

Memoria TFC.

Web para la gestión eficiente de asignación de medios para incorporaciones de personal en empresas



NOMBRE ESTUDIANTE:	MARIO GRANERO BURILLO
ESTUDIOS:	ITIG
NOMBRE CONSULTOR:	FERRÁN PRADOS CARRASCO
FECHA DE ENTREGA:	21 DE JUNIO DE 2015

1 PARTE INICIAL

1.1 RESUMEN

“Web para la gestión eficiente de asignación de medios para incorporaciones de personal en empresas”

El siguiente proyecto tiene como objetivo el diseño de una aplicación web útil para el trabajo colaborativo. Se han aplicado distintos métodos para conseguir el objetivo, pero principalmente se han utilizado técnicas de programación y de diseño de estructuras de datos así como de servicios de virtualización para la publicación de la web. El desarrollo del proyecto ha arrojado distintas conclusiones, cabe destacar que se han identificado numerosas tecnologías que permiten la elaboración de desarrollos en un modo colaborativo, y que gran parte del conocimiento hoy en día es compartido en internet.

Las palabras clave del proyecto son:

Web, Cloud, Java, MySQL, APP, AWS, MVC y JSP.

El título que proponemos en inglés al proyecto es:

Web app for efficient management of resources allocation for the staff movements in corporations

A continuación se muestra una nube de tags con la mayoría de la documentación manejada:

PARTE INICIAL	1
1.1. RESUMEN.....	1
1.2.	2
1.3. GLOSARIO	5
1.4. PREFACIO	6
1 PARTE INICIAL	7
2.1. INTRODUCCIÓN	7
2.2. NÚCLEO DE LA MEMORIA.....	8
2.2.1 <i>Introducción.....</i>	8
2.2.2 <i>Justificación del tfc y contexto en el que se desarrolla: punto de partida y aportación.....</i>	8
2.2.3 <i>Objetivos del TFC.....</i>	9
2.2.4 <i>Enfoque y método seguido.....</i>	9
2.2.5 <i>Planificación del proyecto.....</i>	9
2.2.6 <i>Productos obtenidos.....</i>	10
2.2.7 <i>Breve observación de los próximos capítulos.....</i>	11
2.3. TECNOLOGÍAS APLICADAS	11
2.3.1 <i>Estructura de datos.....</i>	11
2.3.2 <i>Desarrollo de software.....</i>	12
2.3.3 <i>Infraestructura y despliegue.....</i>	13
2.4. FASES DEL PROYECTO	15
2.4.1 <i>Análisis.....</i>	15
Descripción de requisitos.....	17
Identificación de actores y casos de uso.....	17
Diagrama de casos de uso	18
Descripción de los casos de uso.....	20
Diseño de la base de datos	29
2.4.2 <i>Diseño.....</i>	30
Diagrama de actividad	30
Vista de usuario	31
2.4.3 <i>Implementación.....</i>	36
Estructura de tablas.....	36

Lógica de control.....	37
Entrega al usuario	39
Pruebas de carga	40
Mejoras propuestas.....	40
2.5. CONCLUSIONES	42
2.6. REFERENCIAS	44
3. PARTE FINAL.....	48
3.1. ANEXOS	48
3 ÍNDICE DE CONTENIDOS E ÍNDICE DE FIGURAS.....	ERROR! NO S'HA DEFINIT L'ADREÇA D'INTERÈS.
4 DEDICATORIA Y AGRADECIMIENTOS.....	ERROR! NO S'HA DEFINIT L'ADREÇA D'INTERÈS.
4.1.1 Dedicatoria	50
4.1.2 Agradecimientos.....	50

Se utilizan muchos términos técnicos, sobre todo son acrónimos que resumen o bien un estándar o bien un producto.

ECLIPSE: Herramienta de entorno de desarrollo.

AWS: Amazon Web Services. Más adelante, explicaremos más a fondo cómo ha participado en el proyecto.

MySQL: Entorno “open source” para la gestión de bases de datos relacionales.

Open source: Tipo de software que adquiere dicho nombre dado que a efectos de código y licencias son abiertos.

Frameworks: Conjunto de herramientas y utilidades que permiten la reutilización de código en conjuntos ya prediseñados o por lo menos una reutilización del mismo con paquetes y configuraciones prediseñadas.

CRUD: Acrónimo inglés que significa “Create, Read, Update y Delete. Que en su traducción son las acciones básicas a realizar sobre cualquier aplicación que maneje datos: creación, lectura, edición y borrado.

JSP: Extensión registrada por el propio Oracle que define un tipo de archivos del lenguaje de programación Java con orientación a la presentación en páginas web.

MVC: Model, View, Control. Metodología de programación web y programación orientada a objetos que permite proteger el contenido y código para poder utilizar las capas de presentación con seguridad.

IaaS, PaaS, SaaS: Modelos de servicios tecnológicos orientados a servicios en la nube ofrecidos como servicio. Significan Infrastructure as a service, Platform as a service y Service as a service. Con estos tres conjuntos de servicios se podría ofrecer servicios tecnológicos generalmente por coste variable en función del uso a prácticamente cualquier empresa atendiendo sus necesidades.

Página web responsive: Página web que dado su diseño, busca la correcta y óptima visualización del contenido independientemente del dispositivo que la esté abriendo. Esto hoy en día es vital dado el elevado número de tipos (smartphones, tablets, pcs, ...)

1.4 PREFACIO

Este proyecto se basa en muchas tecnologías y muchas de ellas bastante modernas. El interés en los sistemas virtualizados, las aplicaciones web y su alcance global a golpe de “click” han hecho que con este proyecto se dedique muchas horas a la elección de la mejora de las vías según las necesidades y los conocimientos.

La continua tendencia de los servicios en la nube y prácticamente la totalidad de las tecnologías ofertadas “as a “service”, despiertan un interés general sobre su uso.

2 PARTE INICIAL

2.1 INTRODUCCIÓN

En los últimos años hemos presenciado como los cambios de trabajo cada vez son más habituales si los comparamos con años anteriores. Dicen los expertos que cada vez será más habitual los cambios de profesión dado que la tecnología impulsa tanto el resto de sectores que tenemos la educación desactualizada nada más salir una nueva titulación.

Metido en contexto nos situamos en el mundo empresarial y observamos que atender estas altas bajas (de personal) se convierte en una tarea cada vez más recurrente y cada vez hay que hacerlo más rápido para ser más competitivo.

Dentro de los marcos legales y regulatorios, existen también ciertas connotaciones que hay que tener en cuenta y que cada vez son más exigentes. Pongo como ejemplo los usuarios lógicos de una compañía o la situación reputacional y de seguridad respecto a que se pierda un equipo informático con datos sensibles de la compañía.

i) **Diseñaremos e implantaremos lo básico:**

El módulo básico consistirá en dar de alta un nuevo empleado en los sistemas y asignar aquellos servicios y medios que le permitan ponerse a trabajar de una manera fácil y rápida. Se implicarán a los intervinientes más básicos.

ii) **Enumeraremos los módulos personalizables adicionales** que podrían ser útiles en las empresas:

El modelo económico del proyecto (FREEMIUM) permite utilizar gratuitamente el módulo básico (i) y posteriormente adquirir módulos adicionales personalizados (ii).

2.2.1 INTRODUCCIÓN

El proyecto queda **definido** por una herramienta web que permita la introducción de los datos básicos de un trabajador y que la información sea compartida por todos los intervinientes para ser gestionadas las peticiones de medios asociadas a cada trabajador. Existirán distintos roles de usuarios dentro de la herramienta:

- Los que realizarán las peticiones de medios que sean oportunas o introducen datos (*Peticionarios*),
- Los que comunicarán cuándo han sido resueltas (*Resolutores*),
- Los que administrarán el cumplimiento y los distintos participantes del proceso (*controladores y administradores respectivamente*).

El **objetivo** de la herramienta será limitar el habitual e ineficiente uso del correo electrónico para dichas acciones, mejorar los tiempos de puesta en funcionamiento de un trabajador en su empresa y sobre todo, que los intervinientes directos e indirectos trabajen de un modo más eficiente. También cabe destacar que si existiera un resolutor subcontratado, podría servir también para medir su nivel de servicio acordado.

Como **alcance** se definen las funciones básicas y los flujos más genéricos, quedando definidos 2 peticionarios y 4 resolutores, de manera que sirva de ejemplo y permita una evaluación y extrapolación a la empresa e identificar posibles eficiencias. Tampoco quedan incluidas las conexiones con otras herramientas o sistemas como (Correo, ERPs, CAUs, Ticketing...) siendo estas fases opcionales a desarrollar a medida y con posterior coste.

2.2.2 JUSTIFICACIÓN DEL TFC Y CONTEXTO EN EL QUE SE DESARROLLA: PUNTO DE PARTIDA Y APORTACIÓN.

Al margen de cubrir una necesidad de negocio y la que se prevé en el futuro se justifica este proyecto por su alto grado de aprendizaje en muchos entornos de trabajo que han sido estudiados en las distintas asignaturas de la carrera y además, son vitales en las empresas de tecnología de hoy en día.

El contexto de la virtualización y de los modelos de servicios en cloud, es una realidad y las nuevas tecnologías van encaminadas a ello.

Es por ello que las IaaS, PaaS y SaaS tienen que ir de la mano del habitual perfil del programador y que sea capaz de ampliar sus conocimientos para intentar trabajar en otros campos y le permitan explotar sus cualidades al máximo.

Hoy en día, saber programar está bien, pero conseguir implementar todo lo que necesita para que esté accesible a todo el mundo, está mejor.

2.2.3 OBJETIVOS DEL TFC

- ✓ Los objetivos principales del proyecto han sido tocar el máximo número de tecnologías al realizar un desarrollo.
- ✓ Gestionar bases de datos relacionales, creación, modificación, eliminación y acceso.
- ✓ Gestionar la arquitectura de las bases de datos.
- ✓ Gestionar entornos virtuales, instancias, bases de datos, servidores virtuales...
- ✓ Desplegar aplicaciones de entornos de desarrollo a entornos de preproducción.

2.2.4 ENFOQUE Y MÉTODO SEGUIDO

Hemos seguido técnicamente una metodología de ciclo de vida en cascada retroalimentada donde las fases se caracterizaban por ser la entrada de las siguientes. Además, y dadas las mejoras aplicadas en la parte final hemos permitido el retroceso a fases anteriores. En paralelo hemos llevado distinto trabajo de formación e investigación.

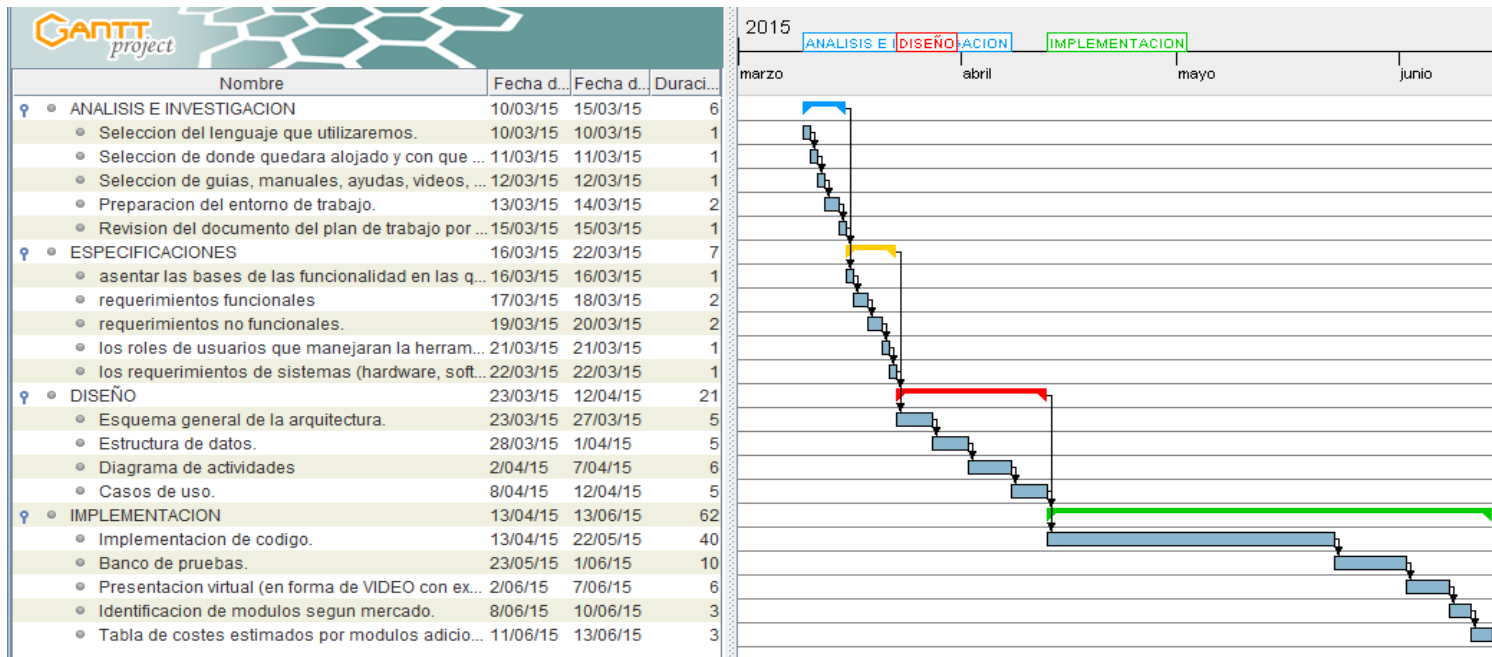
2.2.5 PLANIFICACIÓN DEL PROYECTO

El proyecto se ha separado en tres grandes partes,

1. Por un lado la parte de elección del proyecto, documentación y análisis de requerimientos. Se han realizado distintas entrevistas a distintas personas que trabajan en distintas empresas para comprender si la necesidad se replicaba en distintos entornos y sectores.

2. Por otro lado, y con el ánimo de poder ofrecer escalabilidad a la solución elegida, se ha puesto mucho esmero en la elección de las infraestructuras, entornos de desarrollo y entornos de virtualización.
3. Por último, la parte de implementación, de desarrollo, pruebas y de despliegue.

Los dos últimos puntos han llevado prácticamente el 85% del proyecto dado que se ha tenido que consultar muchas fuentes, revisiones y foros para dar con el esquema final.



2.2.6 PRODUCTOS OBTENIDOS

- ✓ Paquete de desarrollo WAR, que es una carpeta librería JAVA que contiene el desarrollo.
- ✓ Acceso y URL a la aplicación WEB que se ha diseñado.

- ✓ Clave y password para el acceso a la misma.
- ✓ Ficheros de configuración de esquemas de Entidad Model Relación de MySQL, así como las instancias sql de las consultas llevadas a cabo.

Más adelante, en el punto de entrega al usuario se enumeran y detallan los productos finales obtenidos.

2.2.7 BREVE OBSERVACIÓN DE LOS PRÓXIMOS CAPÍTULOS

Los próximos puntos se ha separado en dos grandes bloques, uno con las tecnologías elegidas y otro con el desarrollo del proyecto y las fases habituales de los mismos enfocados al desarrollo de software. Según las recomendaciones, se ha tratado de seguir una serie de puntos lógicos.

2.3 TECNOLOGÍAS APLICADAS

2.3.1 ESTRUCTURA DE DATOS

MySQL.

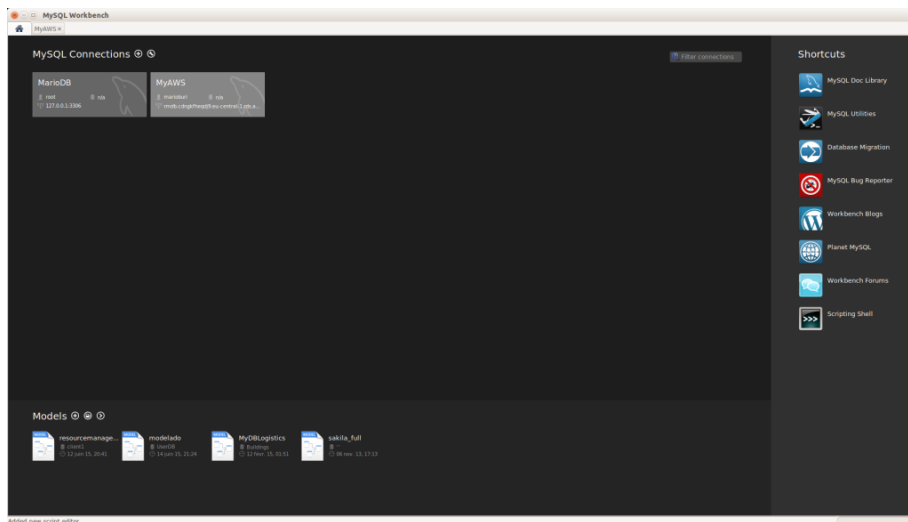
Su definición según Wikipedia es:

MySQL es un [sistema de gestión de bases de datos relacional](#), [multihilo](#) y [multiusuario](#) con más de seis millones de instalaciones.¹ [MySQL AB](#) — desde enero de 2008 una subsidiaria de [Sun Microsystems](#) y ésta a su vez de [Oracle Corporation](#) desde abril de 2009— desarrolla MySQL como [software libre](#) en un esquema de licenciamiento dual.

Durante las asignaturas en las que se trabajó con bases de datos, se trabajó con Postgre, pero ¿Por qué se ha elegido?

Existen numerosos motivos, pero el principal ya se adelantaba en puntos anteriores, la escalabilidad y compatibilidad con distintos sistemas de manera que en el caso que la herramienta se convirtiera en grande, fuésemos capaces de seguir ofreciendo servicio.

Por su licencia, por ser fácil de usar, por compatibilidad con AWS y sobre todo por la capacidad de portabilidad entre sistemas.



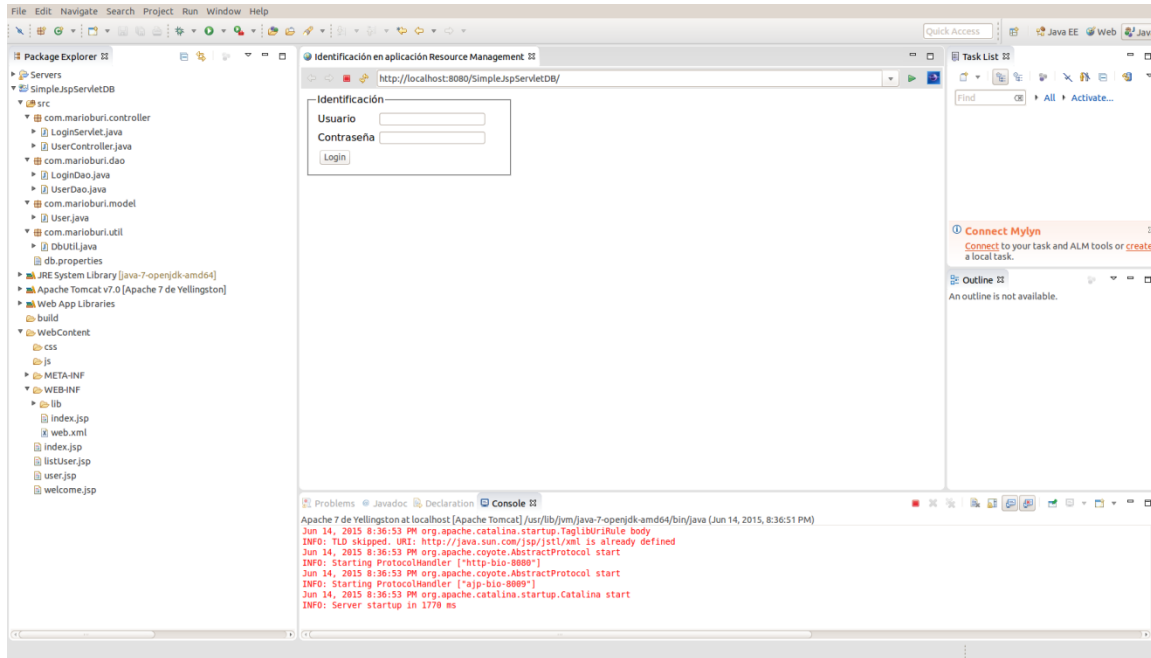
(Imagen de MySql workbench en su página de inicio donde se ven conexiones a bases de datos, locales o virtuales así como los modelos. Destacar que modelos y conexiones se pueden tratar de un modo independiente pero que se pueden llegar a relacionar en cualquiera de los casos con “forward engineer o reverse engineer”)

2.3.2 DESARROLLO DE SOFTWARE

JAVA con IDE Eclipse.

El lenguaje de programación orientado a objetos resulta interesante y óptimo para el desarrollo de las aplicaciones web. En su esquema de MVC, se han utilizado distintos valores de las clases de conexión http, así como distintos servlets que han permitido abstraer la capa de controlador con las bases de datos.

Netbeans es una opción bastante acertada así como otra que se analizó algo más tarde: IntelliJ.



(Entorno de desarrollo web habitual con eclipse, con los mensajes del servidor TOMCAT arrancado y emitiendo su consola para identificar posibles fallos en tiempo de ejecución)

En cuanto a la metodología seguida para llevar a cabo las fase de programación, hemos realizado el análisis de arriba abajo, descomponiendo desde el alto al bajo nivel. Dicha metodología se denomina *Topdown*.

2.3.3 INFRAESTRUCTURA Y DESPLIEGUE

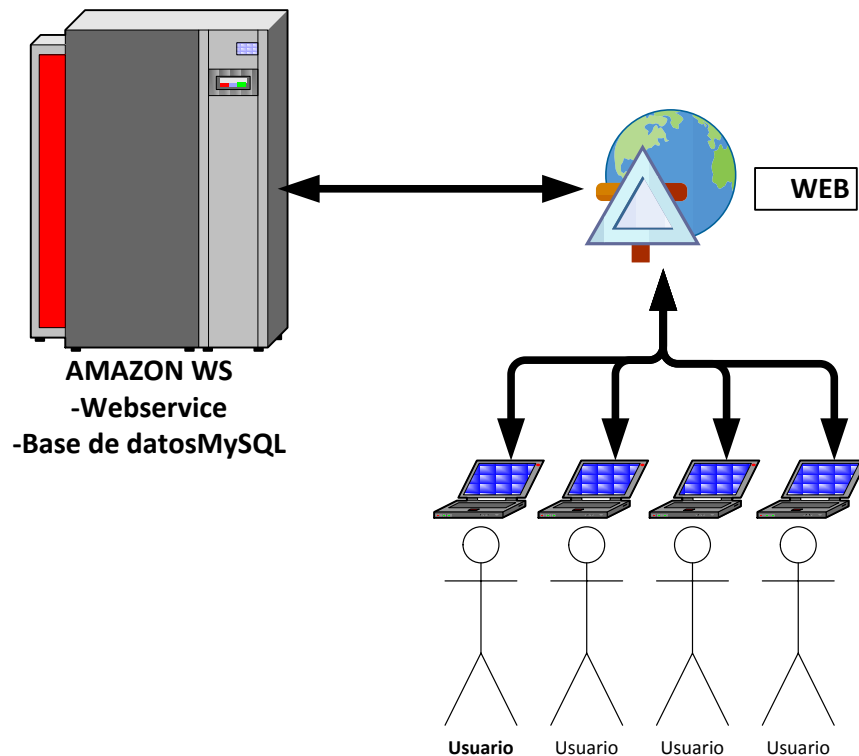
Amazon Web Services ha sido minuciosamente analizada con Azure de Microsoft y con alguna más, pero la versatilidad y facilidad que ofrece AWS no se ha encontrado en otras.

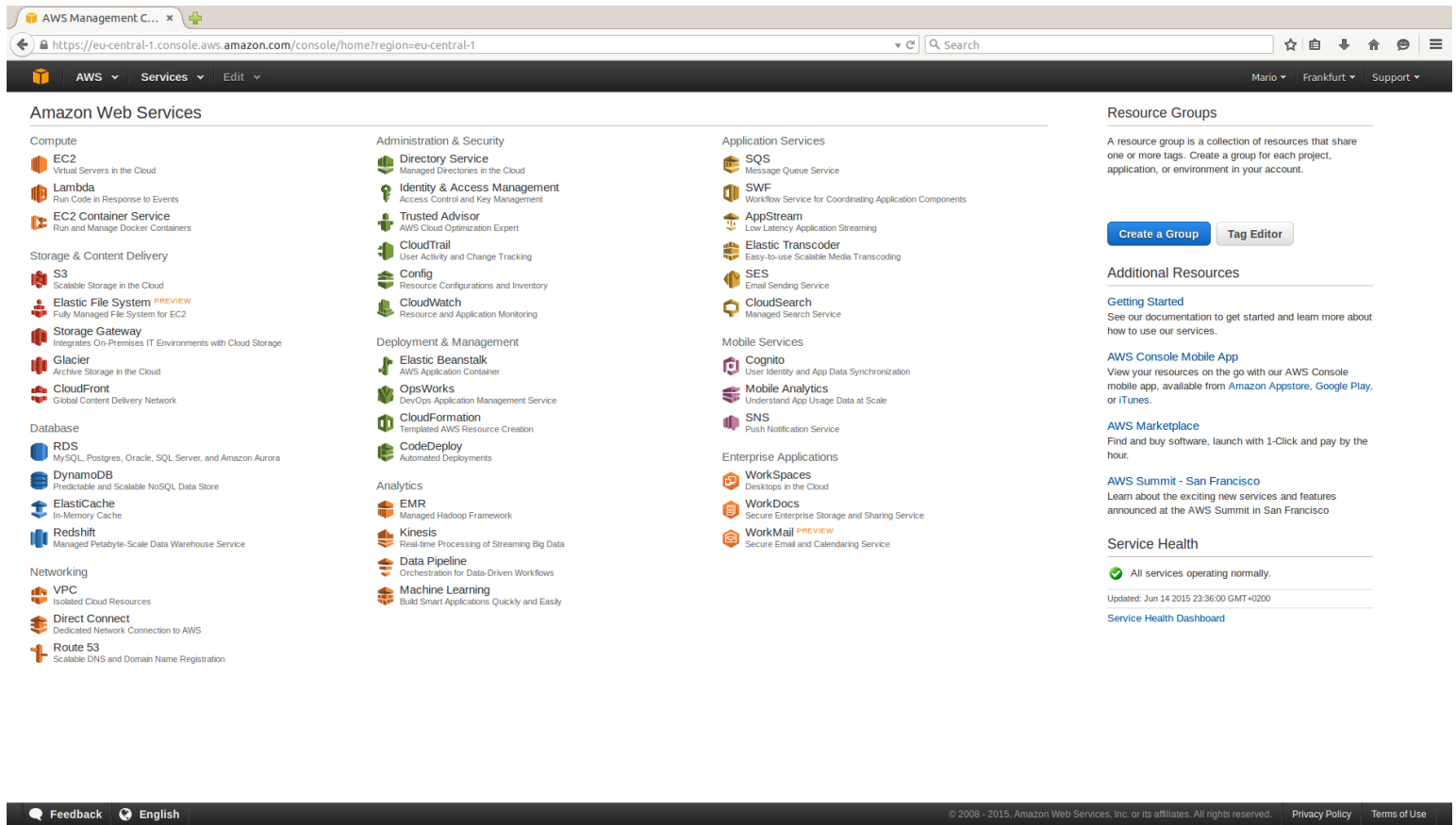
AWS basa su modelo de cloud computing en un catálogo de subproductos muy amigables que categoriza oportunamente.

Para albergar con una instancia la base de datos relacional, se ha optado por el servicio RDS. Se ha desplegado una instancia compatible con MySQL y posteriormente se ha migrado la estructura y datos de la base de datos gracias al asistente del Workbench de MySQL.

Por otro lado, se ha desplegado una instancia de procesamiento EC2 con lo más básico de momento y un sistema operativo Linux del propio Amazon co 64 bits.

Por último se ha elegido el producto Elastic Beanstalk que permite desplegar aplicaciones eligiendo la instancia de proceso y permitiendo que se conecten desde cualquier lugar del mundo gracias a su opción de url pública y fija.





(Catálogo de productos de la consola de AWS)

2.4 FASES DEL PROYECTO

2.4.1 ANÁLISIS

Se analiza que la aplicación web tendrá que contar con los siguientes subsistemas. Será después cuando se analicen esos subsistemas.

Subsistema de administración

Este subsistema estará compuesto por todos los servicios y funciones que permitan la administración de los roles y usuarios, así como su mantenimiento posterior. En las factorías es comúnmente conocido como mantenimiento de las aplicaciones o *back-office*.

Este subsistema (salvo la creación y asignación del primer administrador que será genérico y documentado) estará completamente disponible para el rol de administrador.

Los usuarios y roles que se vayan creando estarán en una base de datos.

Subsistema de control de accesos

Controlará las distintas visibilidades en función de los usuarios creados en el sistema. Además de controlar los accesos en el sistema (log-ins). La visibilidad estará disponible en la base de datos. Tendrá las funciones básicas de preguntar si un usuario tiene permisos o de hacer log-in;

Subsistema de almacenamiento

Este subsistema no requiere de mucha explicación, se encargará de cómo y dónde se guardan los datos que se van usando y cómo luego pueden ser consultados o modificados.

Subsistema de control y consulta

Este subsistema está pensado para que se pueda realizar el control de calidad además de otras consultas que permitan realizar análisis para la mejora continua de los procesos. Si consiguiésemos ser todo lo rigurosos que queremos, realizaremos el tracking de cualquier cambio de estado o envío de información de manera que podremos establecer tiempos de entrega y el control de los tiempos de cada participante.

Así mismo, y con el ánimo de cumplir las múltiples auditorías que existen, también quedarán documentadas las bajas para que se cumplan las leyes sobre protección de datos.

2.4.1.1.1 DESCRIPCIÓN DE REQUISITOS

En la parte básica (FREEMUM), que es la que se aborda en el proyecto, se detallan los siguientes requerimientos que finalmente se pueden comprobar junto con el cliente:

-Permitir la administración propia de los usuarios con algún perfil administrador y no depender de ningún servicio de mantenimiento o administración.

-Permitir que el volumen sea dinámica. A priori y dado que monitorizaremos el estado de nuestros servicios AWS, podremos ofrecer más capacidad tanto de procesamiento como de almacenamiento si las necesidades de los clientes varían.

-Permitir que se pueda usar desde cualquier dispositivo. Aunque no se ha confeccionado un entorno “user friendly” y páginas “responsive”, la aplicación se puede utilizar desde cualquier dispositivo, desde cualquier sitio con conexión a internet.

-No requiere de ninguna conexión que los firewall de las empresas puedan bloquear. Ningún tráfico será cortado dado que todo es web, los encargados de realizar las comunicaciones con las bases de datos de amazon serán ellos mismos gracias a los servlets.

-Utilización de los roles mínimos para una correcta asignación de medios de los empleados.

-Personalizado. Que los menús sean personalizados en función del rol. Los roles permiten identificar las consultas y acciones que podrá desarrollar cada usuario. En las sesiones http se define el rol y luego la lógica de la programación y los controladores apuntarán a vistas de las bases de datos personalizadas para cada uno de ellos. Ello ha evitado mucha de la programación y agilice mucho la respuesta y performance de la misma.

2.4.1.1.2 IDENTIFICACIÓN DE ACTORES Y CASOS DE USO

Nos hemos basado en las estructuras más sencillas de las empresas tienen varios puntos en común, a saber:



Administradores: Serán quienes realicen la administración de los usuarios y programen junto a nuestro equipo de analistas las propiedades y flujograma que seguirán en cada caso, dado que no siempre serán actividades secuenciadas.



Recursos Humanos: Ellos, como en casi todas las empresas, inician el proceso de alta de los empleados, así que darán de alta y de baja los empleados.



Facilities: En este colectivo de usuarios incluimos a los que se encargarán de atender todas aquellas peticiones a nivel de inmuebles, de puesto, de acceso a los edificios y de servicios del empleado.

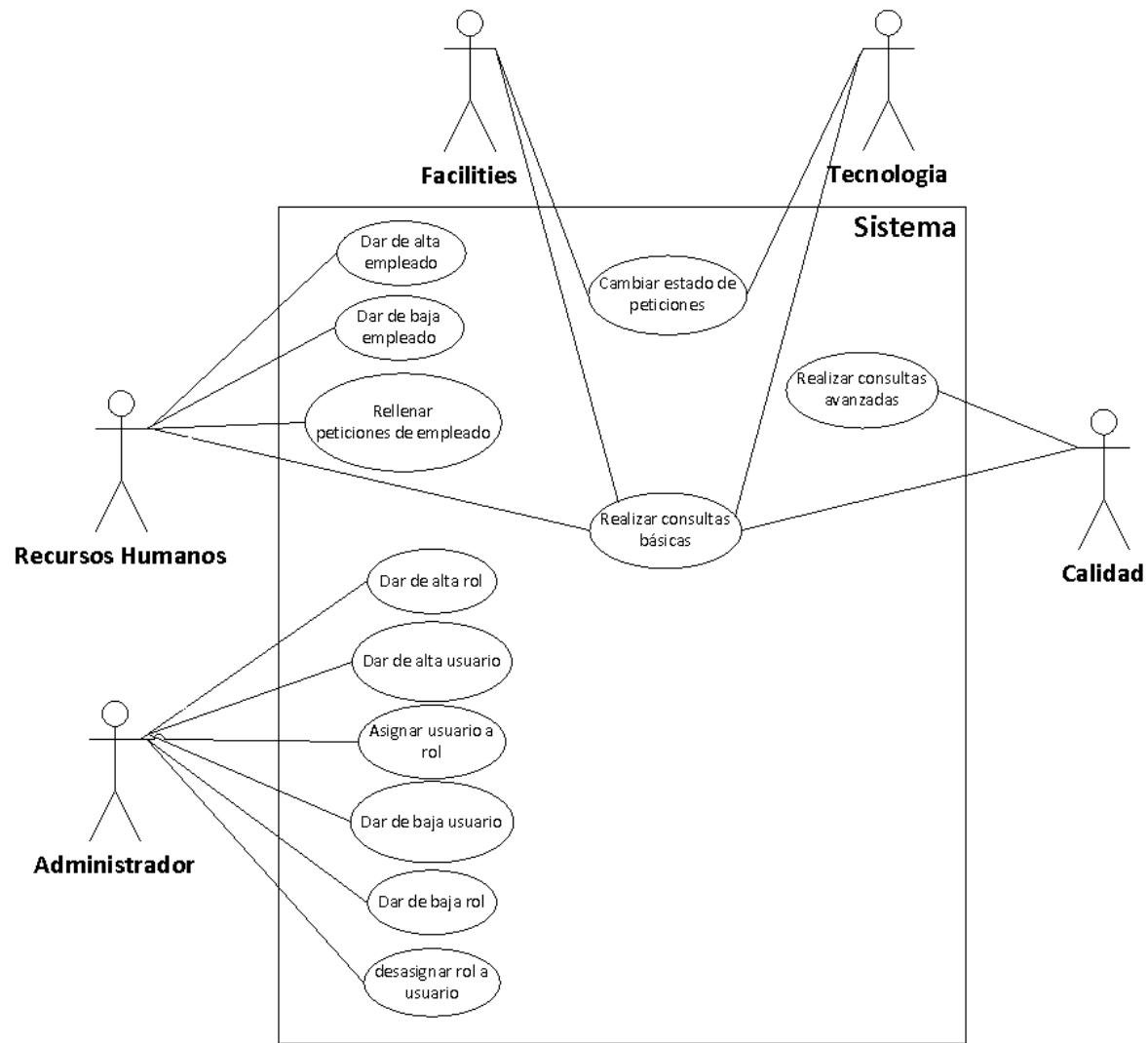


Tecnología: En este colectivo de usuarios incluimos a los que se encargarán de atender todas aquellas peticiones a nivel de tecnología, de hardware, de acceso a los sistemas y de software.








Calidad: En este colectivo quedarán los que controlan los tiempos y la calidad de las resoluciones. Controlan el proceso completo y realizan estadísticas e informes.

2.4.1.1.3 DIAGRAMA DE CASOS DE USO



(Diagrama con los distintos actores que hemos hecho coincidir con los roles)

Icono	Roles	Tareas / Actividades
	Administradores	Dar de alta rol, Dar de alta usuario, Asignar usuario a rol, Dar de baja usuario, Dar de baja rol, Desasignar rol a usuario.
	Recursos Humanos	Dar de alta empleado, Dar de baja empleado. Rellenar peticiones de empleado. Realizar consultas básicas.
	Facilities	Cambiar estado de peticiones. Realizar consultas básicas.
	Tecnología	Cambiar estado de peticiones. Realizar consultas básicas.
	Calidad	Realizar consultas básicas. Realizar consultas avanzadas.

A continuación se detallan las fichas de los casos de uso:

FICHA DE CASO DE USO	
ID	1
NOMBRE	Dar de alta empleado
DESCRIPCIÓN	Se carga un nuevo empleado a la herramienta con los atributos y necesidades del mismo.
FLUJO NORMAL	
ACTORES	Usuarios con el rol de Recursos humanos
PRECONDICIONES	El usuario debe poseer los privilegios necesarios
ACTIVACIÓN	El usuario ingresa un nuevo empleado
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se ingresan los datos 3- El usuario puede acceder a rellenar peticiones del empleado.
POSTCONDICIONES	Hay un nuevo empleado en la herramienta
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	3- Se cancela el alta
POSTCONDICIONES	En la herramienta hay la misma cantidad de empleados
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-
FICHA DE CASO DE USO	
ID	2

NOMBRE	Dar de baja empleado
DESCRIPCIÓN	Se elimina un empleado a la herramienta
FLUJO NORMAL	
ACTORES	Usuarios con el rol de Recursos humanos
PRECONDICIONES	El usuario debe poseer los privilegios necesarios El empleado debe existir previamente
ACTIVACIÓN	El usuario elimina un empleado existente
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se busca el empleado 3- Se confirma la acción
POSTCONDICIONES	Hay un empleado menos en la herramienta
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	3- Se cancela la baja
POSTCONDICIONES	En la herramienta hay la misma cantidad de empleados
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

FICHA DE CASO DE USO	
ID	5
NOMBRE	Dar de alta usuario
DESCRIPCIÓN	Se carga un nuevo usuario en la herramienta.
FLUJO NORMAL	
ACTORES	Administrador
PRECONDICIONES	El usuario debe poseer los privilegios necesarios El usuario no existe previamente o no está activo.
ACTIVACIÓN	El usuario ingresa un usuario
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se ingresan los datos 3- Se confirman los datos
POSTCONDICIONES	Hay un nuevo usuario en la herramienta
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	3- Se cancela el ingreso
POSTCONDICIONES	Hay el mismo número de usuarios.
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

FICHA DE CASO DE USO	
ID	6
NOMBRE	Asignar usuario a rol
DESCRIPCIÓN	Se asigna un rol a un usuario
FLUJO NORMAL	
ACTORES	Administrador
PRECONDICIONES	El usuario debe poseer los privilegios necesarios El usuario debe existir previamente. El rol debe existir previamente.
ACTIVACIÓN	El usuario tiene asignado un nuevo rol
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se busca y selecciona el usuario. 3- Se busca y selecciona rol. 4- Se confirman los datos
POSTCONDICIONES	Hay un nuevo usuario asignado a un rol.
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	<ol style="list-style-type: none"> 2- Se cancela la búsqueda de usuario o no existe. 3- Se cancela la búsqueda de rol o no existe. 4- Se cancela el ingreso.
POSTCONDICIONES	Ningún usuario es modificado.
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

FICHA DE CASO DE USO	
ID	7
NOMBRE	Dar de baja un usuario
DESCRIPCIÓN	Se da de baja un usuario de la herramienta
FLUJO NORMAL	
ACTORES	Administrador
PRECONDICIONES	El usuario debe poseer los privilegios necesarios El usuario a eliminar debe existir previamente y estar activo.
ACTIVACIÓN	El usuario elimina un usuario.
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se busca y selecciona el usuario. 3- Se confirman los datos
POSTCONDICIONES	Hay un usuario activo menos en la herramienta
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	3- Se cancela el ingreso
POSTCONDICIONES	En la herramienta hay el mismo número de usuarios activos.
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

FICHA DE CASO DE USO	
ID	8
NOMBRE	Desasignar rol a usuario-
DESCRIPCIÓN	Se cambia un rol a un usuario
FLUJO NORMAL	
ACTORES	Administrador
PRECONDICIONES	El usuario debe poseer los privilegios necesarios El usuario debe existir previamente. El usuario debe estar asignado a un rol.
ACTIVACIÓN	El usuario no tiene asignado un rol
DESCRIPCIÓN	1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se busca y selecciona el usuario. 3- Se confirman los datos
POSTCONDICIONES	Hay un nuevo usuario sin rol asignado
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	3- Se cancela el ingreso.
POSTCONDICIONES	No se desasigna ningún rol a ningún usuario
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

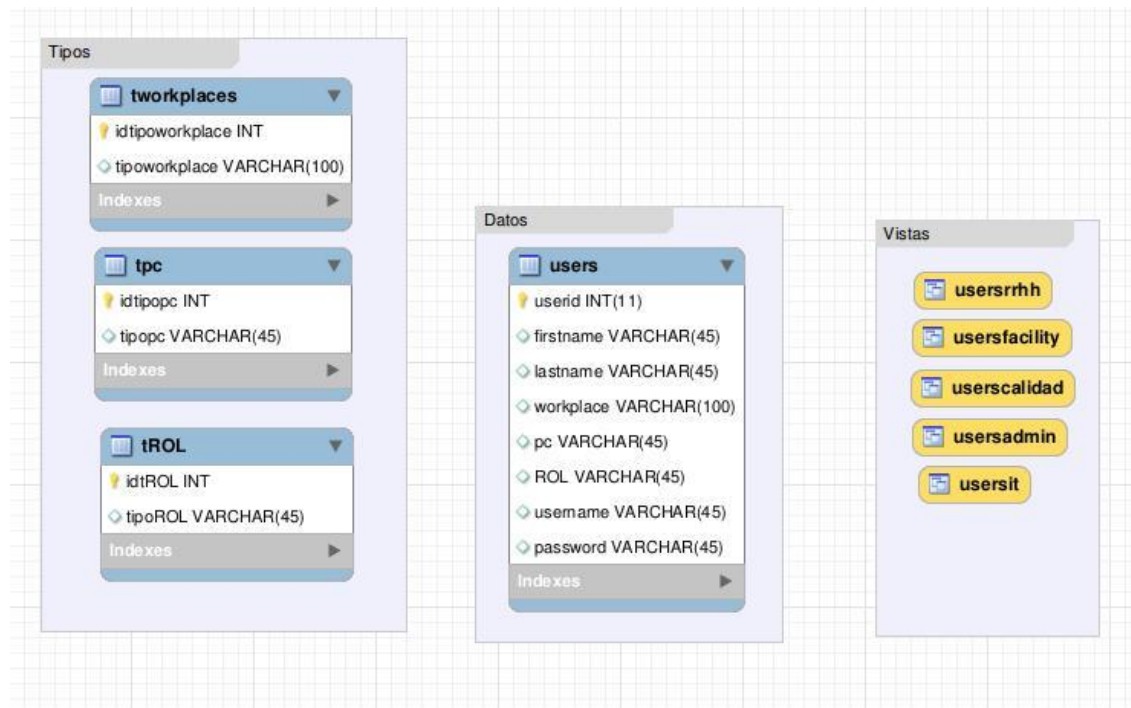
FICHA DE CASO DE USO	
ID	9
NOMBRE	Cambiar estado de peticiones
DESCRIPCIÓN	Se cambian estado de determinadas peticiones
FLUJO NORMAL	
ACTORES	Facilities, Tecnología
PRECONDICIONES	El usuario debe poseer los privilegios necesarios El empleado tiene que tener peticiones abiertas.
ACTIVACIÓN	Una o varias peticiones son modificadas
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Busca el empleado con peticiones pendientes 3- Se confirman los datos
POSTCONDICIONES	Hay estados de peticiones modificados.
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	<ol style="list-style-type: none"> 2- Se cancela la búsqueda de empleado o no existe. 3- No se modifica ningún estado. 4- Se cancela el ingreso.
POSTCONDICIONES	Ningún estado de peticiones es modificado.
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

FICHA DE CASO DE USO	
ID	10
NOMBRE	Realizar consultas avanzadas
DESCRIPCIÓN	Se devuelven consultas básicas bajo demanda
FLUJO NORMAL	
ACTORES	Calidad.
PRECONDICIONES	El usuario debe poseer los privilegios necesarios Deben existir datos en la base de datos.
ACTIVACIÓN	El usuario visualiza una consulta avanzada que ha realizado.
DESCRIPCIÓN	<ol style="list-style-type: none"> 1- El usuario ingresa a la pantalla que tiene dicho fin 2- Se confirma la búsqueda.
POSTCONDICIONES	Ningún dato es modificado. Se lanza una <i>query</i> .
FLUJO ALTERNATIVO 1	
DESCRIPCIÓN	3- Se cancela la búsqueda.
POSTCONDICIONES	No se realiza nada.
FLUJO ALTERNATIVO 2	
DESCRIPCIÓN	-

2.4.1.1.5 DISEÑO DE LA BASE DE DATOS

No hemos diseñado una estructura compleja dado que dentro del modelo MVC, por cada tabla hay que diseñar clases. En el futuro, si se eligiese la opción PREMIUM, confeccionaríamos más tablas para poder ofrecer el tracking temporal de las peticiones, interacciones entre peticionarios y resolutores, etc...

A continuación se inserta el modelo de la base de datos.



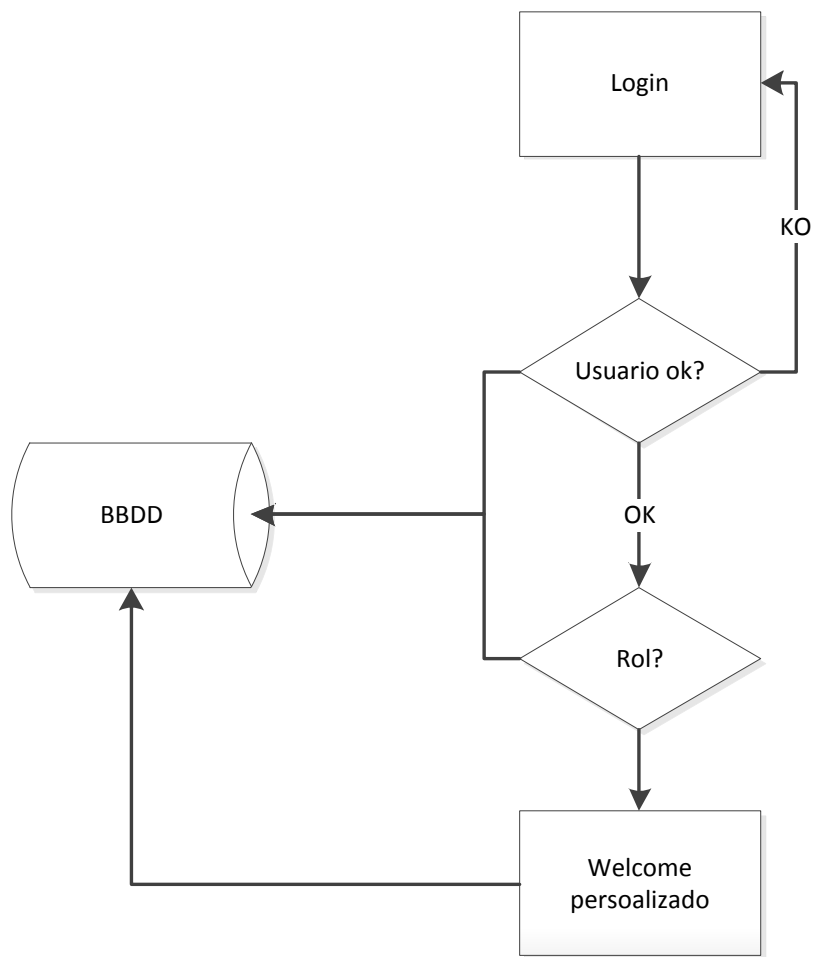
Como comentábamos en puntos anteriores, el uso de las vistas, ofrece una mejora sustancial en la velocidad de la aplicación y en evitar al abuso de líneas de código, de manera que su mantenimiento es más sencillo.

Los datos iniciales se cargaron a mano en MySQL que al conectarlo con el RDS de AWS, han permitido trabajar directamente durante las pruebas con la aplicación trabajando directamente con la base de datos.

2.4.2 DISEÑO

2.4.2.1 DIAGRAMA DE ACTIVIDAD

Durante el login, es donde se generan la mayoría de decisiones dado que primero comprueba que el usuario y contraseña son correctos, después averigua el rol y en función del rol mostrará las opciones disponibles y dejará siempre la opción de volver.



2.4.2.2 VISTA DE USUARIO

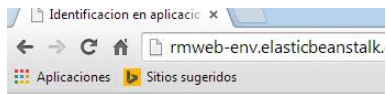
Login

Pantalla de acceso donde al abrir solicitará usuario y contraseña:

- Login correcto: mensaje informativo sobre el correcto acceso.



- Login incorrecto: mensaje informativo sobre error en el inicio de sesión.



Identificación de usuario

Usuario	<input type="text"/>
Password	<input type="password"/>



Aplicacion de Gestion de Recursos.TFC-UOC: Mario Granero
Burillo.Email: marioburi@gmail.com

Menú principal

Pantalla principal personalizada en función del usuario que muestra las distintas opciones que tiene el usuario, además de mostrar el usuario logado y el rol que tiene:



Sesion iniciada por: TFC con el rol de: admin

Opciones del administrador:

[Agregar](#)

[Listar](#)

[Salir](#)

Administrar

La primera pantalla de administración nos muestra el listado de Cliente / Usuario / Roles y las opciones de borrar/desactivar/activar a su lado así como marcar y desmarcar los roles asignados. Podrá filtrar rápidamente para localizar un usuario. También tendrá la opción de crear usuario.

(Ver imagen anterior con login de administrador)

Gestionar

La gestión se hace directamente desde el inicio. Se han pensado en las actividades cotidianas que hace cada rol para que sólo navegue por esas ventanas.

Consultas

Se personalizan las consultas de manera que cada rol sólo ve lo que realmente tiene que hacer, bien porque lo tiene pendiente (resolutores), bien porque todavía no está realizado (solicitantes).



The screenshot shows a web browser window with the address bar containing 'rmweb-env.elasticbeanstalk.com/UserController?act'. Below the browser, there is an illustration of four figures assembling colorful blocks. Underneath the illustration is a table with the following data:

Numero de empleado	Nombre	Apellidos	Puesto de trabajo	Action
275	Mario	Granero	Desk normal	Completar
290	Jacinto	Benavente	Despacho VIP	Completar
292	ddd	ddd	Despacho VIP	Completar

Below the table is a button labeled 'Volver'.

Consultas avanzadas

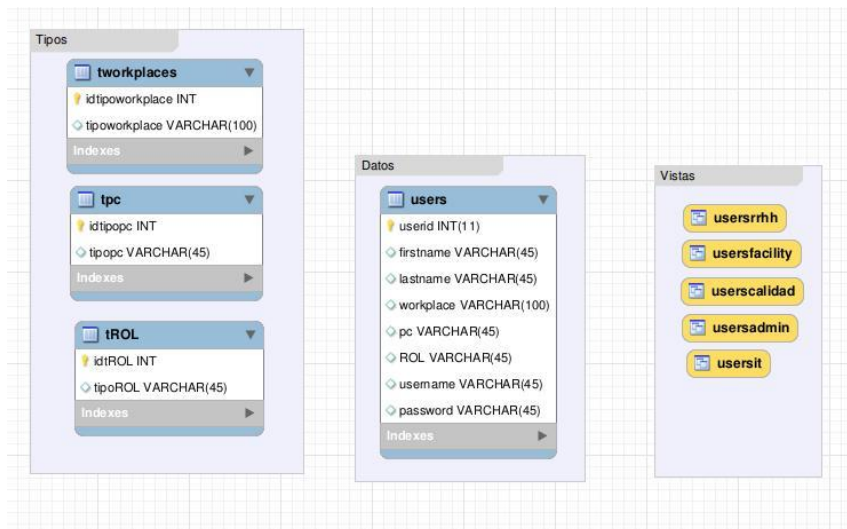
El rol de calidad, tendrá visión de toda la información para bien tramitar una auditoría bien medir la calidad de los procesos.

El rol de admin, observará únicamente los usuarios que tienen permisos en la herramienta, sus roles y sus contraseñas.

2.4.3 IMPLEMENTACIÓN

2.4.3.1 ESTRUCTURA DE TABLAS

Esta es la estructura que se diseñó. En los puntos anteriores se comentaba y justificaba el uso de tablas y vistas mejorando el rendimiento del acceso a las mismas desde la web.

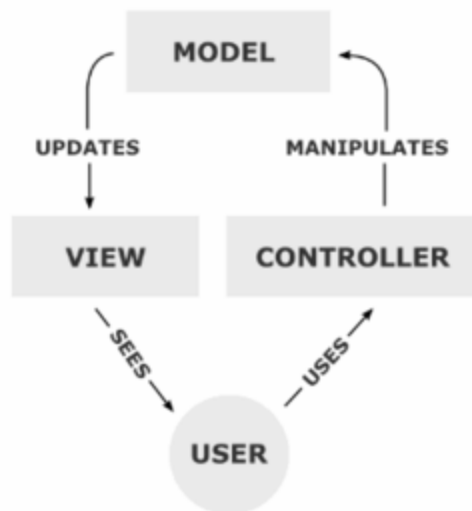


2.4.3.2 LÓGICA DE CONTROL

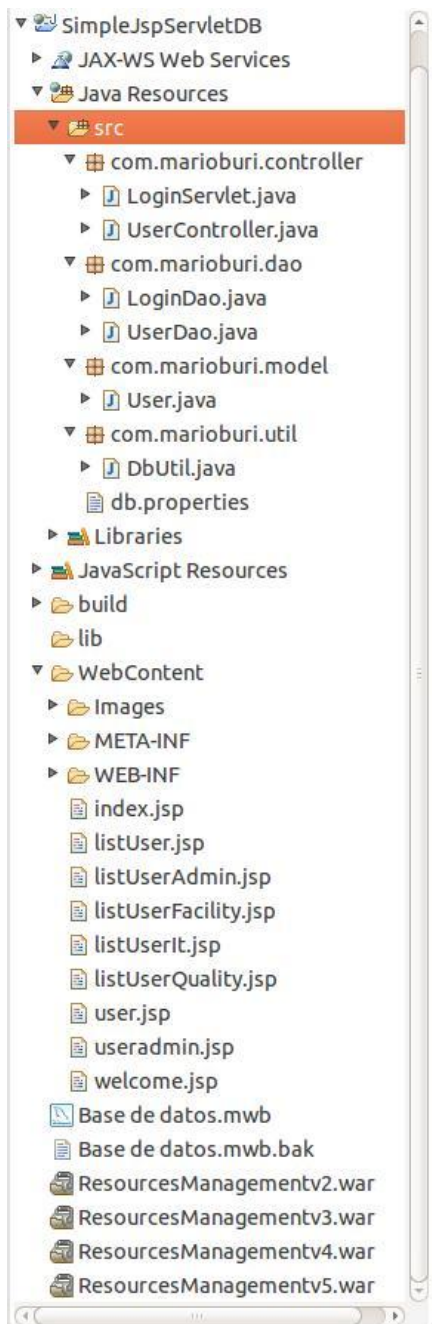
Según Wikipedia:

“El **modelo–vista–controlador (MVC)** es un patrón de [arquitectura de software](#) que separa los [datos](#) y la [lógica de negocio](#) de una aplicación de la [interfaz de usuario](#) y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres [componentes](#) distintos que son el **modelo**, la **vista** y el **controlador**, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.^{1 2} Este patrón de [arquitectura de software](#) se basa en las ideas de [reutilización de código](#) y la [separación de conceptos](#), características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.^{3 4}”

Es por ello



Por ello la estructura de clases y JSP en el proyecto de desarrollo queda justificado bajo el siguiente esquema:



2.4.3.3 ENTREGA AL USUARIO

La entrega de este proyecto consiste en un fichero zip que incluyen los 3 primeros puntos.

1.-Fichero de modelado de base de datos mwb de MySql:

Define la base datos que utilizaremos.

2.-Carpeta del proyecto JAVA:

Contiene las clases y librerías que se utilizan así como el código de los jsp que finalmente se muestran en la web.

3.-Fichero de despliegue final WAR:

Fichero de despliegue preparado para subir, desplegar y ser ejecutada por el servidor. Está optimizado para TOMCAT v7.

4.-URL y usuarios para pruebas:

<http://rmweb-env.elasticbeanstalk.com/>

Username	ROL	Password
tecnico	it	123uoc
facility	facility	123uoc
seleccion	rrhh	123uoc
control	quality	123uoc
administrador	admin	123uoc

5.-Memoria, presentación y video del proyecto.

El uso posterior y la explotación de la misma se haría en una URL distinta por cada cliente que contrate el servicio.

2.4.4 PRUEBAS DE CARGA

Se han realizado pruebas de carga tanto en la base de datos como en la web.

Se han probado bajo todos los usuarios de pruebas, con todas las acciones posibles sin encontrar fallo alguno por lo que la aplicación podrá salir a producción salvo que se encuentre algo más.

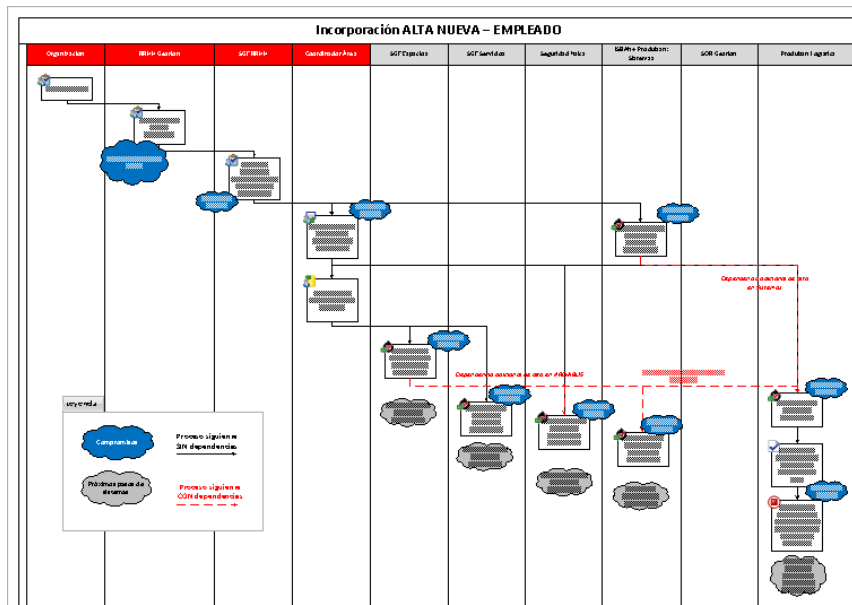


Numero de empleado	Username	ROL	Password	Action	
244	TFC	admin	UOC	Modificar	Eliminar
261	tecnico	it	123uoc	Modificar	Eliminar
262	facility	facility	123uoc	Modificar	Eliminar
263	seleccion	rrhh	123uoc	Modificar	Eliminar
264	control	quality	123uoc	Modificar	Eliminar
282	administrador	admin	123uoc	Modificar	Eliminar

2.4.5 MEJORAS PROPUESTAS

Se presentan las siguientes mejoras:

- 1.-Opción de solicitud de registro con workflow de aprobación por parte de un administrador.
- 2.-Servicio de notificaciones sobre peticiones pendientes de atender.



6.-Asignación de nuevo rol de DOMINIO O ENTIDAD de manera que podamos abarcar más clientes e incluso de cada cliente, su estructura departamental de modo que podamos asignar a quien corresponde y no a todos los grupos.

7.-Implementar avisos de indisponibilidad de la herramienta. Balancear la carga, establecer un plan de backup de la información.

8.-Permitir el tracking de tiempos por si tuviéramos empresas en modelo de servicios y ello pudiera confeccionar un contraste en los SLA's contratados con los mismos.

2.5 PRESUPUESTO

Ni el desarrollo, ni las horas dedicadas al proyecto han supuesto coste, no obstante para el caso de la infraestructura ofrecida por AWS sí que ha supuesto un pequeño coste, pero al ser tan pequeños los requerimientos no es representativo.

En el caso de llevar a cabo la versión PREMIUM sí que llevaríamos a cabo un estudio minucioso para ver cómo podríamos cobrar a los clientes dado que sobre la infraestructura habría que meterle un margen que permita cubrir los costes iniciales más los beneficios de explotación que considerásemos.

2.6 CONCLUSIONES

Las conclusiones del proyecto son las siguientes:

-La escalabilidad es muy importante cuando ofrecemos un producto FREEMIUM, dado que nos podemos enfrentar a un crecimiento elevadísimo de clientes y éstos han de ser tratados correctamente si tenemos intención de lograr algún cliente PREMIUM.

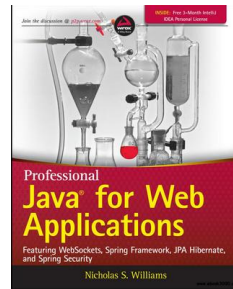
-La presentación es importante. Los usuarios finales, no comprenden qué hay detrás de las herramientas y dicha capa de abstracción debe permanecer siempre. Eso sí, en la versión de pago, hay que ofrecer una capa mucho más cuidada.

-El performance no puede ir empeorando a raíz que se incrementan los datos o los clientes. Es algo que debemos perseguir y lograr tanto con la estructura de datos, como con el código como con los servicios virtualizados. Los usuarios que vean que la aplicación va lenta, no contratarán nunca la versión de pago, aunque se le explique que dicha versión esos problemas se solucionan.

-Hay que ser prudente con los frameworks y hojas de estilo que se usan porque como mucho de ese contenido está ofrecido por terceros en páginas web que no conocemos, estamos ofreciendo un servicio y tenemos una dependencia. Se han observado que existen una multitud de frameworks, cuyas páginas declaradas, tienen caídas, o incluso están en entornos que puedan ralentizar nuestro servicio.

-Se podría ofrecer una versión lite que tan sólo ofrezca un webservice de manera que empresas que tengan sus propios entornos o intranets hagan uso de la lógica y las bases de datos. Ello puede mejorar la seguridad de accesos para sólo permitir dicho acceso desde sus proxies por ejemplo.

2.7 REFERENCIAS



“Java for web applications” Nicholas S. Williams editorial WROX A Wiley Brand



<http://www.lawebdelprogramador.com>



<http://www.ganttproject.biz/>



<http://camstudio.org/>



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



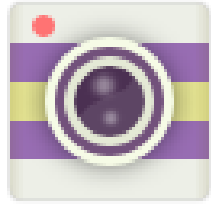
<http://stackoverflow.com/>



<https://netbeans.org/>



www.google.es



<https://launchpad.net/kazam>



<https://www.mysql.com/products/workbench/>



<http://aws.amazon.com/es/>



<https://github.com/>



<http://www.uoc.edu/>

Program Creek

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



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



<http://docs.oracle.com/javase/7/docs/api/>



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

3. PARTE FINAL

3.1 ANEXOS

No se trata ninguna información adicional, tan sólo se reitera en el material entregado:

La entrega de este proyecto consiste en un fichero zip que incluyen los 3 primeros puntos.

1.-Fichero de modelado de base de datos mwb de MySQL: Define la base datos que utilizaremos.

2.-Carpeta del proyecto JAVA: Contiene las clases y librerías que se utilizan así como el código de los jsp que finalmente se muestran en la web.

3.-Fichero de despliegue final WAR: Fichero de despliegue preparado para subir, desplegar y ser ejecutada por el servidor. Está optimizado para TOMCAT v7.

4.-URL y usuarios para pruebas:

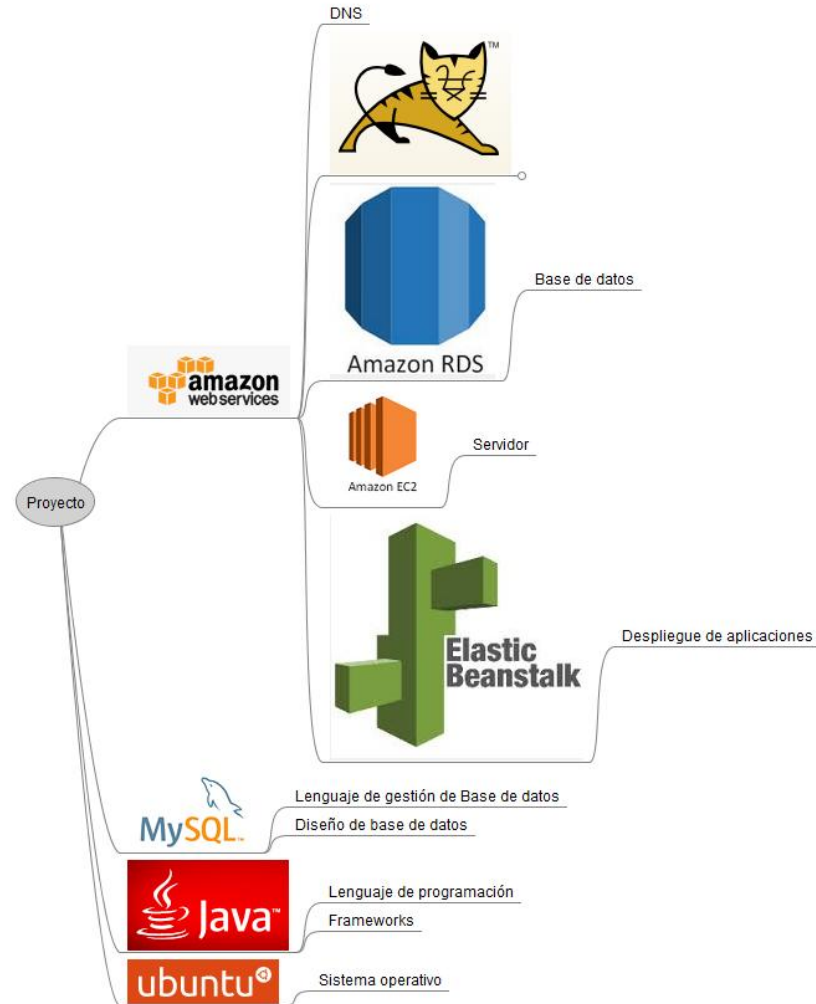
<http://rmweb-env.elasticbeanstalk.com/>

Username	ROL	Password
tecnico	it	123uoc
facility	facility	123uoc
seleccion	rrhh	123uoc
control	quality	123uoc
administrador	admin	123uoc

El uso posterior y la explotación de la misma se harían en una URL distinta por cada cliente que contrate el servicio.

3.2 ÍNDICE DE CONTENIDOS E ÍNDICE DE FIGURAS.

El proyecto queda definido con el siguiente esquema que viene a resumir el conjunto de tecnologías que se han utilizado.



4 DEDICATORIA Y AGRADECIMIENTOS

4.1 DEDICATORIA

Dedico este trabajo a mi mujer Laura,

a su paciencia por dejarla apartada durante días por terminar las entregas,

por sus ánimos cuando mejor te vienen

a sus palabras de ¿qué tal vas?

a sus expresiones en la cara cuando le cuento que ya la tengo subida en la “nube” pero no sé si el despliegue va a funcionar en la instancia que he contratado en Amazon.

Por ella, por su amor y su empatía. Te quiero.

4.2 AGRADECIMIENTOS

Quiero agradecer a distintos compañeros de trabajo que me han animado a seguir. Por invitarme a cafés y poder quitarme el sueño mientras les comento cómo va mi programa. Pablo Sousa y Fernando Castro. A mi jefe por dejarme escaparme un par de días.

También a los blogueros y web masters que comparten sus tutoriales, a Wikipedia, pero sobre todo a Google, por ayudarme a encontrar términos, fallos en segundos, etc. También a todas las comunidades de desarrolladores.

No me quiero olvidar del maestro que me ha animado a terminar este proyecto. Gracias a todos.