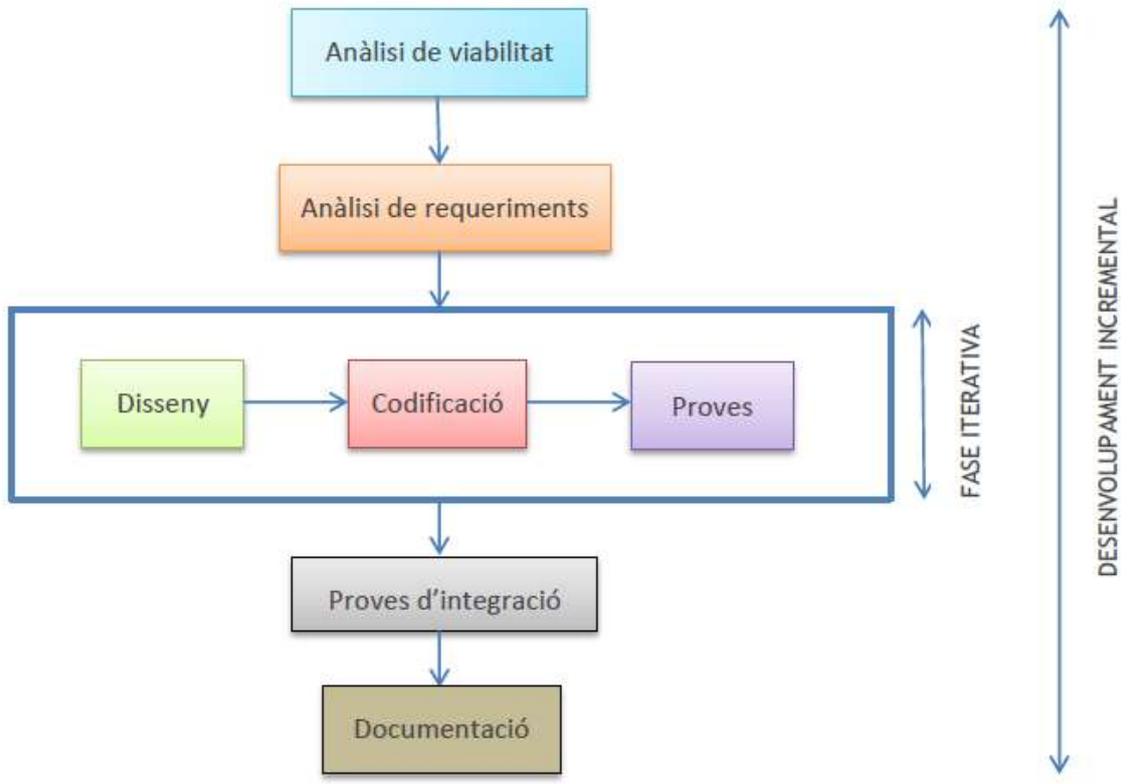
Indra Sistemas se constituyó por la fusión llevada a cabo en 1992 entre la compañía privada Ceselsa y el grupo público Inisel.

A partir de 1992 todas las sociedades existentes: INISEL, CESELSA, ERITEL y DISEL se integran en una única marca, quedando el grupo estructurado en cuatro áreas de negocio (subgrupos):

- Defensa y Tecnologías Duales: Indra DTD, hasta entonces encabezada por CESELSA. Formaban también parte de ella ENOSA, Gyconsa, ELT, EMAC, SAES, Sainsel, AISA, GDI y ASDL.
- Automatización, Control y Comunicaciones: Indra SCA, hasta entonces encabezada por DISEL. Electrónica Ensa y Tesis formaban también parte de esta área de negocio.
- Consultoría y Servicios informáticos: Indra SSI, hasta entonces encabezada por Eritel y encuadrando también a Ceninsa.
- Espacio: Indra Espacio, formada por la antigua Inisel Espacio.

1.7. Metodología de desarrollo

Desde un inicio se llevaron a cabo varias reuniones para marcar objetivos, acotar el problema, investigar las tecnologías a emplear, antecedentes, dividir el proyecto en módulos y planificar los mismos. Una vez llegados a este punto, se llevó a cabo un análisis de los requerimientos funcionales de los módulos en cuestión. Una vez se definieron estos requerimientos, aunque siendo conscientes de que con la evolución del proyecto estos requerimientos cambiarían y evolucionarían, se pasó a una etapa iterativa de diseño, codificación y pruebas. En este punto cobra importancia la metodología empleada durante el proyecto de reuniones periódicas en las que se hacía un repaso del trabajo individual realizado durante la semana o quincena, se exponían los problemas ocurridos durante la misma, se planificaba el trabajo a llevar a cabo durante el período de tiempo hasta la siguiente reunión y se realizaban las posibles pruebas de integración correspondientes entre módulos. Durante estas reuniones periódicas, surgían problemas, discusiones y cambios en los requerimientos que obligaban a iterar en el diseño, la codificación y las pruebas.



Il·lustració 1: Metodologia de desenvolupament

2. ESTUDIO DE VIABILIDAD

2.1. Situación inicial

En la situación actual, la empresa está volviendo a vivir un proceso de expansión y evolución, teniendo así más datos que manejar. De ahí surge la necesidad de implantar un Data Quality para hacer frente a la gestión de los datos. Así entonces, se propone hacer un estudio de una herramienta que nos pueda evitar el “descontrol” de datos que hay actualmente.

En todo esto, se tiene que tener en cuenta que este nuevo software impactará, muy seguramente, en la gestión de un grupo de trabajo, pero esto sería tema de recursos humanos que a nosotros no nos incumbe por el momento.

Una solución muy similar a iQuality la podemos encontrar instalada en el sector financiero, en empresas como La Caixa, así también, en la Administración pública y el sector de Industria.

2.2. Estado del arte

En una multinacional como Indra, con información en exceso, con duplicidad necesaria en caso de fallos para mantener la correcta decodificación de los datos, con duplicidad innecesaria por, mayormente, olvido de ella, etc; es imprescindible un mecanismo de gestión de los datos como el que buscamos.

En la gestión empresarial actualmente el dato inteligente que nos dé información de cómo obtener un beneficio y que nos ayude a tomar decisiones importantes y económicamente factibles a partir de la información de los sistemas, es una prioridad, y para ello hay que dotar a los departamentos de herramientas de BI (Business Intelligence) y BA (Business Analytics). El objetivo de estas herramientas es transformar el dato desordenado y desestructurado en información estructurada e inteligente.

El punto en el que nosotros nos encontramos es, en un primer momento, en el de desconocimiento parcial de la herramienta, es decir, desconocemos cual o cuales son los módulos concretos que se quieren mejorar o desarrollar como nuevos. A cambio, sabemos por parte universitaria la base de cómo funciona y cómo se estructura una base de datos en SQL⁶. Por parte de la empresa conocemos que APEX² es una herramienta RAD⁷ que se ejecuta con una base de datos Oracle. Permite desarrollar prototipos de aplicaciones WEB de forma segura y rápida.

En cuanto a la herramienta iQuality se define por sí misma con la imagen que hay a continuación:



Ilustración 2: iQuality

Hablando de la disponibilidad hardware y software haremos uso del portátil entregado por Indra Sistemas (HP 68ICF Ver. F.50) y el portátil propio (ACER Aspire 5738DZG), en caso de ser necesario algún desarrollo externo a la empresa. Por parte software, tendremos acceso en modo administrador a las herramientas y BBDD⁸ necesarias.

Es importante asegurar la calidad de los datos en soluciones analíticas porque principalmente, la pérdida de ellos puede suponer pérdidas millonarias. En cambio el aseguramiento de calidad es proporcional a un aval fundamentado sobre la credibilidad de la información o producto

generado.

Para el aseguramiento de la calidad se necesita la integración y el completo control de todos los elementos dentro de un área específica de operación para que ninguno esté subordinado a otro. Estos elementos abarcan aspectos tales como administración finanzas, ventas, comercialización, diseño, compras, producción, instalación contratación.

La versión que desarrollaré será la primera porque como he mencionado en los objetivos, no hay una versión estándar/base, para ser implantada en diferentes proyectos. Por lo tanto, tomando como referencia productos existentes que definiremos a continuación y teniendo en cuenta las necesidades generales en la gestión de datos en Indra, crearemos esta versión 1.0.

iQuality, que será la herramienta que desarrollaré como nueva y mejoraré, ahora mismo Indra la tiene implantada en varios clientes de forma no estandarizada, y en fase de implantación en otros; además, se observa mucho interés en muchos clientes para alguna de las funcionalidades de iQuality.

En el fondo no será tan potente como otras existentes pero ofrecerá particularidades que beneficia el uso y extracción de la información al mismo tiempo que estudiaremos y tendremos que modificar la interacción entre sistema gestor-usuario .

Ahora surgirán las preguntas, Qué herramientas son las principales en el mercado? Qué es lo que se evalúa en data quality y que parámetros sigue el cuadrante de Gartner?

2.2.1. Gartner

Gartner Inc. es una empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, Connecticut, Estados Unidos. Hasta 2001 era conocida como Gartner Group. La empresa se concentra en la investigación, programas ejecutivos, consultas y eventos.

Los criterios de inclusión o exclusión para ser incluido en el 'Magic Quadrant', siguen el siguiente criterio:

- Deben ofrecer independiente empaquetado entre herramientas de software y servicios basados en la nube que se colocan, comercializados y vendidos específicamente para aplicaciones de calidad de datos de uso general.
- Deben ofrecer una funcionalidad que se ocupa, como mínimo, de perfiles, análisis sintáctico, estandarización / limpieza y el seguimiento. Los vendedores que ofrecen una funcionalidad limitada (por ejemplo, que sólo admiten la limpieza y validación de direcciones, o que tienen que ver sólo con el juego) se excluyen porque no ofrecen suites completas de herramientas de calidad de datos. En concreto, los vendedores deben ofrecer todas las características siguientes:
 - Perfiles y visualización
 - Análisis
 - Estandarización y limpieza
 - Monitorización
 - Matching
- Deben apoyar esta funcionalidad con capacidades envasados para los datos en más de un idioma y de más de un país.
- Deben apoyar esta funcionalidad tanto en programada (por lotes) y los modos interactivos (en tiempo real).
- Deben apoyar el despliegue a gran escala a través de arquitecturas de tiempo de ejecución basadas en servidor que pueden apoyar a los usuarios y aplicaciones simultáneas.
- Deben mantener una base instalada de al menos 100 de producción, mantenimiento / clientes suscriptores que pagan por el producto(s) de calidad de datos que satisfacen los criterios funcionales anteriores. La base de clientes de producción debe incluir a los clientes en más de una región (América del Norte, América Latina, EMEA y Asia / Pacífico).

- Deben ser capaces de ofrecer a los clientes de referencia que demuestren el uso multidominio y / o multiproyecto del producto (s) que satisfacen los criterios funcionales anteriores.



Ilustración 3: Cuadrante de Gartner

Dada la magnitud y la complejidad del paisaje de datos, a través de organizaciones de todos los tamaños y en todas las industrias, las herramientas para ayudar a automatizar los elementos clave de esta disciplina siguen atrayendo más interés y crecen en valor. Como tal, el mercado de herramientas de calidad de datos continúa mostrando un crecimiento sustancial, mientras que también exhibe la innovación y el cambio.

Este mercado incluye proveedores que ofrecen productos de software independientes para hacer frente a los requisitos funcionales básicos de la disciplina, que son:

- Perfil de datos y medición de la calidad de datos:

El análisis de los datos para capturar estadísticas (metadatos) que permiten conocer la calidad de los datos y ayudan a identificar los problemas de calidad de datos.

- El análisis y la normalización:

La descomposición de los campos de texto en sus componentes y el formato de los valores en los diseños coherentes, basados en estándares de la industria, las normas locales (por ejemplo, normas de autoridad postales para datos de direcciones), reglas de negocio definidas por el usuario, y bases de conocimiento de valores y patrones.

- "Limpieza" generalizada:

La modificación de los valores de datos para cumplir con las restricciones de dominio, las restricciones de integridad u otras reglas de negocio que definen cuando la calidad de los datos es suficiente para una organización.

- Maridaje:

La identificación, la vinculación o la fusión de las entradas relacionadas dentro o a través de conjuntos de datos.

- Monitoreo:

La implementación de controles para garantizar que los datos continúa ajustándose a las reglas de negocio que definen la calidad de datos de una organización.

- Resolución de problemas y el flujo de trabajo:

La identificación, poner en cuarentena, la escalada y la resolución de problemas de calidad de datos a través de procesos e interfaces que permiten la colaboración con las funciones clave, como mayordomo de datos.

- Enriquecimiento:

El aumento del valor de los datos en poder internamente añadiendo atributos relacionados de fuentes externas (por ejemplo, atributos demográficos de los consumidores y los descriptores geográficos).

Además, las herramientas de calidad de datos proporcionan una gama de habilidades funcionales relacionados que no son exclusivas de este mercado, sino que se requieren para ejecutar muchas de las funciones básicas de calidad de los datos, o para aplicaciones específicas de calidad de datos:

Conectividad / adaptadores confieren la capacidad de interactuar con una gama de diferentes tipos de estructura de datos. Apoyo específico en la materia proporciona capacidades de normalización para áreas específicas de los interesados. El apoyo internacional ofrece la posibilidad de ofrecer operaciones de calidad de datos relevantes a nivel mundial (como el manejo de datos en varios idiomas y sistemas de escritura). Gestión de metadatos permiten la capacidad de capturar, conciliar e interoperar metadatos relacionados con el proceso de calidad de los datos.

Habilidades entorno de configuración permiten la creación, gestión y despliegue de normas de calidad de datos. Las operaciones y recursos de administración de apoyar el control, gestión, auditoría y control de los procesos de calidad de datos. Habilitación de servicios ofrece características y apoyo orientadas a servicios para los despliegues de arquitectura orientada a servicios (SOA). Opciones de implementación alternativos ofrecen capacidades para implementar algunas o todas las funciones y / o servicios de calidad de datos más allá de implementaciones locales (por ejemplo, a través de la nube).

Las herramientas proporcionadas por los vendedores en este mercado son generalmente utilizados por las organizaciones para la implementación interna en su infraestructura de TI⁹. Los utilizan para apoyar directamente

a los procesos transaccionales que requieren operaciones de calidad de datos y para que el personal en puestos orientados a la calidad de datos, tales como los administradores de datos, para llevar a cabo obras de mejora de calidad de datos. Fuera del establecimiento soluciones, en forma de ofertas de calidad de datos alojados, modelos de entrega SaaS¹⁰ y servicios en la nube, seguirá evolucionando y creciendo en popularidad.

2.2.2. Fortalezas y debilidades, comparativa de 3 empresas

A partir de los datos obtenidos por parte de Gartner, expondremos los puntos fuertes y débiles de de 3 empresas importantes en este campo como son IBM, Informática y SAP.

2.2.2.1. IBM

Fortalezas

- La profundidad y amplitud de uso
- Integridad, escalabilidad, integración y rendimiento
- Mind acción y presencia en el mercado
- Información mejorada funcionalidad de gobierno

Debilidades

- Modelo de precios
- Más tiempo para desplegar herramientas
- La complejidad de la implementación del producto
- Insuficiente énfasis en modelos de implementación alternativa

2.2.2.2. Informática

Fortalezas

- La profundidad y la amplitud de las capacidades y el uso
- Negocio orientado al apoyo y la facilidad de uso
- Presencia en el mercado y la marca de la conciencia
- La expansión de capacidades a través de la adquisición

Debilidades

- Modelo de costos
- La integración de los componentes del producto
- Rendimiento en implementaciones de nubes
- La alta demanda de habilidades técnicas

2.2.2.3. SAP

Fortalezas

- Capacidades de Custodia
- Presencia en el mercado y el crecimiento
- Nivel de integración con otras aplicaciones y tecnologías SAP
- Soporte para múltiples dominios de datos

Debilidades

- Soporte de producto y actualizaciones de versión
- Documentación del producto
- Tiempo de salida al valor más larga
- Pricing Model

2.3. Estudio de viabilidad del proyecto

2.3.1. Situación inicial

En empresas existentes como La Caixa, una solución muy similar ya se encuentra implantada en proyectos del entorno Financiero, Administraciones Públicas e Industria.

2.3.2. Viabilidad legal

En cuanto a vías legales, por parte de Universidad-Alumno-Empresa, ya está firmado y validado el “Vist-i-plau” que da derechos de exposición, entre otros, a la universidad. Por tanto, ambas partes tienen claros sus roles y funciones. Para el funcionamiento normal de Indra, este proyecto no supondrá un cambio en aspectos legales más que de optimización, actualización y re-implantación de este software.

2.3.3. Viabilidad técnica/económico-financiera

La disponibilidad con la que nos encontraremos a la hora de resolver dudas y orientarnos en el proyecto será casi total por parte de Indra. Podremos hacer uso del email institucional, el personal disponible y a cargo en el departamento en el cual me encuentro, y como costes extra, se me proporcionará un portátil (HP 68ICF Ver. F.50) y horas de tutoría con los encargados del proyecto por parte de empresa. A nivel presupuestario no conocemos los costes reales por privacidad de la empresa.

Antes de empezar con el desarrollo, tenemos que sentarnos para definir la tecnología (en principio Java), así como los frameworks¹¹ que usaremos. Indra tiene uno propio (idynamics), y deberíamos ver si se puede aprovechar.

Una de las cuestiones que puede surgir es, ¿por qué Java y no C#¹²? En este caso, la respuesta es sencilla. Una empresa se centra básicamente en procesos productivos dedicados a sacar al mercado una tipología de producto (fabricación, producción, calidad, etc) y, en procesos de gestión como en cualquier otro ámbito, la demanda es la que tendrá la última palabra, y por tanto quien acabará determinando el precio así como el tipo de producto que mejor se adapta a sus necesidades. Por lo tanto, por mucho que se piense que C# puede suponer una mejora en cuanto a gestión de memoria en comparación a la gestión que hace JVM (Java Virtual Machine), nosotros tenemos que acotarnos a lo demandado, y en un principio desarrollaremos el proyecto con el lenguaje Java.

2.4. Planificación inicial

Basándonos en la parrilla y el diagrama de Gantt que adjuntamos a continuación, ésta era la planificación que se hizo inicialmente. Se estimó la entrega final para Julio, porque en periodo de verano el contacto con los tutores del proyecto será escaso por motivo de vacaciones y, en septiembre, sería la solicitud de lectura y presentación del proyecto.

Tareas	Fechas previstas	
Análisis de requerimientos	09/03/2015	al
	20/04/2015	
Iteraciones semanales/quincenales de entregas de funcionalidades (Documentación, redefinición objetivos, modelización, revisión, ...)	Marzo/2015	a
	Julio/2015	
Diseño de la solución (Implementación, modulado, revisión final, ...)	01/05/2015	al
	30/06/2015	
Confección detalles finales del proyecto	06/07/2015	al
	17/07/2015	
Entrega memoria final PFC	20/07/2015	al
	31/07/2015	
Entrega final PFC	16/09/2015	al
	22/09/2015	

Ilustración 4: Tabla planificación inicial

A lo largo de todo el PFC se planeó tener reuniones varias con los tutores responsables del proyecto para mantener un historial de cómo está yendo la progresión. Por parte de la universidad, a través de reuniones periódicas de seguimiento, y por parte de empresa, llegando a los hitos que vayamos marcando con periodos semanales o quincenales. Esta idea se fue cumpliendo desde el primer momento y sirvió mucho para mantener un feedback con todos los responsables del proyecto.

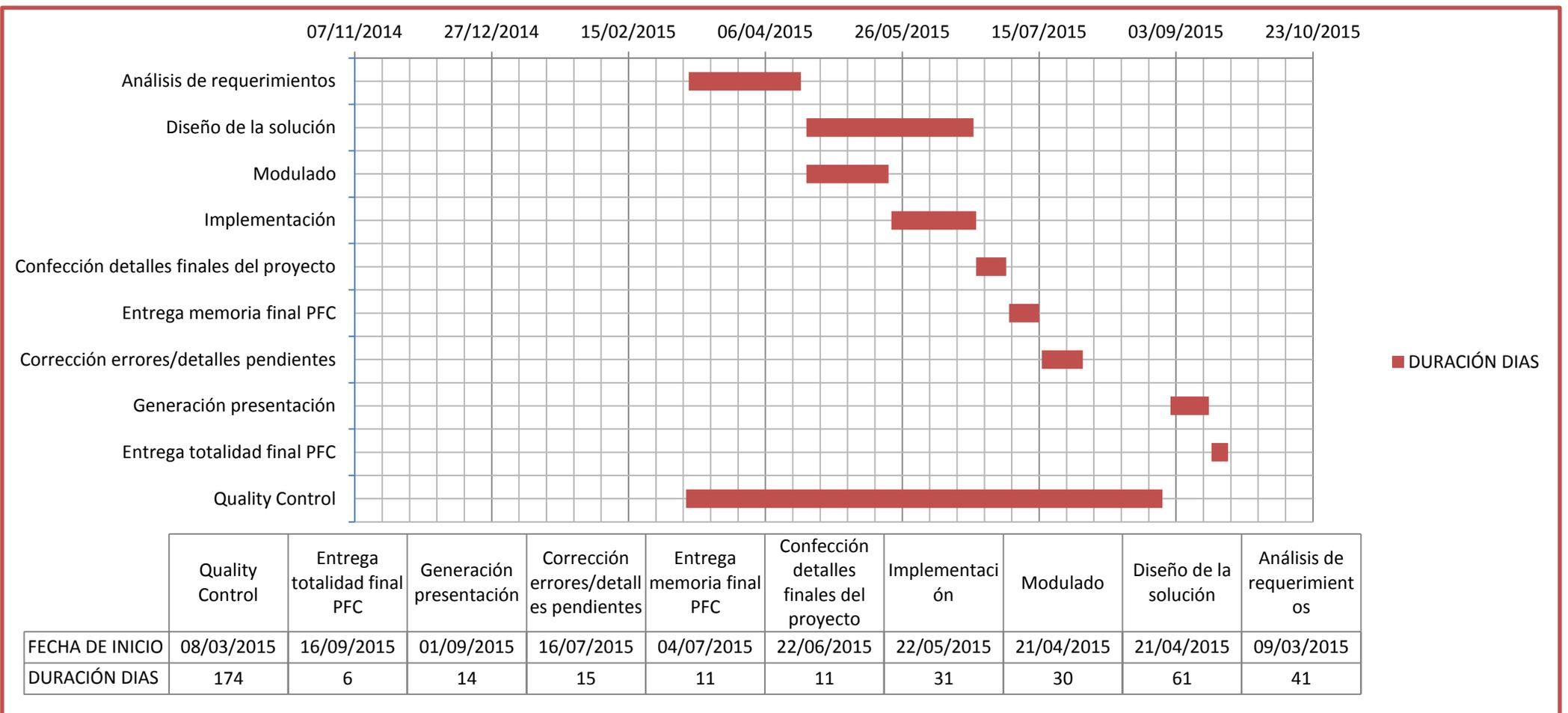


Ilustración 5: Diagrama de Gantt planificación inicial

2.5. Planificación real

2.5.1. Reuniones de seguimiento en Universidad

En las reuniones en la universidad se analizaba principalmente el progreso del proyecto y se trataban aquellos temas que requerían de una especial atención para ser resueltos por generar dificultades en el normal transcurso del proyecto. A parte de los temas técnicos y aspectos de planificación, se han estado tratando también aspectos de tipo principalmente organizativos que han ido surgiendo en el proyecto y que serán muy útiles, en un futuro, para poder prever problemas derivados de un cuello de botella como el que ha surgido en este proyecto. El estancamiento que ha habido en el desarrollo se puede ver en las actas de reunión adjuntadas en el anexo de este documento, apartado Actas de reunión.

2.5.2. Reuniones de seguimiento en empresa

De manera paralela a las reuniones de seguimiento por parte de universidad se propuso hacer un seguimiento semanal o quincenal de los hitos que había que conseguir a nivel de empresa. Este método de trabajo se llamada Scrum y consiste en:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto-organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.

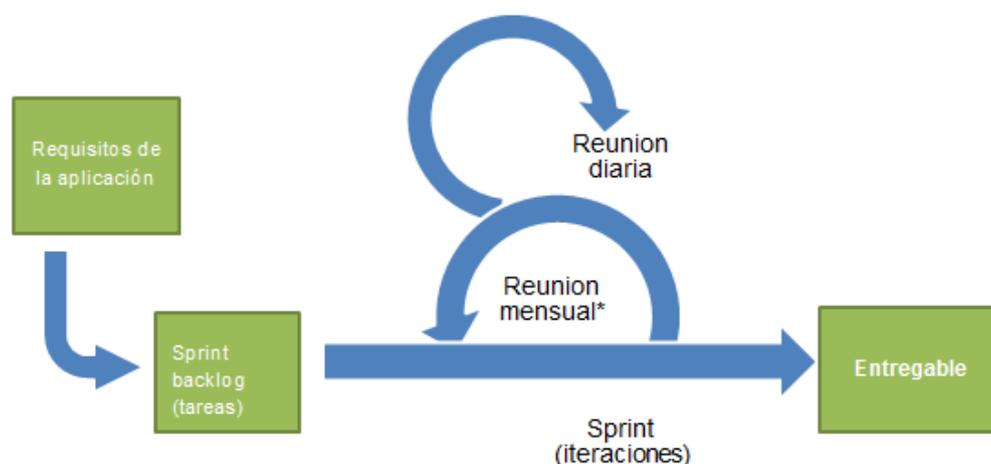


Ilustración 6: Método Scrum

Las primeras reuniones del proyecto se basaron en una serie de brainstorms donde tanto el tutor del PFC por parte de empresa como los compañeros de departamento exponían posibles ideas que había que desarrollar sobre la interfaz, que iba a ser necesario conservar y que no, que extras se le podrían incluir a la nueva herramienta, etc. Así se iba cogiendo nota para empezar a dar forma a lo que serían las pantallas principales y se concretaba una nueva reunión para próximas fechas.

Una vez superada esta fase inicial de recopilación de ideas, se llevo a cabo una serie de especificaciones de requerimientos para los diferentes módulos que aparentemente iban a conservarse de la herramienta anterior. Una vez hecho el estudio de la herramienta actual por parte del proyectista, se proponía que era lo que iba a ser creado o se enseñaba a modo esquemático como sería, por ejemplo, a nivel visual una nueva tabla y que “funcionalidades” tendría.

Llegados a este punto, se propuso que tipo de controlador sería el que gestionaría los datos de la aplicación. Se tuvo en cuenta que el lenguaje sobre el cual se quería trabajar era Java, por lo tanto, por decisión técnica de uno de mis compañeros, se propuso implementar este back-end¹³ con el framework¹¹ Spring-MVC¹⁴.

Una vez definidas todas las ideas se quedó que por carga de trabajo ajeno al proyecto, el proyectista sería el que se iría encargando de convocar para estas reuniones, dejando de ser semanales por falta de entregas en las fechas señaladas debido al cúmulo de errores que se arrastraban por falta de conocimiento en los lenguajes empleados para el proyecto. En las próximas reuniones tratamos de ver que tal iba el avance de la interfaz gráfica donde se exponía qué problemas surgían y me recomendaban como solucionarlos consultando foros online, como el ya más que conocido stackoverflow, o tomando webs de ejemplo de las cuales poder tomar como ejemplo para poder copiar su estructura y diseño.

Una vez metido de lleno en la fase de implementación y teniendo un esqueleto inicial “semi-aceptado” de las vistas web (obviamente con

mejoras a llevar a cabo en un futuro) se comenzó el desarrollo del controlador, el cual ha supuesto un verdadero muro imposible de derribar debido a que la formación sobre este framework¹¹ es inexistente y no ha habido posibilidad de ello. Como consecuencia, el desarrollo del back-end¹³ no tiene una estructura sólida y escalable como se deseaba en un principio, haciendo de esta parte el punto más negativo de todo el proyecto. Este problema ha ido derivando a un desentendimiento bidireccional entre proyectista y tutor PFC en empresa.

Ya en la recta final, teniendo todos los problemas mencionados en el punto anterior, la propuesta inicial de objetivos cambió completamente, dejando a un lado la inteligencia del back-end¹³ para hacer un desarrollador experimental y centrarse en el campo del desarrollo de las vistas. Básicamente, como podemos observar aquí, lo que más he echado en falta en este proyecto ha sido el apoyo técnico para hacer un buen desarrollo de la inteligencia y visualización de la herramienta.

2.5.3. Planificación temporal real

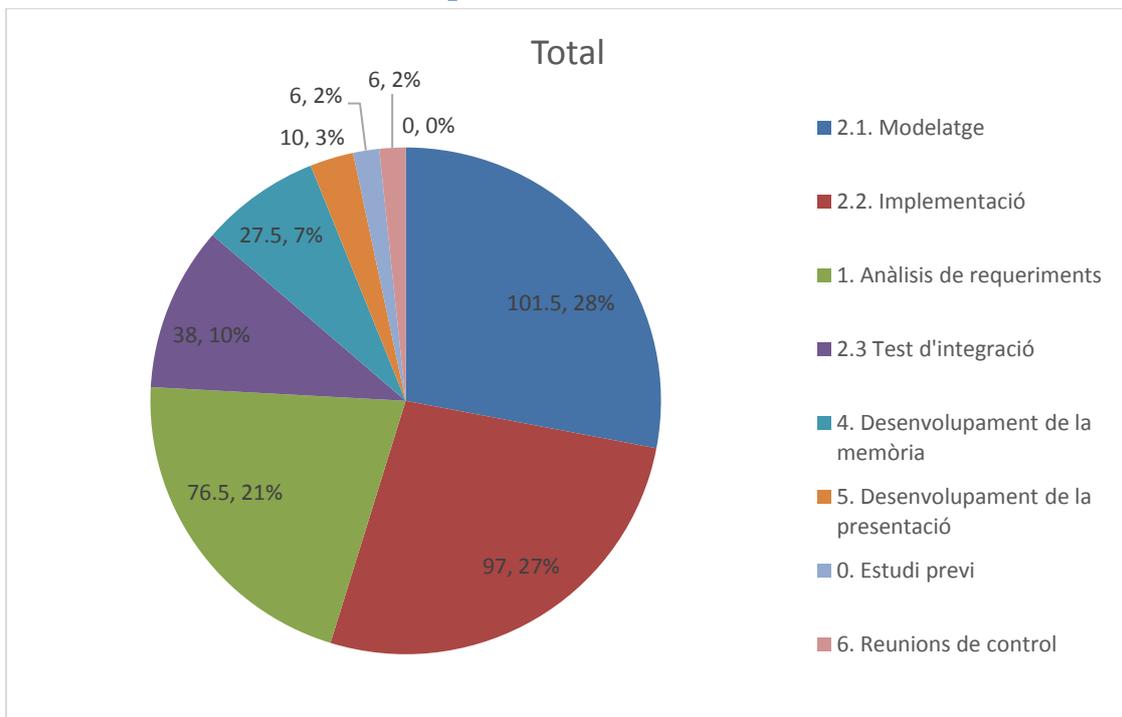


Ilustración 7: Planificación real

A destacar están las fases de Modelado, Implementación y Análisis de requisitos, todas ellas puntos cruciales en el desarrollo de cualquier

proyecto. Normalmente, un análisis de requerimientos se debe hacer con más profundidad, suele ocupar el mayor porcentaje de dedicación, ya que esto nos evita pérdidas en caso de abandonar un proyecto antes de empezarlo y, además, nos estructura de una forma más clara el alcance y sus objetivos.

Como conclusión en este apartado, vemos que el modelado y la implementación ha sido lo que más tiempo ha consumido debido a los problemas técnicos especificados en las conclusiones.

3. RECURSOS

3.1. Recursos humanos

El recurso humano del que se ha dispuesto a parte de las colaboraciones extraordinarias para el proyecto ha sido el equipo del departamento de VidaCaixa el cual ha conformado un total de 4 personas. Dando apoyo técnico Elena Martin-Tereso, Elena Vázquez, Marc Ruiz.

Por parte universitaria mi tutor, Andreu Pérez Martínez, se ha encargado de orientarme en cómo trabajar y encarar un proyecto en el mundo laboral, a la vez que ampliaba la visión que se le da a un proyecto final de carrera.

3.2. Recursos hardware

En cuanto a recursos hardware, se ha hecho servir un HP EliteBook 8470p x64bits con Procesador Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz, 2601 Mhz, 2 procesadores principales y 4 procesadores lógicos. Considerado un PC estándar para la empresa y el cual es proporcionado a los trabajadores de ésta.

3.3. Recursos software

3.3.1. Lenguajes de programación

Para llevar a cabo este proyecto se ha hecho el desarrollo en lenguaje Java con la ayuda del framework¹¹ Spring¹⁶ y Maven¹⁷. Antes de esta elección se había sopesado la implementación en C++¹⁸.

A la hora de determinar que lenguaje utilizar es necesario valorar el compromiso entre portabilidad y potencia. Uno de los principales objetivos del proyecto es que la aplicación sea lo máximo portable posible, de esta manera los tres lenguajes que inicialmente se sugirieron (Java¹, C++¹⁸ y Python¹⁹).

Python fue el primero en descartar por ser un lenguaje interpretado que no ofrece tipado estático, lo cual nos evitaría arrastrar errores de codificación accidentalmente. Por lo que hace a los otros dos candidatos, se valoró la gran potencia de

C++¹⁸ en frente a la gran portabilidad de Java, añadida al hecho de que se tiene que implantar en otra máquinas y se busca en un futuro un desarrollo en terminales Android para su control remoto en tiempo real. La principal idea de la portabilidad hizo decantar la decisión del lenguaje hacia Java, por el objetivo futuro de unificar toda la parte de la programación de los dispositivos bajo un mismo lenguaje, siendo Android la aplicación móvil de los terminales que puede proporcionar la empresa a sus trabajadores.

Un segundo punto muy importante del proyecto es la plataforma en que se tendrá que desarrollar, la interfaz²⁰ gráfica web. El control sobre esta herramienta se hará vía online de manera que sea accesible desde cualquier lugar con conexión a la red interna de la empresa. Por lo tanto, hubo 2 propuestas: HTML5²¹ y JavaFX²², ambas escogidas por su portabilidad y potencia. La decisión final se basó en un criterio de estandarización y portabilidad, por tanto, se escogió HTML5²¹ en frente de JavaFX²².

Finalmente, un punto muy importante en todo el proyecto es la interacción entre HTML5²¹ en este caso y la base de datos. Por eso, se decidió utilizar el framework¹¹ que ya se estaba utilizando en otros proyectos dentro de la empresa para la comunicación entre las vistas hechas en HTML5²¹ (ficheros jsp²³) y los controladores (clases Java¹ con Spring¹⁶ framework¹¹).

3.4. Base de datos

Para almacenar la información previamente disponemos de una base de datos Oracle. En cuanto a la posibilidad de elección de está, no ha sido posible porque la migración de un tipo de base de datos a otra, está fuera del alcance de este proyecto. Esta base de datos comparte espacio con otros sistemas. La capacidad total que ocupa es 60597,3125 MB y el reparto se muestra en Anexo.

Como características de la base de datos también tenemos:

- La localización física depende del cliente

- Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
- PL/SQL⁵ Release 11.2.0.4.0 - Production
- CORE 11.2.0.4.0 Production
- TNS for Linux: Version 11.2.0.4.0 - Production
- NLSRTL Version 11.2.0.4.0 – Production

3.5. Spring Framework

Spring es un framework¹¹ para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

La primera versión fue escrita por Rod Johnson, quien lo lanzó junto a la publicación de su libro Expert One-on-One J2EE Design and Development (Wrox Press, octubre 2002). El framework fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003. El primer gran lanzamiento fue la versión 1.0, que apareció en marzo de 2004 y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005.

Si bien las características fundamentales de Spring¹⁶ Framework¹¹ pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar de que no impone ningún modelo de programación en particular, este framework¹¹ se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (Enterprise JavaBean).

3.5.1. Módulos de Spring

Spring Framework comprende diversos módulos que proveen un rango de servicios:

- Contenedor de inversión de control: permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java, se lleva a cabo principalmente a través de la inyección de dependencias.
- Programación orientada a aspectos: habilita la implementación de rutinas transversales.

- Acceso a datos: se trabaja con RDBMS²⁴ en la plataforma java, usando Java Database Connectivity y herramientas de Mapeo objeto relacional con bases de datos NoSQL.
- Gestión de transacciones: unifica distintas APIs²⁵ de gestión y coordina las transacciones para los objetos Java.
- Modelo vista controlador: Un framework¹¹ basado en HTTP⁶³ y servlets⁴⁴, que provee herramientas para la extensión y personalización de aplicaciones web y servicios web REST²⁶.
- Framework¹¹ de acceso remoto: Permite la importación y exportación estilo RPC²⁷, de objetos Java a través de redes que soporten RMI²⁸, CORBA²⁹ y protocolos basados en HTTP⁶³ incluyendo servicios web (SOAP³⁰).
- Convención sobre Configuración: el módulo Spring Roo ofrece una solución rápida para el desarrollo de aplicaciones basadas en Spring Framework, privilegiando la simplicidad sin perder flexibilidad.
- Procesamiento por lotes: un framework para procesamiento de mucho volumen que como características incluye funciones de registro/trazado, manejo de transacciones, estadísticas de procesamiento de tareas, reinicio de tareas, y manejo de recursos.
- Autenticación y Autorización: procesos de seguridad configurables que soportan un rango de estándares, protocolos, herramientas y prácticas a través del subproyecto Spring Security (antiguamente Acegi³¹).
- Administración Remota: Configuración de visibilidad y gestión de objetos Java para la configuración local o remota vía JMX³².
- Mensajes: Registro configurable de objetos receptores de mensajes, para el consumo transparente desde la a través de JMS³³, una mejora del envío de mensajes sobre las API²⁵ JMS³³ estándar.
- Testing: Soporte de clases para desarrollo de unidades de prueba e integración.

Una de las piezas clave de Spring Framework es su Contenedor de inversión de control (IoC³⁶), que proporciona una forma consistente de configuración y administración de objetos Java. El Contenedor se encarga de gestionar los ciclos de vida de objetos de los objetos específicos: la

creación de estos objetos, llamando a sus métodos de inicialización, y configurando estos objetos cableándolos juntos.

Los objetos creados por el Contenedor también se denominan objetos gestionados o beans. El Contenedor se puede configurar mediante la carga de archivos XML³⁴ o la detección de anotaciones Java específicas sobre la configuración de las clases. Estas fuentes de datos contienen las definiciones que proporcionan la información necesaria para la creación de las beans.

Los objetos pueden ser obtenidos por cualquiera de los medios de dependencia de búsqueda o dependencia de inyección. Dependencia de búsqueda es un modelo donde se pide al objeto contenedor un objeto con un nombre específico o de un tipo específico. Dependencia de inyección es un modelo en el que el contenedor pasa objetos por nombre a otros objetos, ya sea a través de métodos constructores, propiedades, o métodos de la fábrica.

En muchos casos cuando se utilizan otras partes del Spring Framework no necesita utilizar el Contenedor, aunque probablemente su uso le permita hacer una aplicación más fácil de configurar y personalizar. El Contenedor de Spring le proporciona un mecanismo consistente para configurar las aplicaciones, y se integra con casi todos los entornos Java, desde aplicaciones de pequeñas a grandes aplicaciones empresariales.

Algunos critican al Spring¹⁶ Framework¹¹ por no cumplir los estándares. Sin embargo, SpringSource no ve el cumplimiento EJB 3 como un objetivo importante, y afirma que el Spring Framework y el contenedor permiten modelos de programación más potentes. No creas un objeto, sino describes la forma en que deben crearse, definiéndolo en el archivo de configuración de Spring. No llamas a los servicios y componentes, sino dices que servicios y componentes deben ser llamados, definiéndolos en los archivos de configuración de Spring. Esto hace el código fácil de mantener y más fácil de probar mediante la Inyección de Dependencia (IoC³⁶).

3.5.2. Gestión de la web

Su funcionamiento se basa en analizar las peticiones que van llegando y decidir si son válidas o no. Una vez tomada esta decisión, el módulo debe gestionar cada una de estas y realizar las operaciones indicadas en cada momento de la manera más rápida posible. Dentro de estas operaciones hay varias que tratan con la base de datos, tanto en transacciones de escritura como de lectura y otros que se gestionan independientemente de ella. El servidor es capaz de evaluar peticiones simultáneas de manera eficiente con usuarios concurrentes. El servidor debe estar continuamente operativo y "escuchando" toda la información que los diferentes dispositivos de la red envían, gestionando y dando respuesta si es necesario. Con este objetivo, se diseñó el servidor siguiendo la arquitectura modelo-vista-controlador, donde los diferentes elementos son:

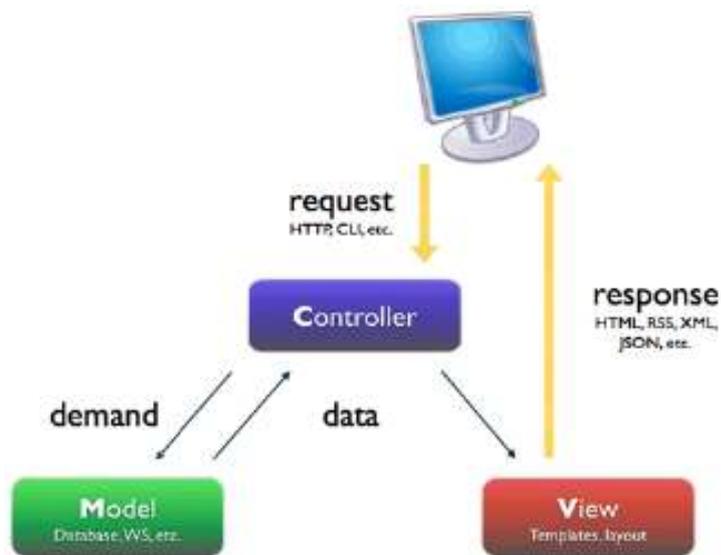


Ilustración 8: Modelo MVC

- **Modelo:** es cualquier elemento dentro de una vista. En iQuality, como modelo tenemos las tablas y los formularios que desarrollamos en html.
- **Vista:** es la presentación del modelo. En iQuality, la vista es la interfaz web.
- **Controlador:** es quien maneja el flujo de la aplicación. En iQuality, hay un controlador por cada vista.

3.5.2.1. Servlet

Un servlet⁴⁴ es un módulo programado en Java que se ejecuta en un servidor web y que, en general, amplía las capacidades de un servidor orientado a petición-respuesta. Este módulo actúa como capa intermedia entre la petición proveniente de un navegador web o cualquier otro cliente HTTP y la base de datos o aplicaciones en el servidor.

Las funcionalidades básicas de un servlet son las siguientes:

1. Leer los datos enviados por un usuario, normalmente provenientes tanto de formularios en páginas web como de applets⁶² de Java o programas cliente HTTP.
2. Buscar cualquier otra información sobre la petición que venga incluida en ella, ya sea capacidades del navegador, cookies, nombre del host del cliente, etc
3. Interactuar con la parte del servidor, ya sea con la base de datos, con aplicaciones, etc
4. Dar formato a la respuesta que se le enviará de vuelta a la página web.
5. Establecer los parámetros de la respuesta HTTP.
6. Enviar la respuesta al cliente.

El funcionamiento básico de un servlet es el siguiente:

1. El navegador web o cualquier aplicación que utilice un protocolo HTTP envía una petición al servidor.
2. El contenedor de servlets procesa los argumentos de la petición y la delega a un servlet particular escogido de entre los servlets que contiene.
3. Este servlet, que es un objeto java, gestiona la petición y se encarga de generar la respuesta.
4. El contenedor devuelve la respuesta en la página web o aplicación.

A continuación se muestra una figura que intenta ilustrar el funcionamiento y la función de un servlet dentro de un servidor web a un alto nivel:

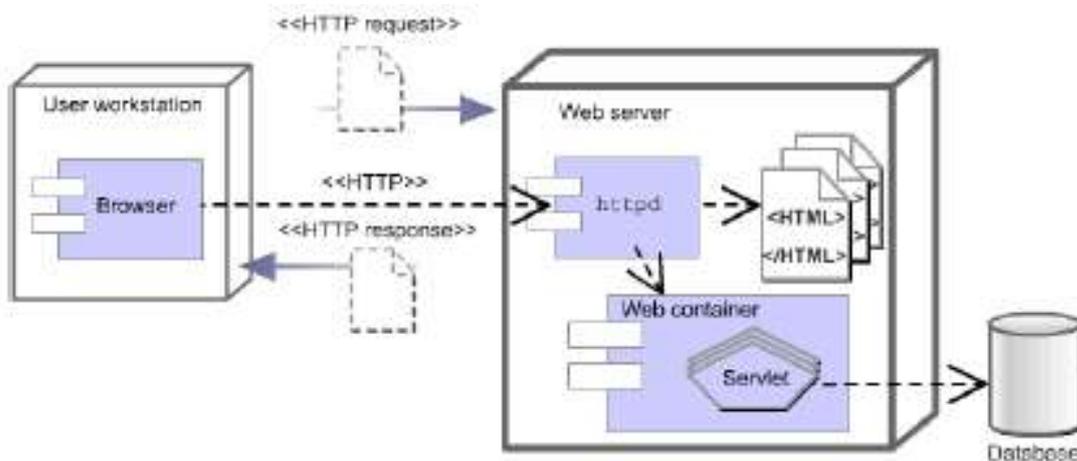


Ilustración 9: Comunicación servlet

Más detalladamente, el modelo de comunicación de un servlet viene descrito en la figura inferior, siguiendo el siguiente flujo:

1. El cliente envía una petición HTTP al servicio HTTP.
2. El servicio HTTP transmite los datos de la petición en el contenedor de servlets.
3. El contenedor de servlets crea un objeto que encapsula los datos de la petición. Dentro de esta petición, se diferencia en:
 - a. Consultas: Peticiones que piden datos.
 - b. Insertos: Peticiones que introducen datos en la base de datos.
 - c. Deletérea: Peticiones que borra información de la base de datos.
 - d. Updates: Peticiones que actualizan información de la base de datos o del sistema en ejecución.
 - e. Login: Peticiones relacionadas con inicios de sesión.
 - f. Incidencias: Peticiones relacionadas con las incidencias
 - g. Bus: Peticiones relacionadas con los autobuses.

En caso de que la petición no corresponda a ninguna de las mencionadas anteriormente, será devuelto un mensaje de error.

4. El contenedor ejecuta el servlet determinado por la acción requerida.
5. La respuesta del servlet se empaqueta en un HTTP Response, que se envía al servicio HTTP y se transmite al cliente.

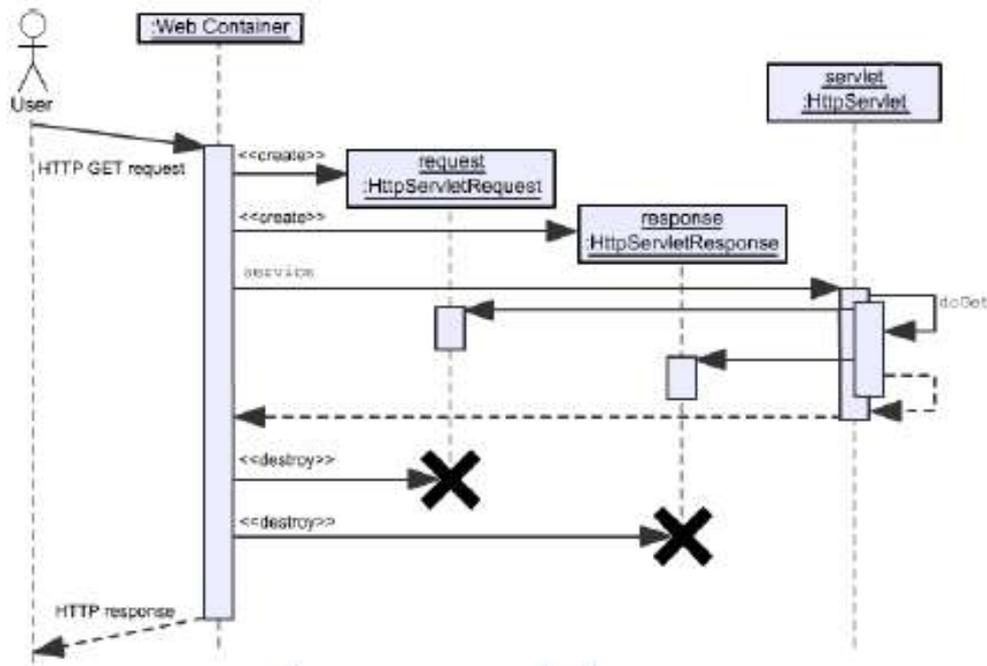


Ilustración 10: Flujo servlet

3.5. Maven

Maven¹⁷ es una herramienta de software para la gestión y construcción de proyectos Java¹ creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant³⁷ (y en menor medida a PEAR³⁸ de PHP³⁹ y CPAN⁴⁰ de Perl⁴¹), pero tiene un modelo de configuración de construcción más simple, basado en un formato XML³⁴. Estuvo integrado inicialmente dentro del proyecto Jakarta⁴² pero ahora ya es un proyecto de nivel superior de la Apache Software Foundation.

Maven utiliza un Project Object Model (POM⁴³) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Una característica clave de Maven es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar plugins de

un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos Open Source en Java¹, de Apache⁴⁴ y otras organizaciones y desarrolladores. Este repositorio y su sucesor reorganizado, el repositorio Maven 2, pugnan por ser el mecanismo de facto de distribución de aplicaciones en Java, pero su adopción ha sido muy lenta. Maven provee soporte no sólo para obtener archivos de su repositorio, sino también para subir artefactos al repositorio al final de la construcción de la aplicación, dejándola al acceso de todos los usuarios. Una caché local de artefactos actúa como la primera fuente para sincronizar la salida de los proyectos a un sistema local.

Maven está construido usando una arquitectura basada en plugins que permite que utilice cualquier aplicación controlable a través de la entrada estándar. En teoría, esto podría permitir a cualquiera escribir plugins para su interfaz con herramientas como compiladores, herramientas de pruebas unitarias, etcétera, para cualquier otro lenguaje. En realidad, el soporte y uso de lenguajes distintos de Java es mínimo. Actualmente existe un plugin para .Net Framework¹¹ y es mantenido, y un plugin nativo para C/C++ fue alguna vez mantenido por Maven 1.

3.6. Servidor Tomcat

Apache³⁷ Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets⁴⁴ y de JavaServer Pages (JSP²³) de Oracle Corporation (aunque creado por Sun Microsystems).

Tomcat es un contenedor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss⁴⁵ o JOnAS⁴⁶. Incluye el compilador⁴⁷ Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo

recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

El servidor que nosotros hemos usado ha sido la versión Tomcat 6.x:

- Implementado de Servlet⁴⁴ 2.5 y JSP²³ 2.1
- Soporte para Unified Expression Language 2.1
- Diseñado para funcionar en Java SE 5.0 y posteriores.
- Soporte para Comet⁴⁸ a través de la interfaz CometProcessor.

3.7. Eclipse

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE⁴⁹), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent⁵⁰ o Azureus⁵¹.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge⁵². Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU⁵³.

Como podemos ver en la *tabla 3 del Anexo*, este IDE⁴⁹ era el más adecuado para trabajar en el desarrollo en Java.

3.8. Dropbox

Dropbox es un servicio que te permite disponer de todos tus videos, documentos y fotos estés donde estés, además de compartirlos fácilmente. Cualquier archivo que almacenes en tu Dropbox se guardará automáticamente en todas tus computadoras, en tu teléfono o tablet y en el sitio web de Dropbox. Con Dropbox, también es fácil compartir. Además, si tu computadora colapsa, puedes restaurar todos tus archivos a través del sitio web de Dropbox en pocos clics.

- Estés donde estés. Al guardar tus archivos en Dropbox, estarán disponibles en todas tus computadoras, todos tus teléfonos o todas tus tablets. Edita documentos, agrega fotos automáticamente y mira videos estés donde estés.
- Comparte archivos con confianza. Comparte fotos con amigos. Trabaja con tu equipo como si usaran una única computadora. Todo es privado de forma automática. Tú controlas quiénes tienen acceso a cada elemento.
- Seguridad y protección. Aunque tu teléfono caiga al agua, tus archivos están siempre seguros en Dropbox y puedes restaurarlos de inmediato. Dropbox incluye cifrado AES⁵⁴ de 256 bits y una verificación de dos pasos para garantizar la seguridad de tus archivos.
- Dropbox para empresas. Hay millones de personas que usan Dropbox en el trabajo. Con Dropbox para empresas, obtienes la potencia y seguridad de Dropbox, además de controles de administración sólidos, un soporte dedicado y todo el espacio que necesitas.

En definitiva, Dropbox ha sido una herramienta útil la cual nos ha permitido tanto a ambos tutores como al proyectista compartir los documentos que el alumno iba generando y desarrollando para seguir una traza del trabajo desarrollado.

4. iQUALITY

iQuality es una herramienta que abarca 4 módulos: Control de carga, Diccionario de conceptos, Gestión de escenarios y Administración. Como funciona esta herramienta la podemos ver a continuación:

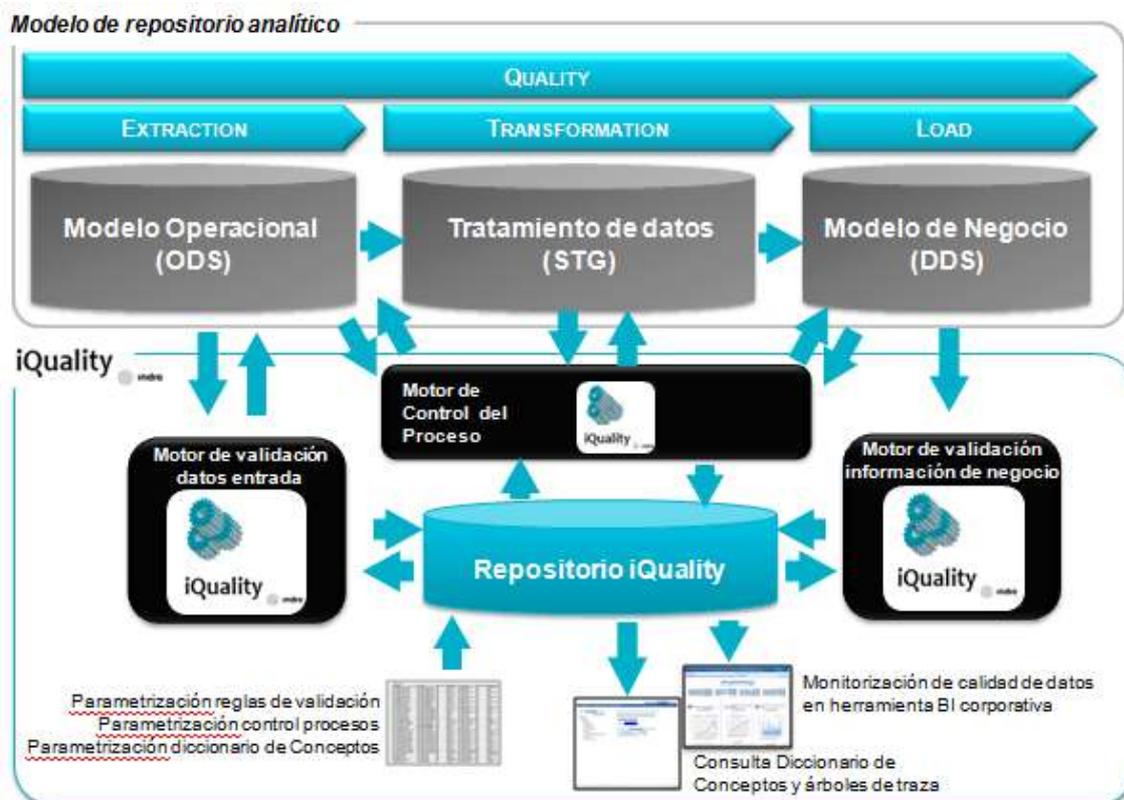


Ilustración 11: Flujo iQuality

iQuality es una herramienta que funciona paralelamente al repositorio analítico, el cual hace un tracking de la gestión que se está haciendo de los pases entre otros. El Modelo Operacional (ODS⁵⁵), el Tratamiento de datos (STG⁵⁶) y el Modelo de Negocio (DDS⁵⁷) son repositorios con los que nos comunicamos guardando así los datos de las ejecuciones y el progreso de ellas para que, en caso de fallo o error, podamos localizarlo y corregirlo. Todo este tracking se hace a través de tablas sobre una base de datos Oracle.

Del primer repositorio, ODS⁵⁵, extraemos los datos y, además, nos dice los ficheros de los que tenemos que tirar para aplicar las fórmulas a esos datos. Esta extracción de datos se hace mediante la comunicación con el motor de validación de datos de entrada. Su función principal es la de gestionar la planificación de los scripts que se lanzarán en cada proceso. En el segundo repositorio, STG⁵⁶, mantenemos un registro del proceso que nos indica que jobs (scripts) se han ejecutado OK, cuales han tenido un fallo y, por lo tanto, acabado en KO o en qué momento de la ejecución se encuentran. Se encarga de ello el Motor de Control del Proceso. Por último, tenemos el repositorio, DDS⁵⁷, que es gestionado por el Motor de validación de la Información de negocio. DDS contiene lo que podemos llamar las tablas resultado de todo el proceso iQuality. Estas tablas son las que más tarde serán usadas como datos para generar, por ejemplo, indicadores y atributos usando otras herramientas como Microstrategy⁵⁸, OBIEE⁵⁹, etc; que a su vez generan informes, pero esto, ya se sale del proyecto que abarcamos.

Como hemos mencionado anteriormente, este proceso es controlarlo por la herramienta iQuality que se define con los siguientes 4 módulos:

- Control procesos de carga: Módulo multifunción que carga ficheros en la base de datos Oracle y gestiona los pases que deben ser lanzados. Todo se parametriza con datos tipo String⁶⁰ que hacen referencia a los ficheros que se encuentran en un repositorio externo a la base de datos. Entre sus funciones principales podemos destacar:
 - Consulta de los resultados de las ejecuciones que generan certificaciones.
 - La consola de control de ejecución ejecuta los pases que se tienen que lanzar según la fecha planificada.
 - La gestión de planificaciones de cargas edita la periodicidad o fecha en que los pases tienen que ser lanzados.

Pase (cadena de jobs) → **Jobs** (scripts) → **Scripts** (código batch⁶¹ que contiene las fórmulas y parámetros que usa para hacer estas certificaciones)

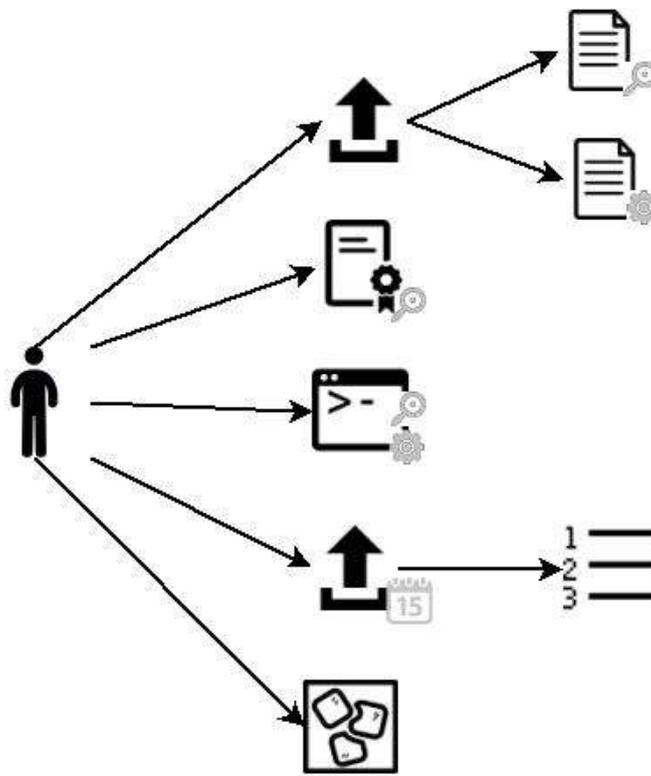


Ilustración 12: Módulo Control Procesos de Carga

- Diccionario de conceptos: Menú navegador en el cual puedes explorar y editar la descripción de atributos y entidades de las diferentes secciones del DMS¹⁵. DMS¹⁵ es un schema de la base de datos.

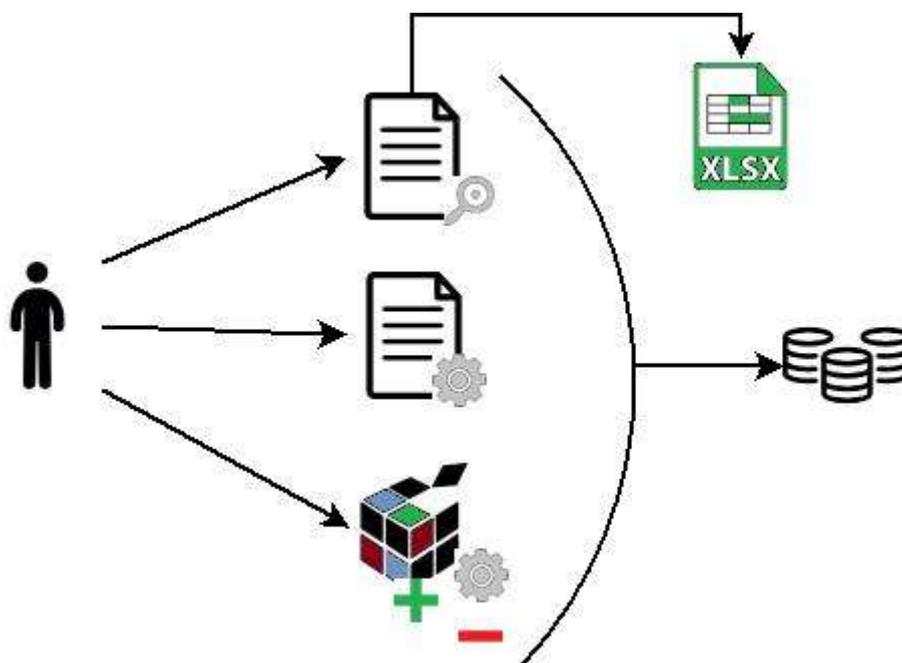


Ilustración 13: Módulo Diccionario de Conceptos

- Gestión de escenarios: Módulo que muy probablemente desaparezca por el uso casi nulo que se le da. Entre sus funciones:
 - Creación de escenarios
 - Eliminación de escenarios
 - Edición de escenarios

Un **escenario** es una situación donde se simula, se ejecutan o se testean los pases.

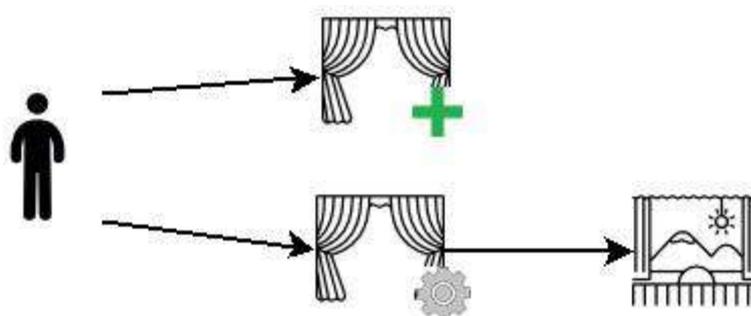


Ilustración 14: Módulo Gestión de Escenarios

- Administración: Módulo que se encarga de la administración de los usuarios. Entre sus funciones:
 - Creación de usuarios
 - Eliminación de usuarios
 - Edición de los usuarios
 - Creación de perfiles
 - Eliminación de perfiles
 - Edición de los permisos de los perfiles y usuarios

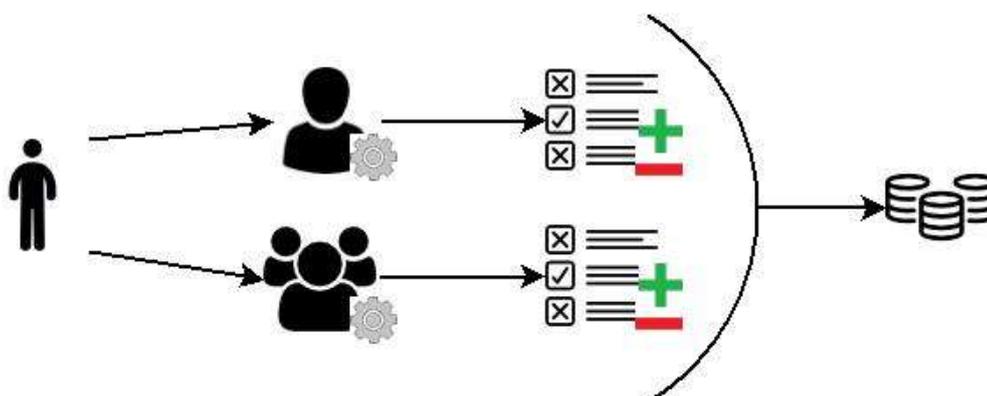


Ilustración 15: Módulo Administración

Los diagramas Entidad-Relación de la base de datos pertenecientes a cada módulo están adjuntados en el Anexo.

4.1. Interfaz web

La interfaz web del iQuality es la puerta de entrada al propio centro de control. El esqueleto principal de todas las pantallas está dividido en 3 como se puede ver con las marcas rojas en la imagen a continuación.

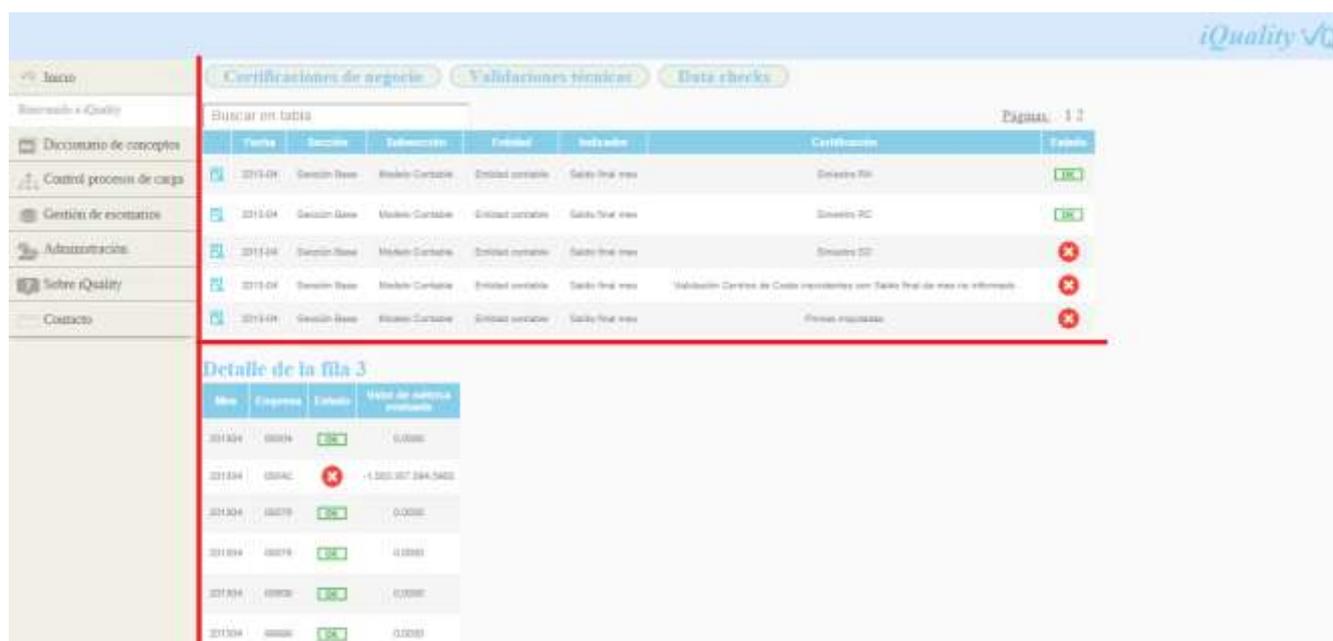


Ilustración 16: Diseño interfaz

A la izquierda, un menú siempre accesible para poder navegar entre pantallas y a la derecha la visualización de las tablas del módulo seleccionado. La parte de 'display' de la derecha también está dividida en 2 porque la parte superior será la que contenga los datos principales de ese módulo mostrándolos en una tabla. La parte inferior está dedicada a mostrar el detalle de la fila seleccionada en la tabla principal.

4.2. Perfiles de usuario

El uso de esta herramienta requiere un conocimiento técnico básico a nivel de usuario para su uso. Este problema se puede solventar con unas explicaciones sobre la navegación, creación y gestión de ejecuciones,

identificación de las tablas que entran en juego, etc. Un cálculo aproximado de esta formación sería un par de días, unas 16 horas laborales máximo de trabajo.

La herramienta está orientada a usuarios con un calibre técnico básico, en cambio los perfiles de acceso se pueden desglosar en 3; detallados en los siguientes puntos. Cada perfil explicado tiene sus derechos + derechos ampliados referente al perfil que le precede. Esto se puede observar en la imagen a continuación:



Ilustración 17: Perfiles de Usuario

4.2.1. Perfil Validación

Los usuarios de validación son los que en mayor parte son usuarios de Control de Riesgos.

Tienen como funcionalidades: la consulta del diccionario de conceptos, consulta del estado de los datos y de las cargas del DMS¹⁵, carga de ficheros Excel.

4.2.2. Perfil Funcional

Son usuarios funcionales los interlocutores definidos de Control de Riesgos y Sistemas de la Información.

Tienen como funcionalidades: la modificación del diccionario de conceptos (en el entorno de trabajo apropiado) y consulta de otros módulos.

4.2.3. Perfil Administrador

Los usuarios de sistemas o de mantenimiento de la aplicación son aquellos que tienen acceso a cualquier módulo de la herramienta. Entre sus funcionalidades está la gestión de los mismos y desarrollo o corrección de funcionalidades.

Como aclaración, tal y como se explica en el apartado 2.5, por motivos de falta de tiempo no se ha podido llegar más que a hacer un estudio de la viabilidad de la gestión de perfiles de usuarios.

4.3. Capturas de pantalla de la aplicación

Para visualizar la nueva herramienta, en este apartado añadiremos algunas capturas de pantalla de la aplicación.



Ilustración 18: Log In

Para empezar, en la imagen anterior podemos observar el formulario de 'Log In' donde se tiene que introducir usuario y contraseña. Un fichero JavaScript⁶⁴ se encarga de verificar de forma secuencial que el usuario y la contraseña son correctos y, en cuyo caso, te direcciona a la página principal donde se puede navegar al módulo que se desee; en caso contrario, un aviso en forma de alerta aparece en el navegador señalizando si el usuario o la contraseña han sido incorrectas como observamos en la imagen siguiente.

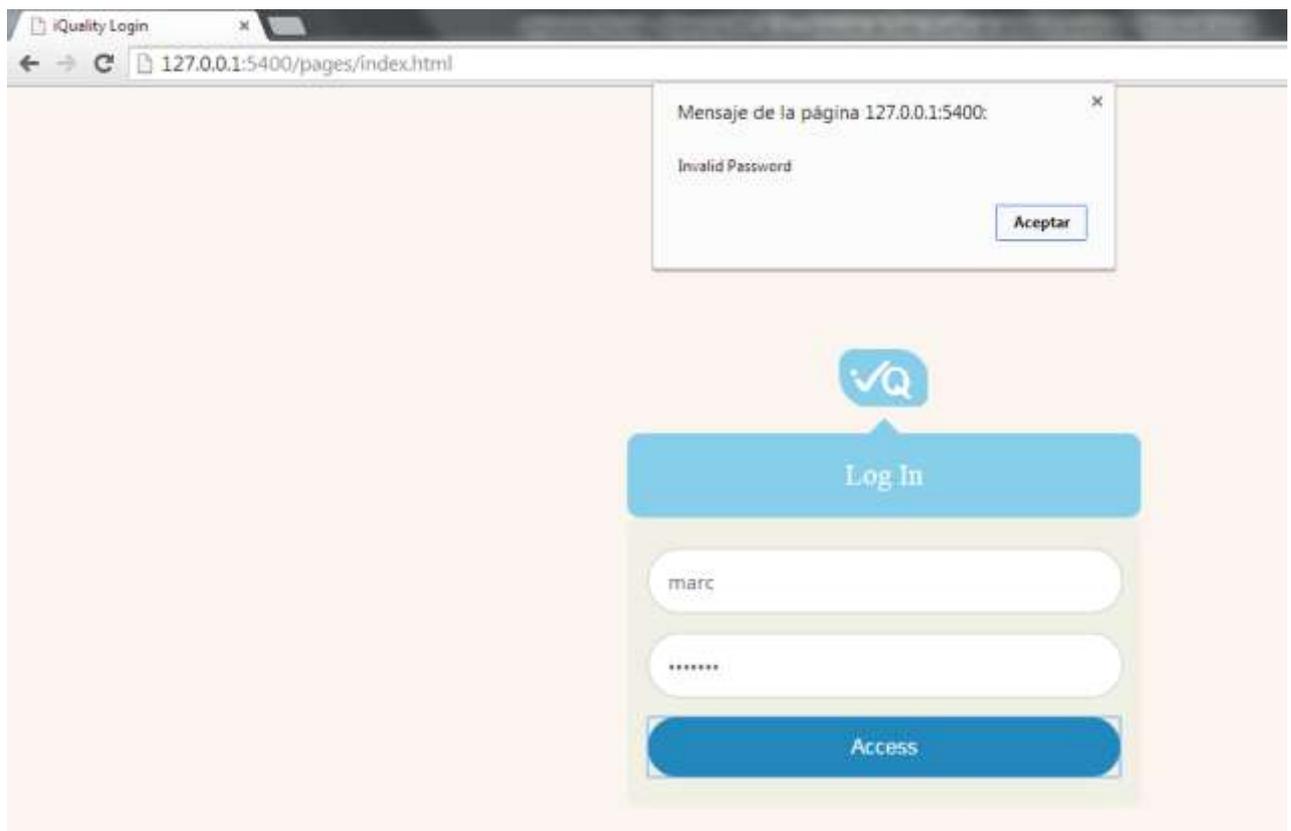


Ilustración 19: Log In fail

Desde el menú principal, que es la imagen que se muestra a continuación, vemos el menú de navegación desplegable a la izquierda, que estará presente en todas las pantallas para mantener la navegabilidad en todo momento, y en la parte central, enlaces directos a lo que se consideran las páginas principales de cada módulo.

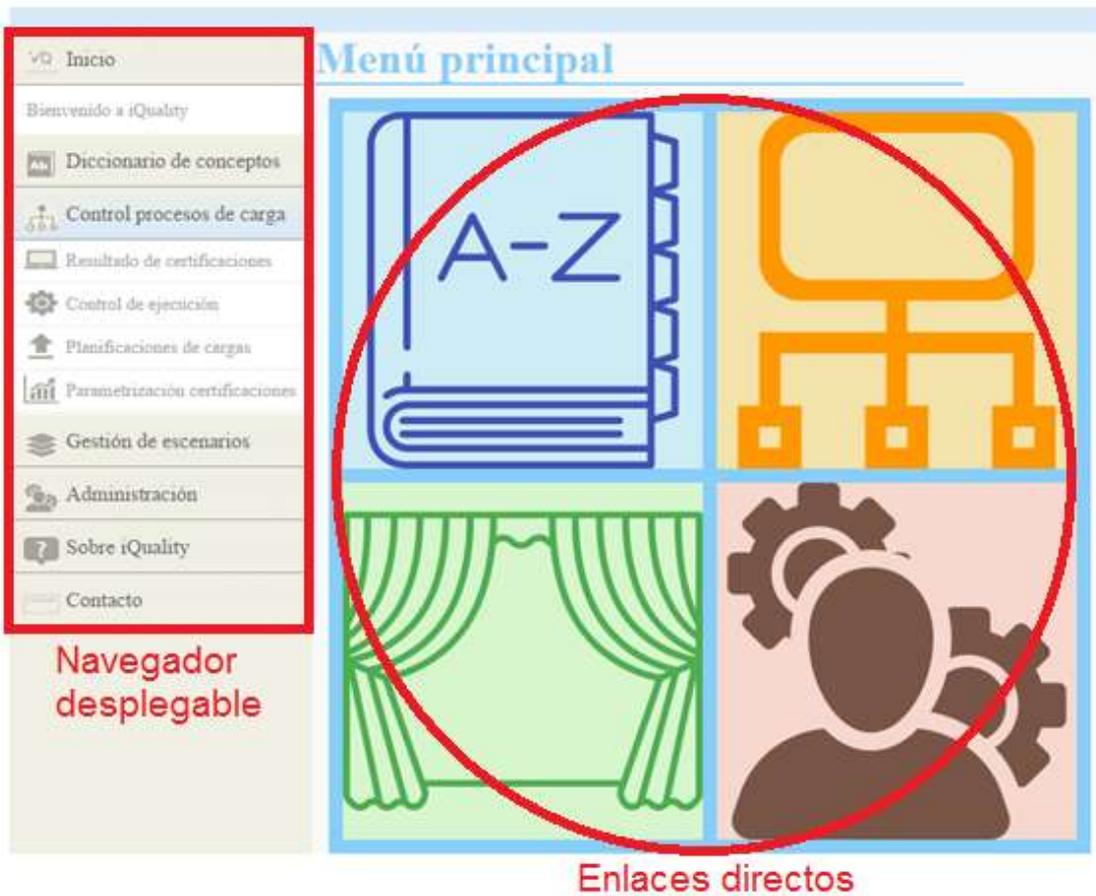


Ilustración 21: iQuality Main Page

Una vez introducidos dentro de los módulos, examinamos el módulo Control de Procesos de Carga – Resultado de Certificaciones, donde visualizamos el buscador de la tabla y una paginación, ambas controladas por un fichero Javascript⁶⁴. En su parte superior, encontramos los enlaces que nos redireccionan a los sub-módulos con sus tablas correspondientes y sus elementos complementarios (buscador y paginación).

Fecha	Sesión	Modelo Contable	Entidad contable	Indicador	Certificación	Estado
2013-04	Sesión Base	Modelo Contable	Entidad contable	Saldo final mes	Semestre RA	OK
2013-04	Sesión Base	Modelo Contable	Entidad contable	Saldo final mes	Semestre RC	OK
2013-04	Sesión Base	Modelo Contable	Entidad contable	Saldo final mes	Semestre SD	X
2013-04	Sesión Base	Modelo Contable	Entidad contable	Saldo final mes	Validación Centros de Coste Inversibles con Saldo final de mes no informado	X
2013-04	Sesión Base	Modelo Contable	Entidad contable	Saldo final mes	Financ impudable	X

Ilustración 20: iQuality tablas ejemplo

Para ejemplificar ahora la introducción de datos en una base de datos, adjuntamos a continuación la imagen de la creación de un nuevo usuario.

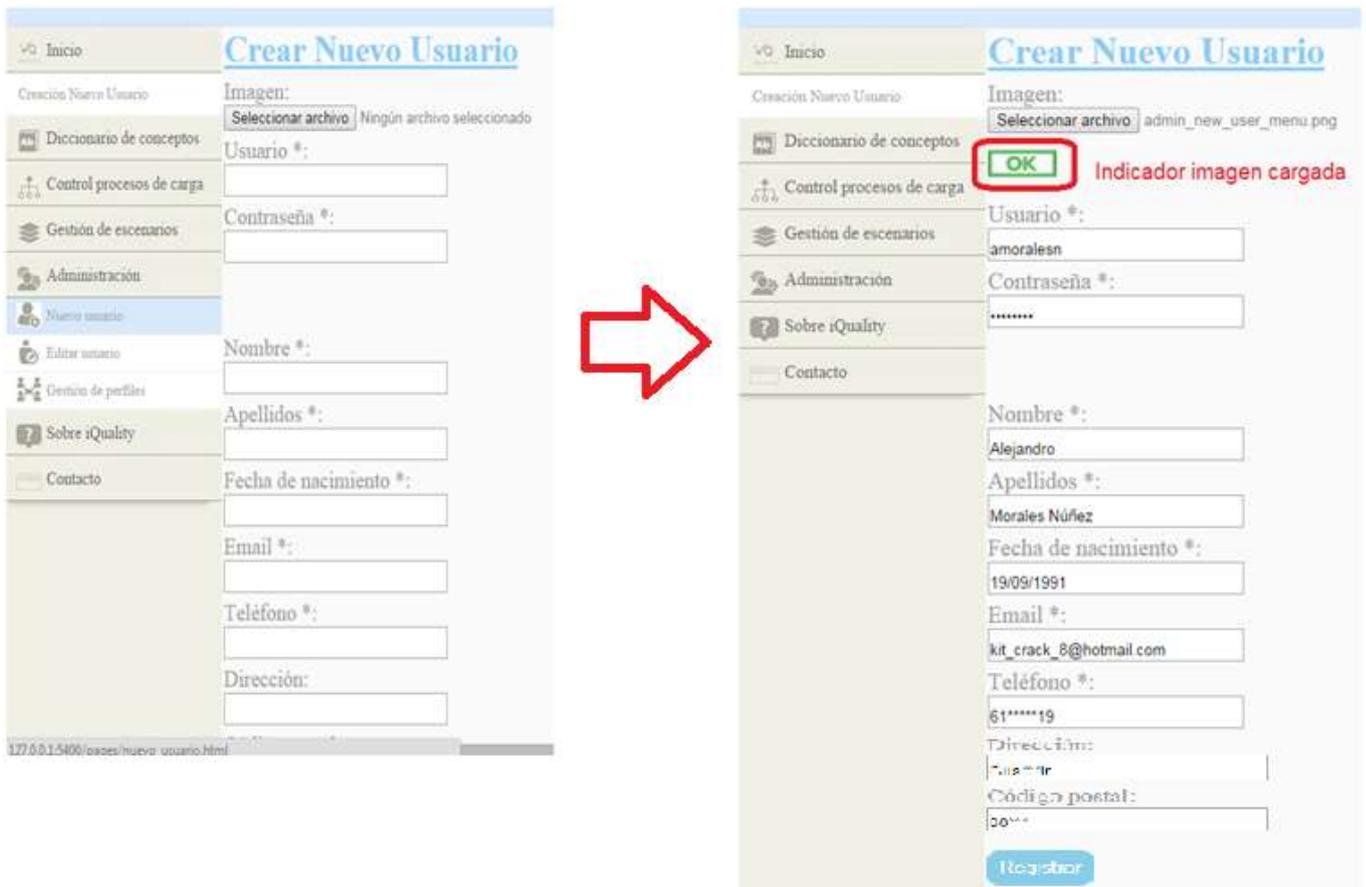


Ilustración 22: iQuality form ejemplo

Donde una vez pulsado el botón Registrar, una instrucción insert es enviada a la tabla de usuarios con los datos a insertar.

Esta forma de creación e inserción de nuevos datos, es el patrón que siguen las diferentes pantallas de nuestra aplicación, es decir, para la creación de un Pase el formulario tiene un aspecto semejante, pero los datos introducidos hacen referencia a códigos, acrónimos y datos bancarios que se introducen según petición del cliente.

5. CONCLUSIONES

5.1. Conclusiones del proyecto

En general, el balance del proyecto no ha sido del todo bueno. Por una parte, no se ha alcanzado el objetivo general del proyecto que era conseguir una herramienta con una serie de mejoras que otros productos similares no ofrecían, a la vez que se pretendía finalmente hacer un estudio del nivel de optimización de esta herramienta en comparación a la anterior.

En cuanto al equipo de trabajo, tampoco se ha conseguido un buen entendimiento entre los grupos de trabajo y una definición exacta de que queríamos abarcar con el proyecto. Se empezó con muy bien pie, pero ha derivado a una distorsión de objetivos por los problemas técnicos mencionados en el apartado 1.

Del mismo modo, siendo críticos con el proyecto, la planificación ha sido muy poco realista. En un principio estaba planeado llegar a la convocatoria de presentación de proyectos de fin de carrera en el mes de Septiembre, pero no se tuvieron en cuenta los posibles retrasos de cada módulo y la época de exámenes y trabajos finales de varias asignaturas que provocaron una gran desviación de horas en cuanto a trabajo a poder realizar fuera del horario laboral. Esto provocó, tal y como se ha comentado en apartados anteriores, que el proyecto terminara por desviar sus objetivos o dar por finalizado el proyecto con desarrollos de los controladores poco inteligentes.

En líneas generales, se ha demostrado que un proyecto educativo, complejo y de un tamaño considerable como es este, si se compara con lo que se está acostumbrado a trabajar en la carrera, es difícil sacar adelante en 5 meses sin un equipo de trabajo con conocimientos, constancia, trabajo y, sobre todo, una coordinación adecuada, haciendo que el proyecto responda a todas las exigencias que se pactaron al inicio del proyecto y teniendo en cuenta las faltas de conocimientos técnicos que pueden surgir y pérdidas de tiempo no calculadas.

5.2. Conclusiones personales

Por lo que hace a los objetivos personales alcanzados, puedo decir que no estoy del todo contento con el proyecto. Tal como se detalló en el apartado de Motivaciones Personales, uno de mis objetivos personales era el aprendizaje y alcanzar una serie de conocimientos y habilidades de desarrollo en tecnologías con las que no había tratado nunca. Esa era la motivación más grande del proyecto y llegados a este punto del proyecto, puedo afirmar sin ningún tipo de duda de que he logrado el objetivo de aprender aunque queda un sabor agridulce de no haber desarrollado algo suficientemente competente por falta de conocimientos técnicos.

Así mismo, otra motivación personal que hizo decidirme a formar parte de este proyecto era el de hacerlo en una empresa. Soy una persona a la que le gusta liderar, gestionar (tanto personas como proyectos) y dirigir, y aunque la experiencia me ha dicho que el estar sujeto a unos varemos de formación, económicos, horarios, ... y al ser el último eslabón de la cadena, es difícil gestionar y organizar a tu antojo cuando tienes gente por encima. Como conclusión personal en este apartado puedo afirmar que ha sido una experiencia muy enriquecedora en este aspecto pero dura por los enfrentamientos en la divergencia de opiniones y objetivos personales.

Finalmente, la satisfacción personal de poner en marcha un proyecto, desarrollarlo y pensarlo en cerrarlo como un producto con salida real en el mercado es la mayor de las recompensas que me hubiera dado este proyecto. No obstante, tengo que remarcar que el sabor agridulce que me deja el no finalizar iQuality en su totalidad, es una espina que espero algún día poder quitarme.

6. Referencias

[Informática Data Quality] ¿Qué es Informática Data Quality?

<http://international.informatica.com/la/products/data-quality/>

[iQuality] Objetivos iQuality en Indra

<https://neptu.uab.es/PROJFC/>

[Imagen fases iQuality] Descripción gráfica iQuality

Ppt proporcionado por Indra; [Data Quality en entornos analíticos v.2013 2.pptx](#)

[Fases de un proyecto] Imagen y texto de las fases

<http://www.dataprix.com/fases-de-implantacion-de-un-data-warehouse>

[Imagen cuadrante Gartner] Imagen y Información sobre evaluación herramientas

http://www.gartner.com/technology/reprints.do?id=1-259PU78&ct=141126&st=sb&mkt_tok=3RkMMJWWfF9wsRokv63Le%252B%252FhmjTEU5z17eQpWaWwh4kz2EFye%252BLIHETpodcMS8NjNa%252BTFawTG5toziV8R7HNJc160s8QXBjm

[Gartner] ¿Qué es Gartner?

http://es.wikipedia.org/wiki/Gartner_%28empresa%29

[Diagrama de Gannt] Diagrama temporal

<https://www.youtube.com/watch?v=4xMIUqFvwiw>

[Otros, HTML] Información general del Software utilizado

<https://es.wikipedia.org>

<http://www.tutorialspoint.com/html/index.htm>

[Spring-MVC] Tutorial desarrollo aplicación base con Spring-MVC

<http://www.uv.es/grimo/teaching/SpringMVCv3PasoAPaso/part1.html>

<http://www.tutorialspoint.com/servlets/servlets-first-example.htm>

<http://stackoverflow.com/questions/5055358/need-help-with-undestand-modelandview-in-java-spring>

[Otros] Resolución dudas en los desarrollos de las vistas y los controladores

<http://www.stackoverflow.com>

[HTML + css + js] Maquetación vistas

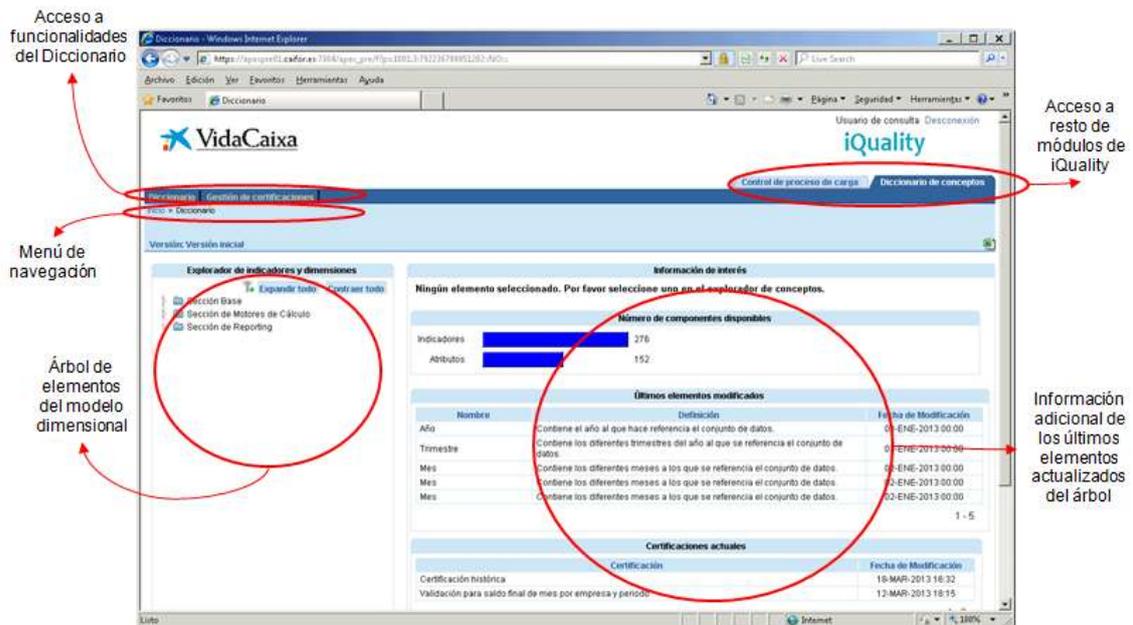
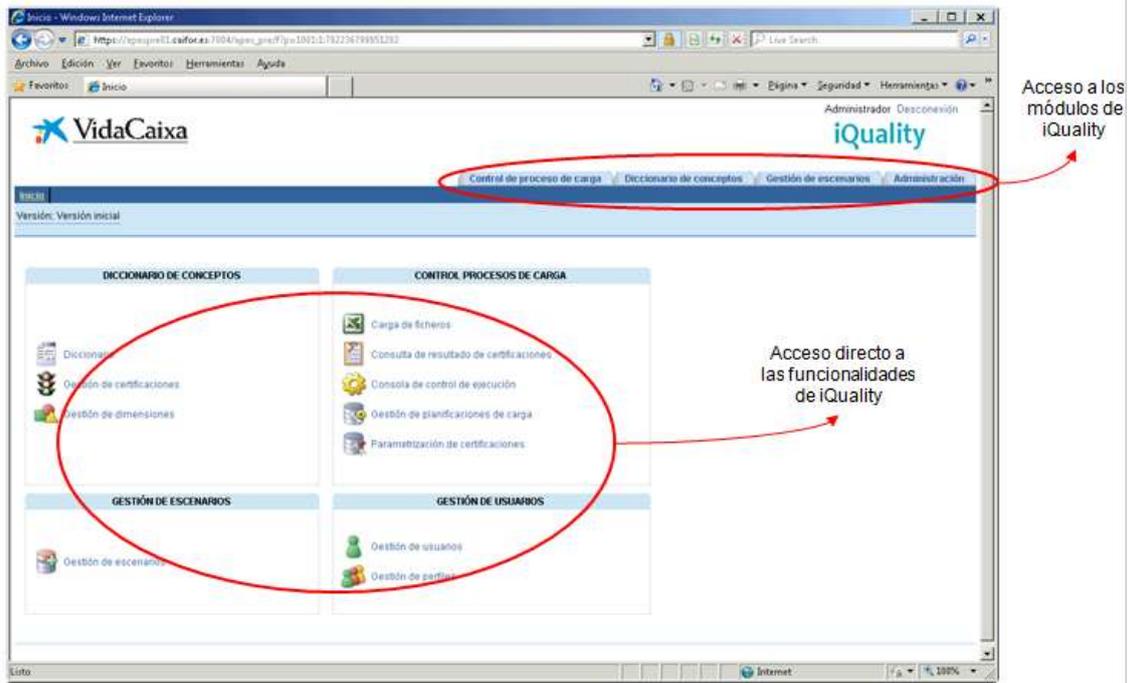
<http://market.envato.com/>

<http://themeforest.net/item/piluku-bootstrap-admin-angularjs-admin-template/full-screen-preview/11511932>

<http://www.flaticon.com/>

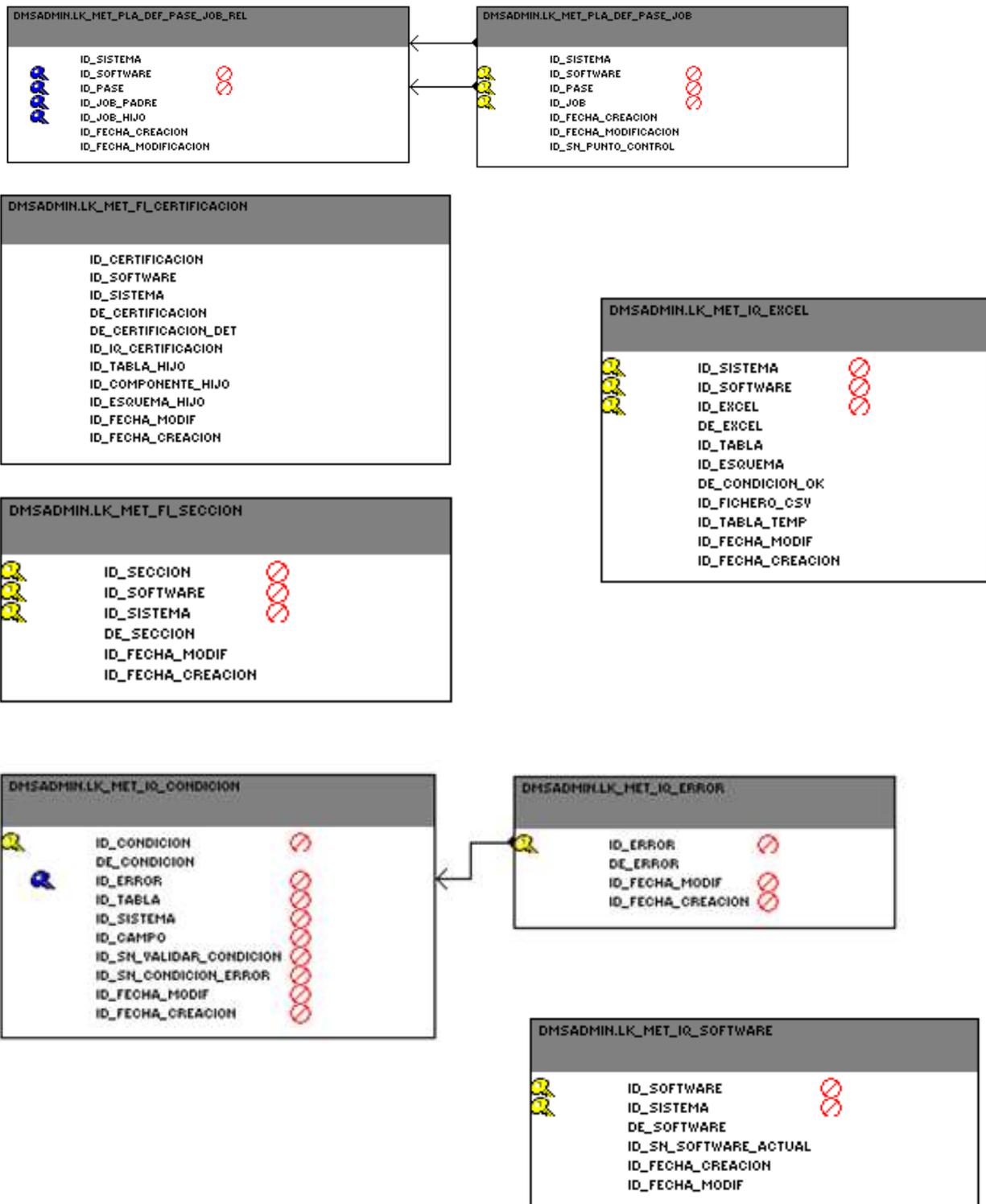
7. Anexo

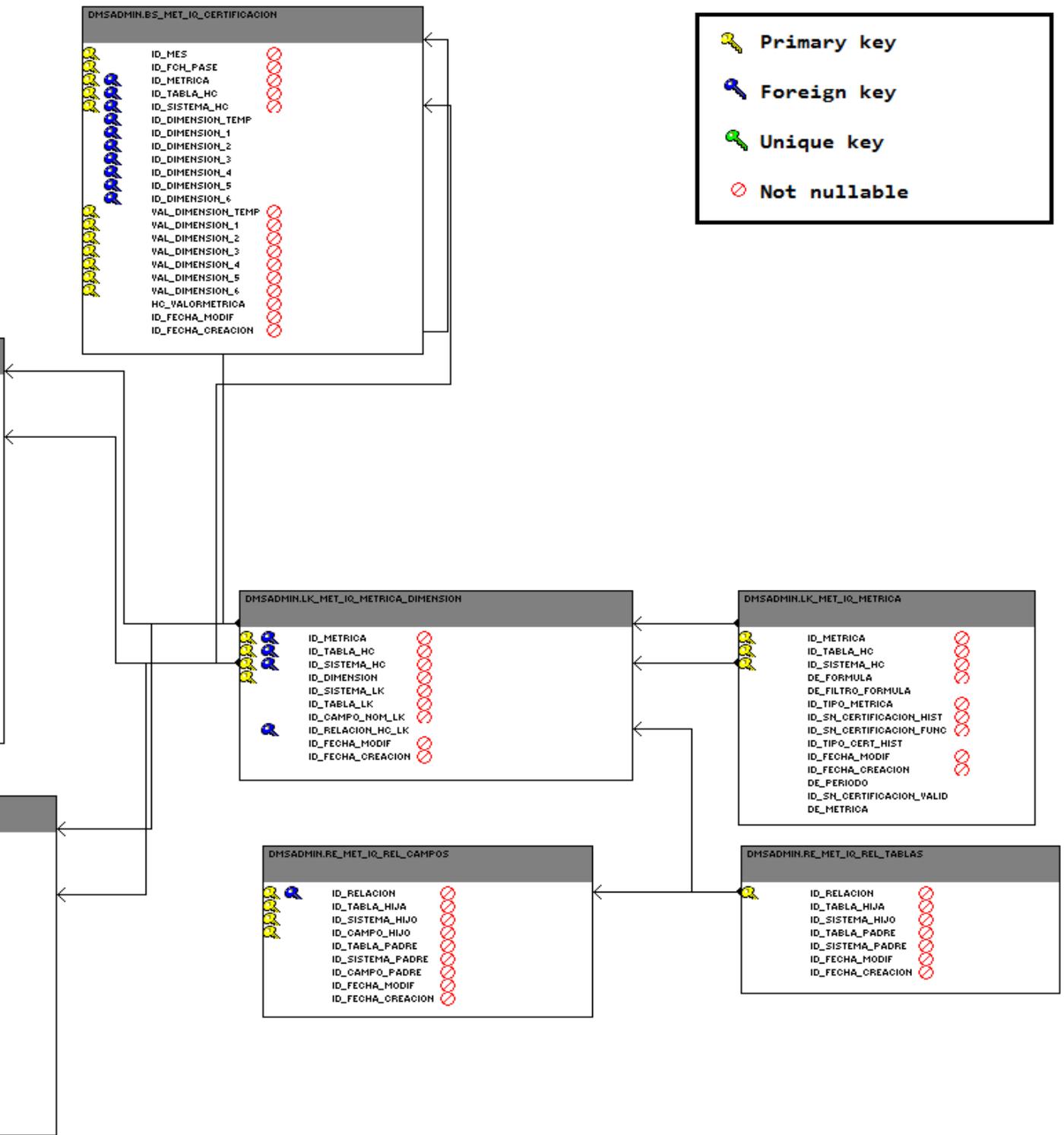
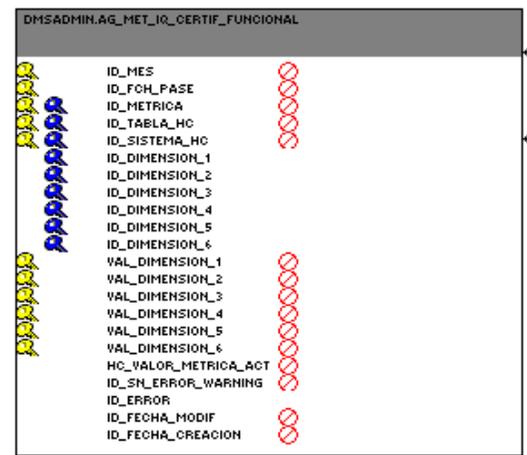
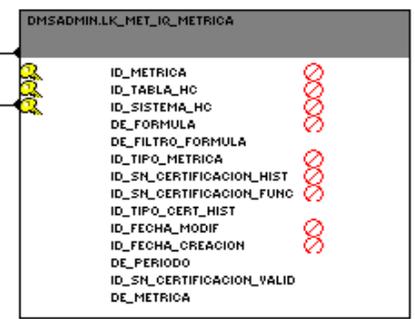
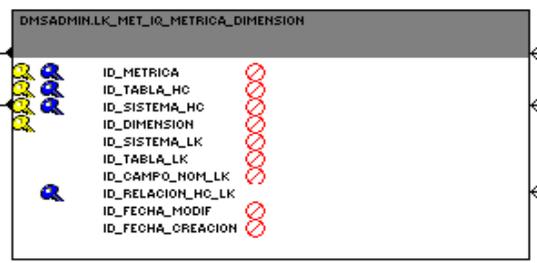
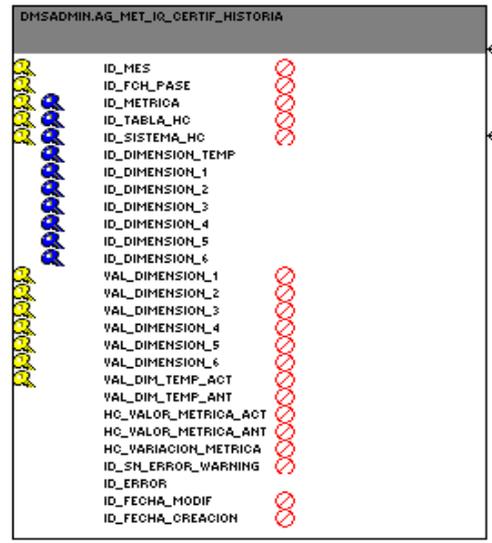
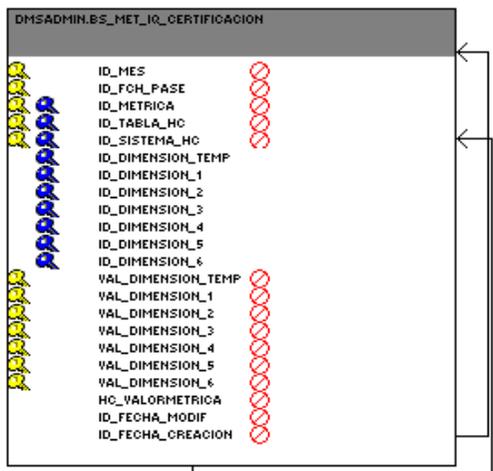
MODULOS Y FUNCIONALIDADES

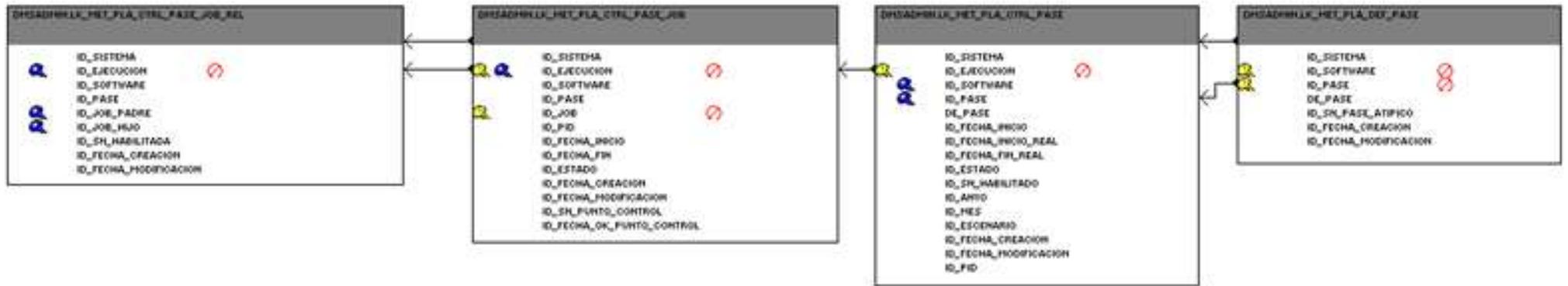


Diagramas Entidad-Relación

Control procesos de carga









DMSADMIN.LK_MET_IR_TABLA_CAMPO		
ID_TABLA	🔑	⊗
ID_ESQUEMA	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
ID_CAMPO	🔑	⊗
DE_DATA_TYPE		
DE_DATA_LENGTH		
DE_DATA_PRECISION		
DE_DATA_SCALE		
DE_NULLABLE		
DE_COLUMN_ID		
DE_AVG_COL_LEN		
DE_COMMENT		
ID_COMPONENTE		
ID_SH_PK		
ID_FECHA_MODIF		
ID_FECHA_CREACION		

DMSADMIN.LK_MET_IR_TABLA		
ID_TABLA	🔑	⊗
ID_ESQUEMA	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_COMMENT		
DE_TABLA		
ID_SH_USUARIO	🔑	
ID_SH_LOGICA	🔑	
ID_HOJA_ORI	🔑	
ID_SECCION_ORI	🔑	
ID_BLOQUE	🔑	
ID_SUBSECCION	🔑	
DE_HISTORICO		
ID_SH_CERTIFICACION_FUNG	🔑	
ID_SH_CERTIFICACION_HIST	🔑	
ID_SH_VALIDAR_CONDICIONES	🔑	
ID_FECHA_LAST_DDL	🔑	
ID_FECHA_CREACION	🔑	
ID_FECHA_MODIF	🔑	
ID_SH_MAEISTRA	🔑	
ID_PERIODO_ACT	🔑	
ID_TIPO_ACTUALIZACION	🔑	
DE_VERSIONADO		
ID_SH_ESCENARIO	🔑	
ID_SH_CERTIFICACION_VALID	🔑	

DMSADMIN.LK_MET_IR_BLOQUE		
ID_BLOQUE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_BLOQUE		
ID_FECHA_CREACION	🔑	
ID_FECHA_MODIF	🔑	

DMSADMIN.LK_MET_FL_SUBSECCION		
ID_SUBSECCION	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_SUBSECCION		
ID_SECCION	🔑	
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	

DMSADMIN.LK_MET_FL_TIPO_ACTUALIZACION		
ID_TIPO_ACTUALIZACION	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_TIPO_ACTUALIZACION		
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	

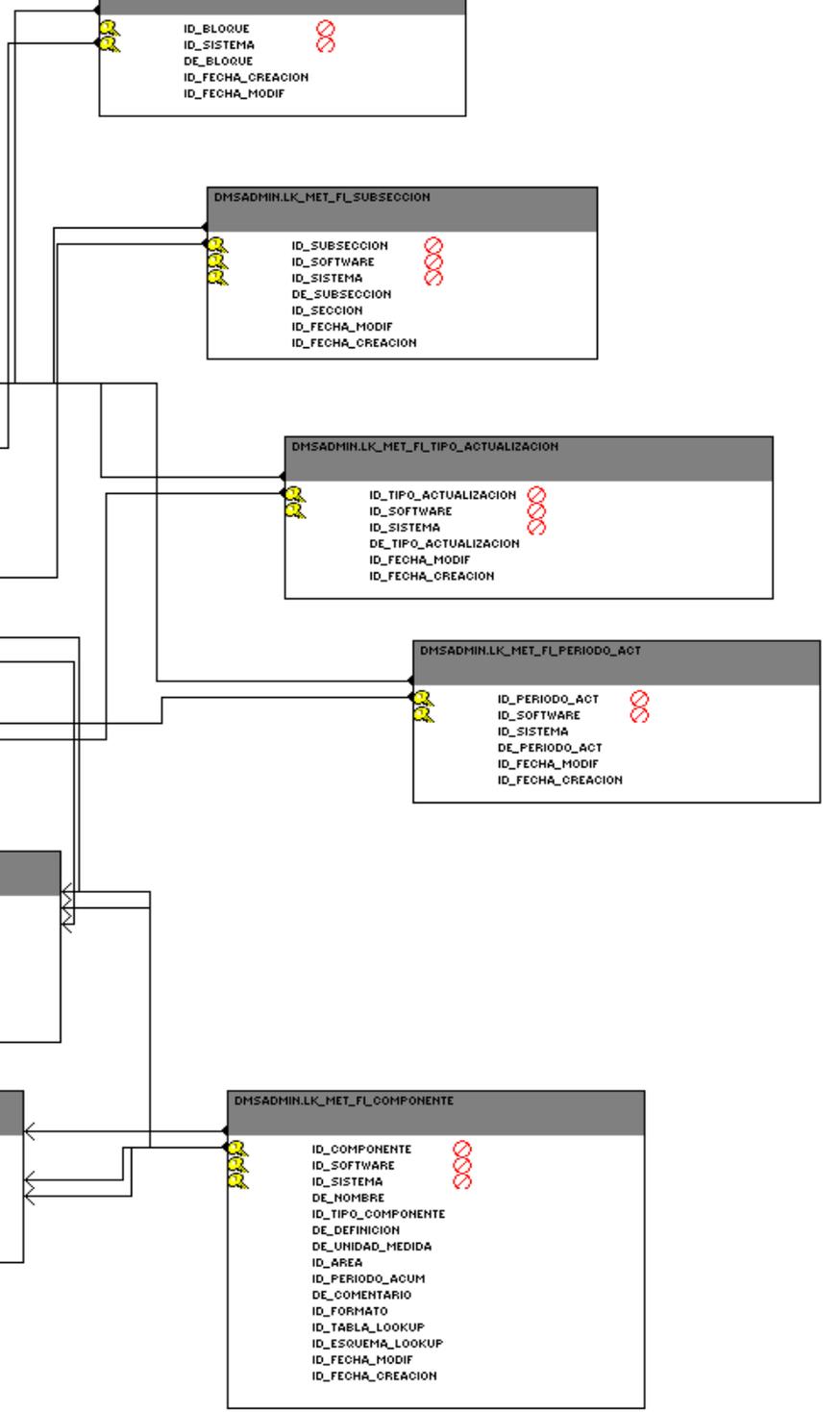
DMSADMIN.LK_MET_FL_PERIODO_ACT		
ID_PERIODO_ACT	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_PERIODO_ACT		
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	

DMSADMIN.RE_MET_FL_COMPONENTE_TABLA_DEP		
ID_COMPONENTE	🔑	⊗
ID_TABLA	🔑	⊗
ID_ESQUEMA	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
ID_ORDEN	🔑	⊗
ID_TABLA_HIJO	🔑	⊗
ID_ESQUEMA_HIJO	🔑	⊗
ID_COMPONENTE_HIJO	🔑	⊗
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	

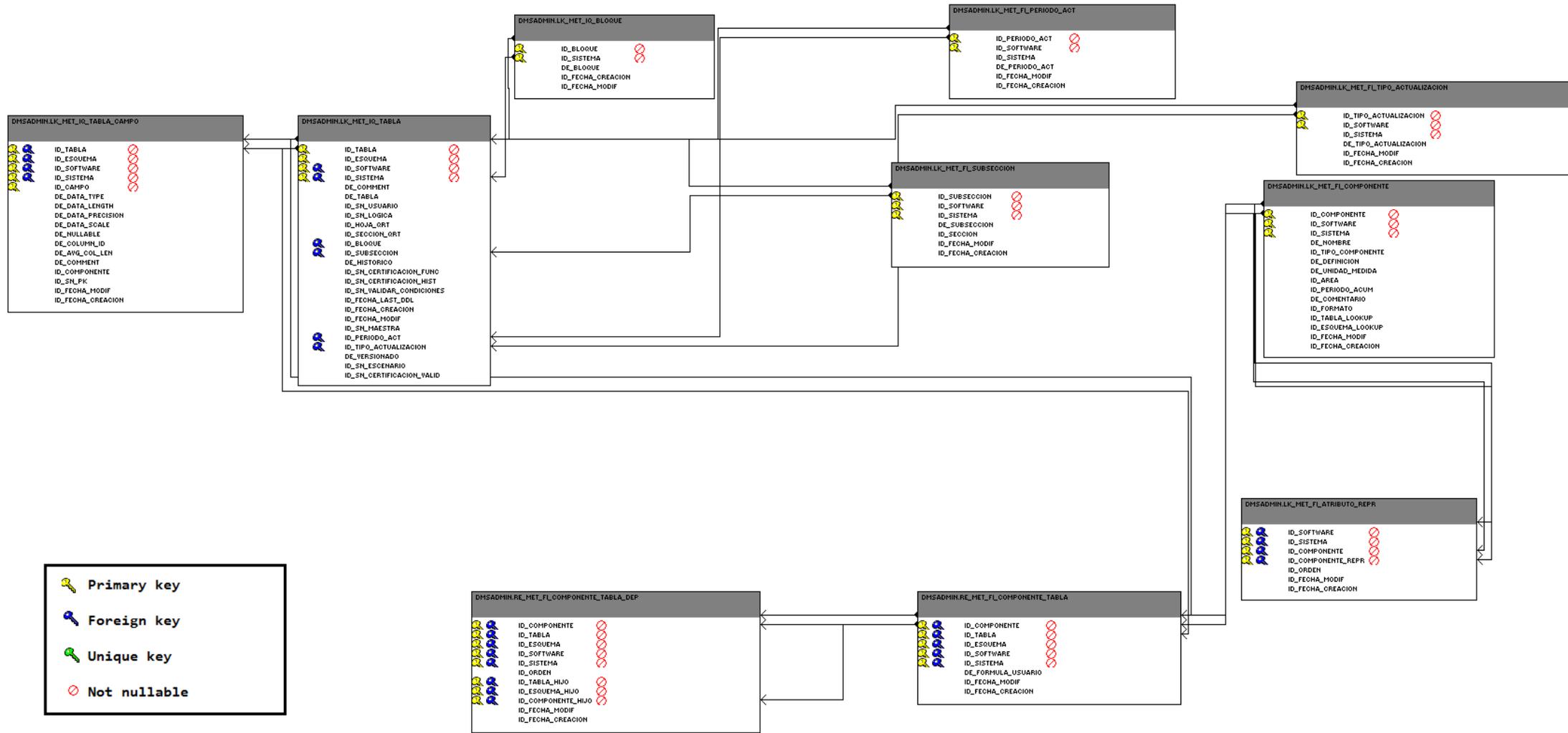
DMSADMIN.RE_MET_FL_COMPONENTE_TABLA		
ID_COMPONENTE	🔑	⊗
ID_TABLA	🔑	⊗
ID_ESQUEMA	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_FORMULA_USUARIO		
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	

DMSADMIN.LK_MET_FL_ATRIBUTO_REPR		
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
ID_COMPONENTE	🔑	⊗
ID_COMPONENTE_REPR	🔑	⊗
ID_ORDEN	🔑	
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	

DMSADMIN.LK_MET_FL_COMPONENTE		
ID_COMPONENTE	🔑	⊗
ID_SOFTWARE	🔑	⊗
ID_SISTEMA	🔑	⊗
DE_NOMBRE		
ID_TIPO_COMPONENTE	🔑	
DE_DEFINICION		
DE_UNIDAD_MEDIDA		
ID_AREA	🔑	
ID_PERIODO_ACUM	🔑	
DE_COMENTARIO		
ID_FORMATO	🔑	
ID_TABLA_LOOKUP	🔑	
ID_ESQUEMA_LOOKUP	🔑	
ID_FECHA_MODIF	🔑	
ID_FECHA_CREACION	🔑	



Diccionario de conceptos



DMSADMIN.LK_MET_FI_AREA			
	ID_AREA	NUMBER (3)	 IDX_1
	ID_SOFTWARE	NUMBER (10)	 IDX_1
	ID_SISTEMA	VARCHAR2 (3)	
	DE_AREA	VARCHAR2 (30)	
	ID_FECHA_MODIF	DATE	
	ID_FECHA_CREACION	DATE	

DMSADMIN.LK_MET_FI_PERIODO_ACUM		
	ID_PERIODO_ACUM	
	ID_SOFTWARE	
	ID_SISTEMA	
	DE_PERIODO_ACUM	
	ID_FECHA_MODIF	
	ID_FECHA_CREACION	

DMSADMIN.LK_MET_FI_CERTIFICACION	
ID_CERTIFICACION	
ID_SOFTWARE	
ID_SISTEMA	
DE_CERTIFICACION	
DE_CERTIFICACION_DET	
ID_IQ_CERTIFICACION	
ID_TABLA_HUO	
ID_COMPONENTE_HUO	
ID_ESQUEMA_HUO	
ID_FECHA_MODIF	
ID_FECHA_CREACION	

DMSADMIN.LK_MET_FI_SECCION		
	ID_SECCION	
	ID_SOFTWARE	
	ID_SISTEMA	
	DE_SECCION	
	ID_FECHA_MODIF	
	ID_FECHA_CREACION	

DMSADMIN.RE_MET_FI_CERTIFICACION_DEP	
ID_CERTIFICACION	
ID_SOFTWARE	
ID_SISTEMA	
ID_TABLA_HUO	
ID_ESQUEMA_HUO	
ID_COMPONENTE_HUO	
ID_ORDEN	
DE_FORMULA_USUARIO	
ID_FECHA_MODIF	
ID_FECHA_CREACION	

DMSADMIN.LK_MET_FI_SECCION_QRT		
	ID_SECCION_QRT	
	ID_SOFTWARE	
	ID_SISTEMA	
	DE_SECCION_QRT	
	ID_HOJA_QRT	
	ID_FECHA_MODIF	
	ID_FECHA_CREACION	

DMSADMIN.LK_MET_FI_FORMATO			
	ID_FORMATO		
	ID_SOFTWARE		
	ID_SISTEMA		
	DE_FORMATO		
	ID_FECHA_MODIF		
	ID_FECHA_CREACION		

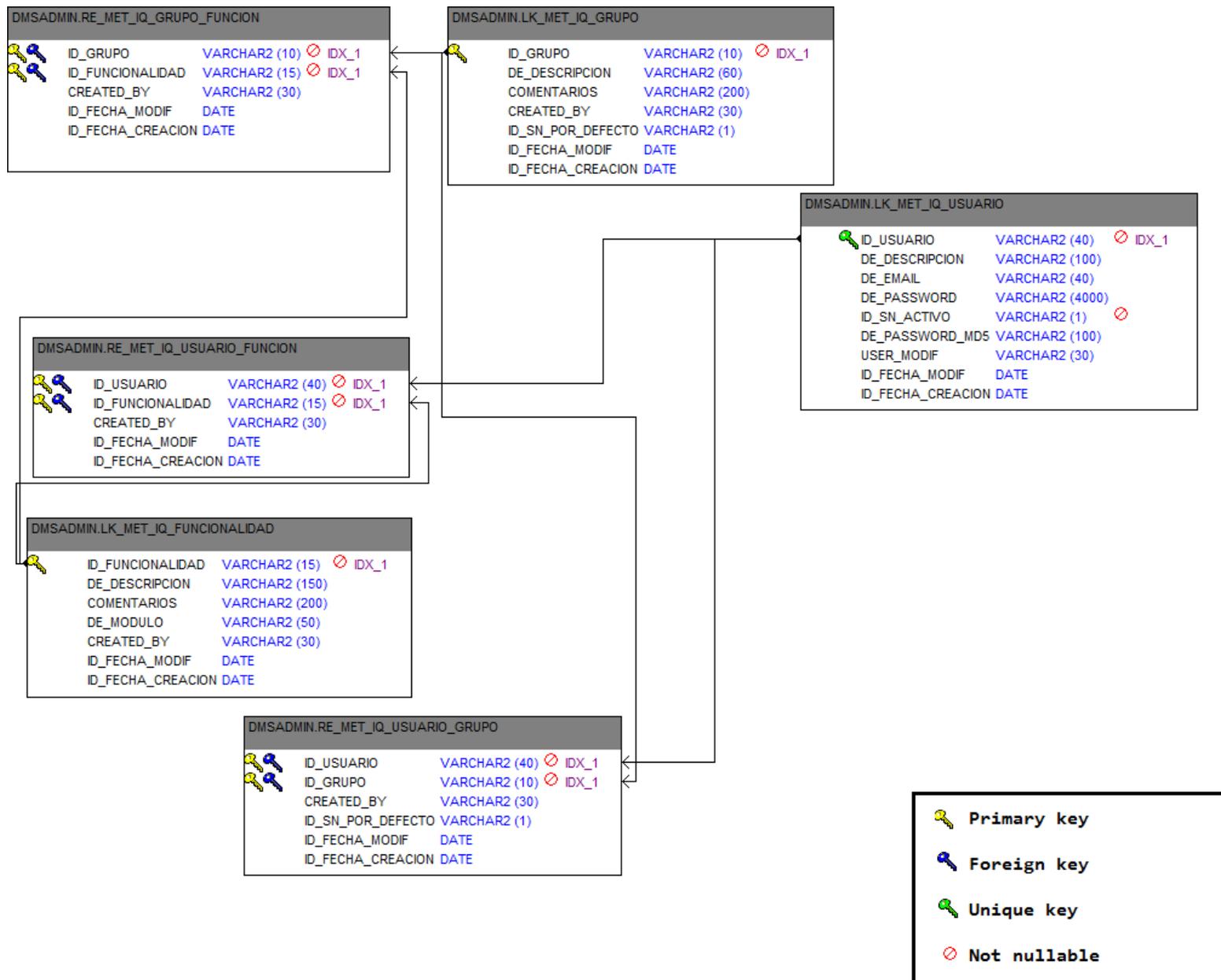
DMSADMIN.LK_MET_FI_TIPO_COMPONENTE		
	ID_TIPO_COMPONENTE	
	ID_SOFTWARE	
	ID_SISTEMA	
	DE_TIPO_COMPONENTE	
	ID_FECHA_MODIF	
	ID_FECHA_CREACION	

DMSADMIN.LK_MET_FI_HOJA_QRT			
	ID_HOJA_QRT		
	ID_SOFTWARE		
	ID_SISTEMA		
	DE_HOJA_QRT		
	ID_FECHA_MODIF		
	ID_FECHA_CREACION		

Gestión de escenarios

DMSADMIN.LK_MET_IQ_ESCENARIO	
ID_SOFTWARE	
ID_SISTEMA	
ID_ESCENARIO	
DE_ESCENARIO	
ID_SN_OFICIAL	
ID_FECHA_MODIF	
ID_FECHA_CREACION	

Administración



Tablespace	MB Tamaño	MB Usados	MB Libres	Fichero de datos
APEX_13075022603 18798	6,06	5,13	0,94	+DMS_DATA/dmst01/datafile/apex_1307502260318798.26 2.852568265
APEX411	100	1	99	+DMS_DATA/dmst01/datafile/apex411.264.857919263
PRUEBA01	100	1	99	+DMS_DATA/dmst01/datafile/prueba01.265.857919411
SYSAUX	4720	1966,75	2753,25	+DMS_DATA/dmst01/datafile/sysaux.257.848575037
SYSTEM	1480	1476,06	3,94	+DMS_DATA/dmst01/datafile/system.256.848575027
TB_DATOS	10240	4847,63	5392,38	+DMS_DATA/dmst01/datafile/tb_datos.277.875955517
TB_DATOS	10240	5672,88	4567,13	+DMS_DATA/dmst01/datafile/tb_datos.270.858080125
TB_DATOS	10240	5743,06	4496,94	+DMS_DATA/dmst01/datafile/tb_datos.269.858080107
TB_DATOS	10240	5096,63	5143,38	+DMS_DATA/dmst01/datafile/tb_datos.278.875955721
TB_DATOS	10240	5663,31	4576,69	+DMS_DATA/dmst01/datafile/tb_datos.268.858080089
TB_DATOS	10240	5154,13	5085,88	+DMS_DATA/dmst01/datafile/tb_datos.276.875955039
TB_DATOS	5120	5059,69	60,31	+DMS_DATA/dmst01/datafile/tb_datos.271.858080145
TB_DATOS	5120	4847,25	272,75	+DMS_DATA/dmst01/datafile/tb_datos.273.858080179
TB_DATOS	5120	4940,31	179,69	+DMS_DATA/dmst01/datafile/tb_datos.274.875808005

TB_DATOS	5120	4827,44	292,56	+DMS_DATA/dmst01/datafile/tb_datos.275.875808623
TB_DATOS	5120	4852,88	267,13	+DMS_DATA/dmst01/datafile/tb_datos.272.858080161
TB_INDICES	25600	211	25389	+DMS_DATA/dmst01/datafile/tb_indices.266.858079819
TB_INDICES	25600	203	25397	+DMS_DATA/dmst01/datafile/tb_indices.267.858079913
UNDOTBS1	25600	44,5	25555,5	+DMS_DATA/dmst01/datafile/undotbs1.258.848575043
UNDOTBS2	25600	2,25	25597,75	+DMS_DATA/dmst01/datafile/undotbs2.263.857919261
USERS	5	1	4	+DMS_DATA/dmst01/datafile/users.260.848575061

Tabla 1-Datos Base de datos Oracle

TABLESPACE_NAME	COPIA	KB
TB_DATOS	AG_	557248
SYSAUX	AQ\$	384
TB_DATOS	BS_	862720
SYSAUX	BSL	576
SYSAUX	COO	192
SYSAUX	CSW	192
SYSAUX	CWM	768
TB_DATOS	DGL	4096
SYSAUX	DUP	192
SYSAUX	EM_	768
SYSAUX	EXF	192

TB_DATOS	FK_	192
SYSTEM	HS\$	64
SYSTEM	HS_	64
SYSTEM	I_M	3072
TB_DATOS	IN_	192
SYSTEM	I_P	1600
SYSTEM	I_X	64
SYSTEM	JAV	256
TB_DATOS	LK_	17472
SYSTEM	LOG	192
SYSTEM	MET	384
SYSAUX	MGM	12352
SYSTEM	MVI	128
SYSAUX	PAR	192
SYSAUX	PK_	960
TB_DATOS	PK_	8000
TB_INDICES	PK_	3072
TB_DATOS	RE_	5440
SYSTEM	REP	320
SYSAUX	SDO	960
SYSAUX	SYS	1024
TB_DATOS	TM_	192
TB_DATOS	TMP	192
TB_DATOS	TV_	5312

SYSAUX	WRH	53056
SYSAUX	WRI	8896
SYSTEM	WRR	128
SYSTEM	X\$	64
SYSTEM	XST	64

Tabla 2-Tamaño que ocupan tablas de iQuality

Lenguaje	Líneas de código	%
<u>Java</u>	1.911.693	92,66%
<u>ANSI C</u>	133.263	6,46%
<u>C++</u>	10.082	0,49%

<u>JSP</u>	3.613	0,18%
<u>sh</u>	2.066	0,10%
<u>perl</u>	1.468	0,07%
<u>php</u>	896	0,04%
<u>sed</u>	2	0,00%

Tabla 3-Percentage código usado en Eclipse

Actas de Reunió

Acta de reunió seguiment 02/03/2015 despatx professors UAB

Temes tractats :

- Revisió dates (8/4 – informe previ i 10 al 14/09/15 memòria)
- Data de lliurament finals (16 al 22/09/15)
- Quedem que intentarem tenir la memòria per al mes de Juliol
- Cal revisar-se documentació Iquality que ha proporcionat INDRA

Compromisos

- Demanar i revisar informes previs d'altres projectes (AM) – abans 26/03/15
- Demanar memòries (AM) – abans del 26/03/15
- Compartir directori Dropbox (AM) – 04/03/15
- Proposta informe previ (AM) – 24/03/15
- Propera reunió de control 26/03/2015 a les 15:30h

Acta de reunió extraordinària 17/03/2015 via telefònica

Temes tractats :

- Revisió dels compromisos acordats. Tots els compromisos s'han efectuat segons la planificació establerta
- Revisió proposta estudi previ desenvolupada per l'alumne amb comentaris tutor empresa i del director de projecte

Compromisos

- Clarificar millor els objectius de l'estudi previ (AM) – 26/03/15
- Afegir a l'estat de l'art les alternatives que hi ha al mercat a part d'Iquality (AM) – 26/03/15
- Retocar i fer una planificació més global (AM) – 26/03/15
- Afegir un diagrama de Gantt a la planificació (AM) – 26/03/15
- Es confirma la propera reunió de control 26/03/2015 a les 15:30h

Acta de reunió seguiment 26/03/2015 despatx professors UAB

Temes tractats :

- Revisió dels compromisos acordats. Tots els compromisos s'han efectuat segons la planificació establerta, excepte el diagrama de Gantt.
- Revisió i validació de la proposta estudi previ desenvolupada per l'alumne retocada segons comentaris direcció de projecte.

Compromisos

- Afegir un diagrama de Gantt a la planificació (AM) – 27/03/15
- Presentar l'estudi previ a aplicació UAB (AM) – 07/04/2015
- Validar l'estudi previ a aplicació UAB (AP) – 08/04/2015
- Es confirma la propera reunió de control 22/04/2015 a les 15:30h

Acta de reunió seguiment 22/04/2015 despatx professors UAB

Inici: 16:20h

Durada: 90min

Temes tractats :

- Revisió dels compromisos acordats. Tots els compromisos s'han efectuat segons la planificació establerta.
- Revisió del plantejament de la memòria i la metodologia de treball.
- Elaboració full imputació horària.

Compromisos

- Omplir imputació horària (AM) . durant tot el projecte
- Anàlisi de requeriments (AM) – 27/05/15
- Esquema global de mòduls (escenaris, diccionari, administrador usuaris y processos de càrrega) (AM) – 27/05/2015
- Descripció funcional de cada mòdul (AM) – 27/05/2015
- Esquema particular de cada mòdul (AM) – 27/05/2015
- Insertar estructura de la base de dades (AM) – 27/05/2015
- Es confirma la propera reunió de control 27/05/2015 a les 15:30h

Acta de reunió seguiment 27/05/2015 despatx professors UAB

Inici: 16:10h

Durada: 90min

Temes tractats :

- Revisió dels compromisos acordats.
- Imputació horària feta segons l'acordat.
- Esquema global de mòduls (escenaris, diccionari, administrador usuaris y processos de càrrega): esta fet i cal afegir retocs especificats a compromisos
- Desenvolupat fluxe de navegació de tots el mòduls i submòduls, ara cal complementar amb la descripció funcional
- La estructura de la base de dades es molt complexe i no aporta res al projecte. Es pensa guardar alguns pantallassos en cas d'haver d'ensenyar

Compromisos

- Descripció funcional de cada mòdul (AM) – 22/06/2015 - 27/05/2015
- Esquema “entitat-relació” de cada mòdul (AM) – 22/06/2015 – 27/05/2015
- Afegir login esquema global modificat (AM) – 22/06/2015
- Es confirma la propera reunió de control 22/06/2015 a les 15:30h

Acta de reunió seguiment 30/06/2015 despatx professors UAB

Inici: 16:10h

Durada: 80min

Temes tractats :

- Revisió dels compromisos acordats.
- Imputació horària feta segons l'acordat.
- Descripció funcional de cada mòdul: falta esquema pantalles que s'han de desenvolupar. Cal un dibuix de cada pantalla amb els camps que s'han de mantenir.
- Esquema “entitat-relació” de cada mòdul: falta per a cada mòdul (Diccionari d conceptes, Control de processos de càrrega, Gestió

d'escenaris i Administració). Un esquema d'entitats que intervenen i relacions entre elles.

Compromisos

- Memòria provisional per al 30/07
- Propera reunió
- Presentar descripció funcionalitats disseny de pantalles de cada mòdul.
- Presentar esquema "entitat-realció" acabat de cada mòdul.

Acta de reunió seguiment 29/07/2015 despatx professors UAB

Inici: 15:10h

Durada: 80min

Temes tractats :

- Revisem mòdul usuaris. Disposem del diagrama entitat – relació però falta saber la funcionalitat i significat de les diferents entitats.
- Revisem mòdul de càrregues i ens trobem amb el mateix problema.
- Veiem que tenim desenvolupats el submòdul de resultats de certificacions que es troba al mòdul de control de processos de càrrega però hi ha problemes tècnics de connexió amb l'html.
- Decidim :
 - o resoldre el problema tècnic de connexió amb html per poder mostrar l'aplicatiu
 - o acabar la memòria amb el seguiment de projecte i els dissenys conceptuals, de requeriments i funcionals dels mòduls i submòduls
- Propera reunió dimarts 01/09/2015 a les 15h per decidir si demanem o no la lectura de la memòria.

Compromisos

- Fer els diagrames conceptuals (estil diagrama global) dels diferents mòduls de l'aplicatiu: Diccionari de conceptes, Control de processos de càrrega i gestió d'usuaris. Fer també els submòduls.
- Acabar l'apartat de desenvolupament de projecte (punt 4.).

- Especificar a 4.5 una descripció de cada mòdul, les funcionalitats i pantallassos del que s'ha desenvolupat tal i com hem comentat al primer punt.
- A l'apartat 4.5 fer una descripció de les principals millores a nivell global que s'han aconseguit. (navegació, menús, etc).
- Presentar la versió definitiva de la memòria per validar pel director

8. Glosario

1. **Java:** Lenguaje de programación orientado a objetos
2. **Apex:** Herramienta de desarrollo rápido de aplicaciones que se ejecuta con una base de datos Oracle.
3. **BI:** Business Intelligence
4. **Oracle:** es una compañía de software que desarrolla bases de datos y sistemas de gestión de bases de datos
5. **PL/SQL:** Procedural Language/Structured Query Language es un lenguaje de programación incrustado en Oracle
6. **SQL:** Structured Query Language es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas
7. **RAD:** Rapid Application Development es un proceso de desarrollo de software. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades de ingeniería asistida por computadora
8. **BBDD:** Base de datos
9. **TI:** La infraestructura de TI de una empresa provee la base para dar servicio a los clientes, trabajar con los distribuidores y gestionar los procesos de negocios internos. Incluye la inversión en hardware, software y servicios (como consultoría, educación y capacitación) que se comparten a través de toda la empresa o de unidades de negocios completas de ésta
10. **SaaS:** Software as a Service es un modelo de distribución de software donde el soporte lógico y los datos que maneja se alojan en servidores de una compañía de tecnologías de información y comunicación (TIC), a los que se accede vía Internet desde un cliente
11. **Framework:** palabra inglesa que se define como marco de trabajo. En términos generales se conoce como un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar
12. **C#:** es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET

- 13.Back-end:** puede traducirse al español como motor, dorsal final o zaga. Son términos que se relacionan con el final del proceso. Estos términos adquieren una relevancia mayor en ciertas áreas particulares
- 14.Spring-MVC:** nombre compuesto que se le da a la herramienta formada por el framework de Spring y el Modelo-Vista-Controlador
- 15.DMS:** Data Management System es una colección de datos de múltiples fuentes organizadas para la distribución, el intercambio, y a menudo compartición de subconjuntos. En general, esta distribución de datos está en la forma de una arquitectura radial
- 16.Spring:** es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java
- 17.Maven:** es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML
- 18.C++:** es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos
- 19.Python:** es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible
- 20.Interfaz:** en informática se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles
- 21.HTML:** siglas de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web
- 22.JavaFX:** es una familia de productos y tecnologías de Sun Microsystems, adquirida por Oracle Corporation, para la creación de Rich Internet Applications (RIAs), esto es, aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas

- 23.Jsp:** JavaServer Pages es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos
- 24.RDBMS:** Sistema de gestión de bases de datos relacionales, es un sistema de gestión de base de datos (DBMS) que se basa en el modelo relacional
- 25.API:** Application Programming Interface es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción
- 26.REST:** Representational State Transfer es un estilo de arquitectura software para sistemas hipermedia distribuidos como laWorld Wide Web
- 27.RPC:** Remote Procedure Call es un protocolo de red que permite a un programa de computadora ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambas
- 28.RMI:** Remote Method Invocation es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías debe utilizarse CORBA o SOAP en lugar de RMI
- 29.CORBA:** Common Object Request Broker Architecture (CORBA) es un estándar definido por Object Management Group (OMG) que permite que diversos componentes de software escritos en múltiples lenguajes de programación y que corren en diferentes computadoras, puedan trabajar juntos; es decir, facilita el desarrollo de aplicaciones distribuidas en entornos heterogéneos
- 30.SOAP:** Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML
- 31.Acegi:** es un framework Java / Java EE que proporciona autenticación, autorización y otras características de seguridad para aplicaciones empresariales

- 32.JMX:** Java Management eXtensions es la tecnología que define una arquitectura de gestión, la API, los patrones de diseño, y los servicios para la monitorización/administración de aplicaciones basadas en Java
- 33.JMS:** Java Message Service es la solución creada por Sun Microsystems para el uso de colas de mensajes. Este es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma Java2 crear, enviar, recibir y leer mensajes
- 34.XML:** eXtensible Markup Language es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible
- 35.EJB:** Enterprise JavaBeans son una de las interfaces de programación de aplicaciones que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Oracle Corporation
- 36.ioC:** describe un diseño en el que porciones medida escrita de un programa informático reciben el flujo de control de un genérico, librerías reutilizables
- 37.Apache:** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual
- 38.PEAR:** PHP Extension and Application Repository es un entorno de desarrollo y sistema de distribución para componentes de código PHP
- 39.PHP:** es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo webde contenido dinámico
- 40.CPAN:** Comprehensive Perl Archive Network es un enorme archivo de software escrito en Perl, así como de documentación sobre el mismo
- 41.Pperl:** es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación
- 42.Proyecto Jakarta:** proyecto que crea y mantiene software de código abierto para la plataforma Java. Opera como un proyecto paraguas

- bajo el auspicio de la Apache Software Foundation, y todos los productos producidos por Jakarta son liberados bajo la Licencia Apache
- 43. POM:** responde a las siglas de Project Object Model, es un fichero XML, que es la “unidad” principal de un proyecto Maven. Contiene información a cerca del proyecto, fuentes, test, dependencias, plugins, versión, etc
- 44. Servlet:** es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web
- 45. JBoss:** es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java
- 46. JOnAS:** es un servidor de aplicaciones J2EE de código abierto implementado en Java. Forma parte de la iniciativa de código abierto de ObjectWeb, la cual fue lanzada en colaboración con varios socios, entre los que se encuentran Bull, France Télécom e INRIA.
- 47. Compilador:** es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación.¹ Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio (bytecode), o simplemente texto
- 48. Comet:** describe un modelo de aplicación web en el que una petición HTTP mantenida abierta permite a un servidor web enviar datos a un navegador por Tecnología Push, sin que el navegador los solicite explícitamente
- 49. IDE:** Integrated Development Environment es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software

- 50.BitTorrents:** es un protocolo diseñado para el intercambio de archivos punto a punto (peer-to-peer) en Internet. Es uno de los protocolos más comunes para la transferencia de archivos grandes
- 51.Azerus:** cliente BitTorrent
- 52.VisualAge:** era el nombre de una familia de desarrollos integrados informáticos en entornos de desarrollo de IBM, que incluye soporte para múltiples lenguajes de programación
- 53.GNU:** es un sistema operativo de tipo Unix desarrollado por y para el Proyecto GNU y auspiciado por la Free Software Foundation
- 54.Cifrado AES:** Advanced Encryption Standard, también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos
- 55.ODS:** es un contenedor datos activos, es decir, está diseñado para integrar datos de múltiples fuentes con los que realizar operaciones adicionales en los propios datos
- 56.STG:** empresa de software especializada en soluciones de alto rendimiento de negocio a negocio
- 57.DDS:** es un formato para almacenar datos informáticos derivados de cinta de audio digital (DAT)
- 58.Microstrategy:** es una compañía que ofrece software OLAP, de inteligencia de negocio y de informes para empresas. El software de MicroStrategy permite crear informes y análisis de datos almacenados en una Base de datos relacional y otras fuentes
- 59.OBIEE:** Oracle Business Intelligence Enterprise Edition Plus, también denominado OBI EE Plus, es conjunto de herramientas de inteligencia de negocios y cuenta con sala de ex inteligencia empresarial Siebel Systems y Hyperion ofrendas de inteligencia de negocios Soluciones de Oracle Corporation
- 60.String:** cadena de caracteres
- 61.Código batch:** un archivo batch es un archivo de procesamiento por lotes. Se trata de archivos de texto sin formato, guardados con la extensión .BAT que contienen un conjunto de instrucciones MS-DOS. Cuando se ejecuta este archivo, las órdenes contenidas son ejecutadas

en grupo, de forma secuencial, permitiendo automatizar diversas tareas

62.Applets: es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo en un navegador web. El applet debe ejecutarse en un contenedor, que le proporciona un programa anfitrión, mediante un plugin,¹ o en aplicaciones como teléfonos móviles que soportan el modelo de programación por "applets"

63.HTTP: es el protocolo usado en cada transacción de la World Wide Web

64.Javascript: (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico