

Universidad
Politécnica
de Madrid

ETSI **SISTEMAS
INFORMÁTICOS**

**Escuela Técnica Superior de
Ingeniería de Sistemas Informáticos**

**APLICACIÓN WEB: SISTEMA DE GESTIÓN
DE INCIDENCIAS**

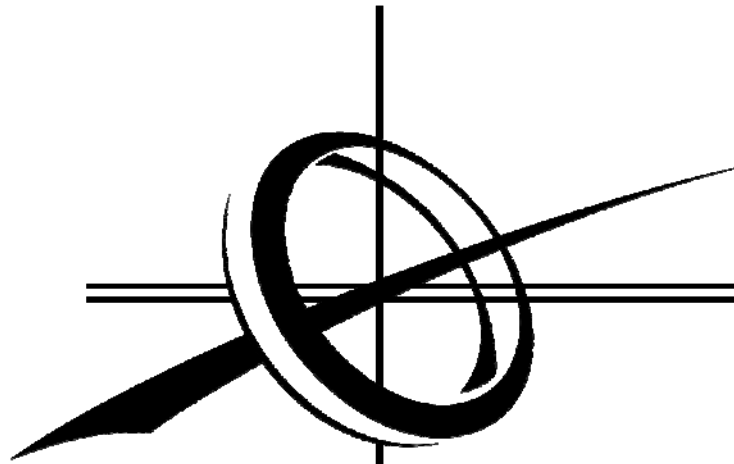
TRABAJO FIN DE CARRERA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso 2016/2017

Autor: Miguel Ambrós Mendioroz





Universidad
Politécnica
de Madrid

ETSI **SISTEMAS
INFORMÁTICOS**

**Escuela Técnica Superior de
Ingeniería de Sistemas Informáticos**

**APLICACIÓN WEB: SISTEMA DE GESTIÓN
DE INCIDENCIAS**

TRABAJO FIN DE CARRERA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso 2016/2017

Autor: Miguel Ambrós Mendioroz

Director: José Eugenio Naranjo



AGRADECIMIENTOS:

A Santi, Yul y Eo, por motivarme para hacer (de una vez) el proyecto

A Almu por todo tu apoyo y ayuda, sin ti no habría podido



TABLA DE CONTENIDOS

TABLA DE CONTENIDOS	5
TABLA DE ILUSTRACIONES	7
ABSTRACT	9
INTRODUCCIÓN.....	11
ANTECEDENTES.....	11
OBJETIVOS	12
ESTADO DEL ARTE.....	15
DEFINICIÓN DE INCIDENCIA.....	15
ORIGEN DE LAS INCIDENCIAS	15
GESTIÓN DE INCIDENCIAS	16
REGISTRO DE INCIDENCIAS.....	17
CICLO DE VIDA DE UNA INCIDENCIA.....	18
ARQUITECTURA, METODOLOGÍA Y DESARROLLO	24
ARQUITECTURA	24
HARDWARE Y SISTEMA OPERATIVO	25
SERVIDOR WEB Y BASE DE DATOS.....	25
ARQUITECTURA LAMP	26
DESARROLLO.....	29
METODOLOGÍA	29
HERRAMIENTAS.....	31
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	34
REQUISITOS	34
LISTA DE REQUISITOS	34
PLANIFICACIÓN.....	40
ANÁLISIS Y DISEÑO	42
ANÁLISIS Y DISEÑO DEL SPRINT 1	42
ANÁLISIS Y DISEÑO DEL SPRINT 2	54
ANÁLISIS Y DISEÑO DEL SPRINT 3	57
IMPLEMENTACIÓN	62
IMPLEMENTACIÓN DEL SPRINT 1	62



IMPLEMENTACIÓN DEL SPRINT 2	67
IMPLEMENTACIÓN DEL SPRINT 3	68
PRUEBAS Y CERTIFICACIÓN	70
MANUALES Y DOCUMENTACIÓN	71
MANUAL DE USUARIO	71
CONCLUSIONES	85
REFERENCIAS	88



TABLA DE ILUSTRACIONES

Ilustración 1 – CICLO DE VIDA DE UNA INCIDENCIA.....	19
Ilustración 2 - DIAGRAMA DE ARQUITECTURA WEB	24
Ilustración 3 – ARQUITECTURA LAMP	26
Ilustración 4 – ESTRUCTURA DE PAGINA WEB.....	63
Ilustración 5 – FLUJO VALIDACIÓN USUARIO	65





ABSTRACT

This work shows analysis, design and development of a Web Application that works as a Defect Tracking System.

This Application provides a clear centralized overview of defects detected in different software systems. Lets user to create defect records and manage them. It also provides tools to generate reports of the recorded defects and their state.

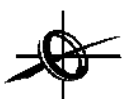
This Application has been developed with HTML5/CSS/PHP/JS over LAMP architecture.

* * * *

Este proyecto de fin de carrera detalla el proceso de análisis, diseño e implementación de una aplicación web dedicada a la Gestión de Incidencias.

Esta aplicación permite realizar una gestión centralizada de las incidencias detectadas en una serie de sistemas de software. Permite a los usuarios registrar incidencias y gestionarlos. También proporciona herramientas para generar informes sobre las distintas incidencias registradas y su estado.

Esta aplicación web está desarrollada en HTML5/CSS/PHP/JS utilizando la arquitectura LAMP.





INTRODUCCIÓN

En cualquier empresa de prácticamente cualquier tamaño que se dedique al desarrollo de software, o tenga un departamento dedicado a ello, resulta fundamental tener la capacidad de gestionar los defectos que se vaya registrando en el software que se desarrolla, o de llevar un control de aquellas mejoras que se vaya a hacer sobre él.

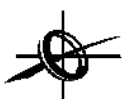
Desde el primer momento que se comienza con el desarrollo de un proyecto software ya se hace necesario disponer de una herramienta que permita hacer la gestión de las incidencias que vayan surgiendo. A mayor sea la complejidad del software desarrollado más importante se hace realizar correctamente esta gestión. En proyectos de software de una cierta entidad lo habitual es que se disponga de un departamento de pruebas y control de calidad dedicado entre otras cosas a la detección de defectos dentro del software desarrollado. Una mala gestión de las incidencias detectadas puede impactar directamente en la calidad del software desarrollado, multiplicando los costes.

ANTECEDENTES

Una compañía dedicada al comercio minorista posee un departamento propio dedicado al desarrollo de las aplicaciones de punto de venta que son utilizadas en sus tiendas. Este departamento se encarga del desarrollo de aplicaciones para terminales de punto de venta, aplicaciones para gestión de tienda, y todo el software para utilizar los distintos elementos desplegados en sus tiendas (pistolas de inventario, comprobadores de precios, equipos de fila única).

El control de las incidencias que se detectan en las aplicaciones que están en desarrollo se hace de forma rudimentaria, utilizando hojas de cálculo para el registro de incidencias detectadas. El seguimiento de las incidencias con este sistema resulta complejo, lo cual se traduce en un mayor esfuerzo para realizar la gestión de las incidencias.

Para realizar un correcto control de las incidencias que van surgiendo en las distintas aplicaciones que se desarrolla dentro del departamento, cada vez más numerosas y complejas, se observa la necesidad de disponer de una herramienta que facilite esta tarea.



Por esto se decide encargar el desarrollo de una aplicación dedicada al control de incidencias.

Se opta por desarrollar una aplicación web debido a las ventajas que esto ofrece frente a las aplicaciones de escritorio:

- Para realizar el acceso a una aplicación web basta con disponer de una navegador web y conexión al servidor web. Esto permite ahorrar costes, ya que no requieren de grandes recursos, hoy en día es posible realizar el acceso mediante un teléfono móvil, tableta, portátil o PC de escritorio de gama baja.
- Las aplicaciones web ofrecen un interfaz sencillo e intuitivo, no requieren de conocimientos previos de informática.
- Con una aplicación web resulta muy fácil centralizar todas las áreas de trabajo. Se ofrece la posibilidad de realizar acceso concurrente por varios usuarios. Los datos se almacenan en un servidor único
- Se dispone de una total inmediatez de acceso, ya que no es necesario realizar una instalación previa de la aplicación.
- Existe gran cantidad de soluciones de software libre gratuito que permiten montar una aplicación web.

Después de realizar reuniones con los jefes de desarrollo de las distintas áreas, y con el encargado del equipo de pruebas, se perfila una serie de requisitos que debe cumplir esta herramienta.

OBJETIVOS

El objetivo principal como se ha comentado anteriormente es desarrollar una aplicación web para realizar la gestión de incidencias. Este desarrollo debe realizarse de la forma más rápida posible.

Esta aplicación debe permitir realizar un seguimiento sencillo de las incidencias detectadas, de forma acorde con las necesidades de los departamentos de desarrollo y



Aplicación Web: Sistema de Gestión de Incidencias

pruebas. Se define inicialmente la necesidad de establecer un acceso controlado a la aplicación mediante la definición de usuarios. Dentro de la aplicación se debe de ofrecer la opción de crear las incidencias detectadas, realizar listados de estas incidencias aplicando criterios de filtrado, y obtener informes resumidos y detallados de las mismas.

Debido a la urgencia con la que se quiere disponer de la aplicación web, no se desea realizar un desarrollo demasiado complejo ni a largo plazo, ni invertir demasiados recursos. La implementación de esta aplicación será realizada por el propio equipo de desarrollo de la empresa.





ESTADO DEL ARTE

En este apartado hablaremos de qué se puede considerar una incidencia y donde suelen tener su origen. También se comentará la necesidad de realizar el seguimiento de las incidencias a lo largo del ciclo de vida del proyecto, y de cómo hacer una correcta gestión de las mismas.

DEFINICIÓN DE INCIDENCIA

El desarrollo de software puede ser una tarea compleja y muy exigente, y requiere una correcta organización de cara a conseguir unos resultados satisfactorios. Los equipos de desarrollo habitualmente siguen algún tipo de metodología para conseguir sus metas, ya que sin ella les resultaría mucho más complicado.

Un desarrollo de software perfecto es imposible de conseguir. En algún momento del ciclo de desarrollo es normal que surjan problemas. Se considera una incidencia a un problema determinado que se produce por un fallo de la aplicación. En el momento que la aplicación no cumple los requerimientos, o no cumple las expectativas del cliente, también se considera que se produce una incidencia: desde un error de ortografía en una página, hasta una total indisponibilidad de una aplicación.

ORIGEN DE LAS INCIDENCIAS

Las incidencias no tienen por qué tener su origen en un desarrollo incorrecto de una aplicación, aunque es lo más habitual. Se producen a lo largo de todo del ciclo de vida del producto:

- Fase de toma de requisitos

Cualquier malentendido entre el cliente y el equipo encargado de la toma de requisitos puede provocar problemas en la aplicación, que a la larga sean considerados incidencias. La aplicación puede cumplir perfectamente los requisitos fijados, pero aun así ser considerada incorrecta por el cliente o usuario final.



- Diseño de arquitectura

La elección de una arquitectura inapropiada también puede ser una fuente de incidencias. Puede ocurrir que la arquitectura elegida no permita cumplir alguno de los requisitos, debido por ejemplo a limitaciones técnicas. También pueden surgir problemas entre los interfaces de los distintos módulos que forman la arquitectura, que terminen afectando a la aplicación, por ejemplo errores de integración entre el sistema operativo y el hardware o la base de datos.

- Fase de desarrollo

El desarrollo de software es un proceso muy complejo: hay que codificar los requerimientos, diseñar los algoritmos, y todo esto es realizado por personas.

- Fase de pruebas

En esta fase es habitualmente donde se detecta la mayoría de las incidencias. Aquellas incidencias que no sean detectadas en esta fase impactarán directamente sobre la calidad del software probado.

- Implantación y mantenimiento

Aunque un desarrollo haya superado todas las fases correctamente una incorrecta implantación, o un incorrecto mantenimiento, pueden causar muchos problemas. Por ejemplo si una aplicación no se configura correctamente puede producir muchos problemas, aunque el desarrollo sea perfecto.

GESTIÓN DE INCIDENCIAS

La gestión de incidencias surge de la necesidad de controlar todas las incidencias que puedan ocurrir a lo largo de todo el ciclo de vida del software desarrollado. Se trata de tener los mecanismos adecuados para registrar las incidencias con toda la información necesaria, y poder conocer su estado y evolución. También se hace necesario tener las



herramientas para poder conocer las incidencias registradas aplicando una serie de filtros, y poder conocer el detalle de cada una de ellas.

REGISTRO DE INCIDENCIAS

El control de incidencias comienza por el registro de incidencias. Ya que prácticamente desde el comienzo del desarrollo de una aplicación ya es posible que aparezcan incidencias, es necesario plantearse desde el primer momento de qué forma se van a registrar las incidencias.

Cada desarrollo tiene sus propias características y condiciones, y estas pueden influir en la forma en la que se debe registrar con la incidencia, sobre todo lo relativo a que información se debe registrar. Aun así es posible establecer una serie atributos concretos de la incidencia que conviene añadir a la hora de hacer el registro:

- Identificador

Cada incidencia debe tener un identificador único con el cual referirse a ella. Este identificador puede ser asignado directamente por la persona que registra la incidencia, o puede ser la propia herramienta que se utilice para la gestión de incidencias la que se encargue de ello. Esto último es lo habitual.

- Descripción (breve y detallada)

De cara a identificar la incidencia de forma sencilla habitualmente se añade una breve descripción de la incidencia, suficientemente corta como para que pueda figurar en listados, pero suficientemente larga para que sea descriptiva. Esto no quita que luego sea necesario almacenar también una descripción detallada de la incidencia, con el máximo de información, que pueda ser utilizada para un posterior análisis de la incidencia.

- Proyecto



Habitualmente se utiliza la misma herramienta de gestión de incidencias para controlar incidencias de varios proyectos o desarrollos distintos. En ese caso es necesario distinguir en que proyecto se debe registrar la incidencia.

- Estado

El estado definirá la situación en la que se encuentra una incidencia. Las incidencias pueden pasar por varios estados distintos: puede estar recién creada, en fase de análisis, en fase de corrección, resuelta, cerrada, etc. Más adelante en este libro se detallará en profundidad el ciclo de vida de una incidencia.

- Severidad

La severidad de una incidencia indica en qué modo afecta al funcionamiento de la aplicación. Determinar la severidad de una incidencia es un proceso bastante subjetivo, puede ocurrir que lo que al equipo que realice las pruebas no le parezca un problema importante, al usuario final de la aplicación le parezca crítico.

- Registro

Resuelta conveniente añadir de alguna forma un registro a la incidencia que permita obtener información de los cambios que se han ido haciendo sobre ella. Como explicaremos más adelante las incidencias siguen un ciclo de vida, a lo largo del cual van cambiando de estado, y resulta muy útil conocer estos cambios.

Como se ha comentado, aparte de estos atributos citados, que se podrían considerar los necesarios a la hora de registrar una incidencia, se podría añadir otros. Esto viene condicionado por características concretas de los proyectos cuyas incidencias se quiere gestionar.

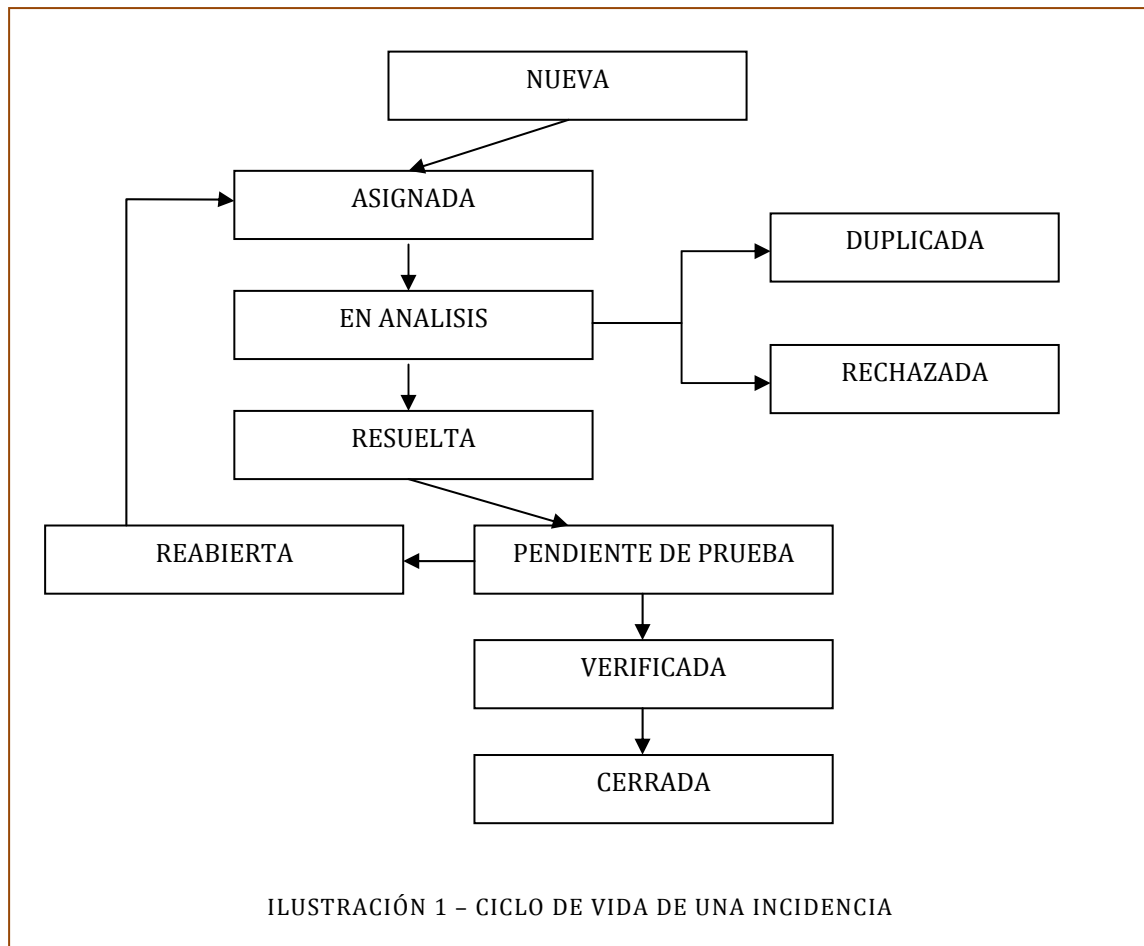
CICLO DE VIDA DE UNA INCIDENCIA

Como se ha comentado las incidencias tienen un estado asociado, debido a que están sujetas a cambios. Este estado indica en qué situación está una incidencia en un momento determinado. Desde que una incidencia se registra hasta que se da por resuelta o se



descarta, esta va atravesando una serie de estados, lo que se denomina el ciclo de vida de la incidencia.

Esta figura detalla cual se podría considerar el ciclo de vida de una incidencia, atendiendo a los cambios que va sufriendo en su estado:



Esta sería la descripción de cada estado:

- Nueva

Es el estado inicial de la incidencia, cuando se acaba de crear. Se entiende que esta incidencia está registrada, pero no se ha realizado ninguna acción sobre ella todavía. Las incidencias suelen ser registradas por el equipo de pruebas, aunque pueden ser registradas por otros equipos. Por ejemplo el propio equipo de desarrollo puede



registrar incidencias encontradas en sus pruebas unitarias, si lo considera conveniente.

- Asignada

Este estado se utiliza para tener un control de a qué persona o equipo se ha asignado la incidencia. De esta forma es posible saber quien se encargará de su análisis, y probablemente también de la resolución.

En proyectos de cierta entidad y nivel de organización puede existir una persona o equipo de personas que se encargan de hacer un análisis preliminar de las incidencias es este estado, y deciden a quien son asignadas. En proyectos pequeños a veces es la propia persona que registra la incidencia la que decide a quien se asigna. Las incidencias normalmente se asignan a la persona o equipo que se va a encargar de la resolución.

- En análisis

La persona o equipo que tienen asignada la incidencia la están analizando. Este análisis puede acabar de varias formas distintas:

- La incidencia puede ser aceptada como tal, y entonces se iniciará el trabajo para resolverla.
- La incidencia no se considera válida, bien por considerar que ya se ha registrado previamente, en cuyo caso estaría duplicada, o bien porque no se considera que se trate de una incidencia, en cuyo caso será rechazada.

- Duplicada

Después de la fase de análisis se ha visto que la incidencia es un duplicado de otra registrada previamente

- Rechazada

Esto significa que la persona o equipo encargados del análisis han concluido que no se trata de una incidencia real. Esto puede ocurrir si lo que es el funcionamiento normal del software se considera incorrecto por la persona que ha registrado la incidencia.



- Resuelta

Las incidencias en este estado se considera que ya han analizadas y se ha encontrado la forma de resolverlas. La resolución puede implicar cambios en el código desarrollado, pero no siempre es así. También puede ocurrir que la corrección se base en un cambio de configuración, ya sea en la propia aplicación o de alguno de los elementos que forman la arquitectura de la aplicación.

- Pendiente de prueba

Cuando una incidencia está en este estado significa que se ha hecho una acción correctiva para resolverla, pero es necesaria una verificación por parte de la persona o equipos de pruebas. El paso a este estado también suele indicar que la incidencia se ha incluido en una nueva versión o paquete correctivo.

Si el resultado de las pruebas es satisfactorio se puede considerar que se ha verificado la resolución de la herramienta, en cuyo caso su estado se debería cambiar a Verificada. Si por el contrario las pruebas indican que la incidencia no está resuelta se debe cambiar el estado a Reabierta.

- Reabierta

Una incidencia se encuentra en este estado su cuando se ha ido a realizar la verificación de la resolución de la incidencia se ha comprobado que no está resuelta. Esto significa que la incidencia debe comenzar de nuevo todo el ciclo.

- Verificada

La persona o equipos de pruebas han comprobado que la incidencia ya no tiene lugar, dando por válida la corrección.

- Cerrada

Es el estado final de la incidencia, una vez completa su ciclo de vida.



Idealmente, a lo largo del ciclo de vida de la incidencia se debe ir añadiendo información a medida que esta cambia de estado: quien se encargó del análisis y que se determinó en él, quien se encargó de la resolución y como lo hizo, en que versión o paquete se incluyó el correctivo, quien se encargó de la verificación, etc. Toda esta información puede resultar muy útil. Por ejemplo ante una incidencia nueva similar a otra ya cerrada se puede consultar que correctivo se hizo. También por ejemplo resulta práctico para saber que versión concreta de un producto contiene el correctivo de una incidencia determinada.





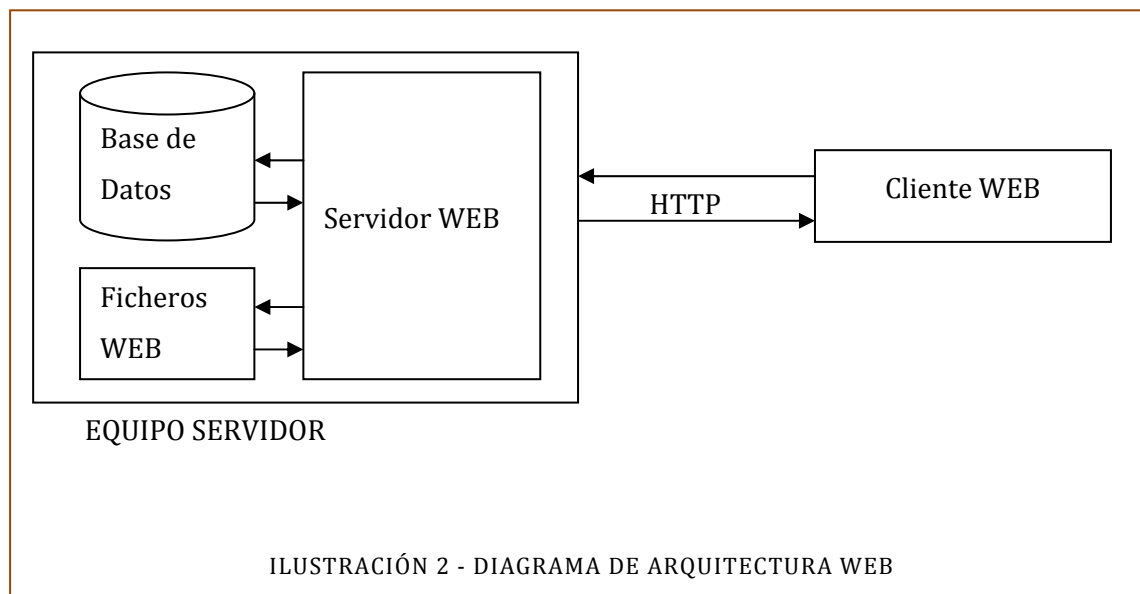
ARQUITECTURA, METODOLOGÍA Y DESARROLLO

En este capítulo se detalla la arquitectura que se ha decidido utilizar en la aplicación, y las razones que han motivado esta decisión.

También se detalla la metodología elegida para realizar el desarrollo software de la aplicación.

ARQUITECTURA

La arquitectura básica de una aplicación web se compone habitualmente de un equipo, ya sea físico o virtual, en el que corre un servidor web. Este servidor web opcionalmente puede utilizar una base de datos, que puede encontrarse en el mismo equipo o en otro equipo distinto. El equipo está conectado a una red a través de la cual se recibe las peticiones vía HTTP, y el servidor web las recibe y contesta.



Existen muchas opciones a la hora de escoger la arquitectura de un servidor web. Tanto a nivel de hardware como de software. En este caso la elección viene condicionada en parte

por el hardware disponible, y por los conocimientos técnicos de los equipos de arquitectura y desarrollo.

HARDWARE Y SISTEMA OPERATIVO

En el CPD de la empresa se encuentran los servidores que utiliza el departamento de desarrollo, por ejemplo los servidores que alojan el software utilizado para realizar el control de versiones de los distintos desarrollos, con sus distintos repositorios. Estos servidores son mainframes de HP o IBM, y utilizan el sistema operativo Linux. En el CPD también se dispone de una serie de entornos no productivos que replican las distintas configuraciones que se pueden encontrar en los centros de producción, que no son otros que las tiendas donde se realiza la venta. Por un lado están los servidores centrales de las tiendas, y por otro lado los terminales de punto de venta. Para ambos tipos de máquinas el sistema operativo elegido también es Linux.

De esta forma lo más sencillo de cara a montar la aplicación web es utilizar el hardware y sistema operativo ya existentes. Así no se hace necesario ningún gasto extra, y además como el departamento de arquitectura ya tiene los conocimientos necesarios de administración no es necesario dedicar tiempo a la formación. Otra ventaja es que al contar con experiencia en este tipo de entornos hay menos posibilidad de cometer errores, y se reduce el tiempo dedicado a la preparación de la arquitectura.

SERVIDOR WEB Y BASE DE DATOS

Como se ha comentado los centros de producción (tiendas), existe una aplicación web que se utiliza para la gestión de la tienda: operativas de apertura y cierre de tienda, control de tickets, facturas, informes de venta, etc. Esta aplicación corre en un servidor web apache, y accede a una base de datos MySQL. Tanto la base de datos como el servidor web se ejecutan dentro de un servidor con sistema operativo Linux. Esto es lo que se conoce como arquitectura LAMP.

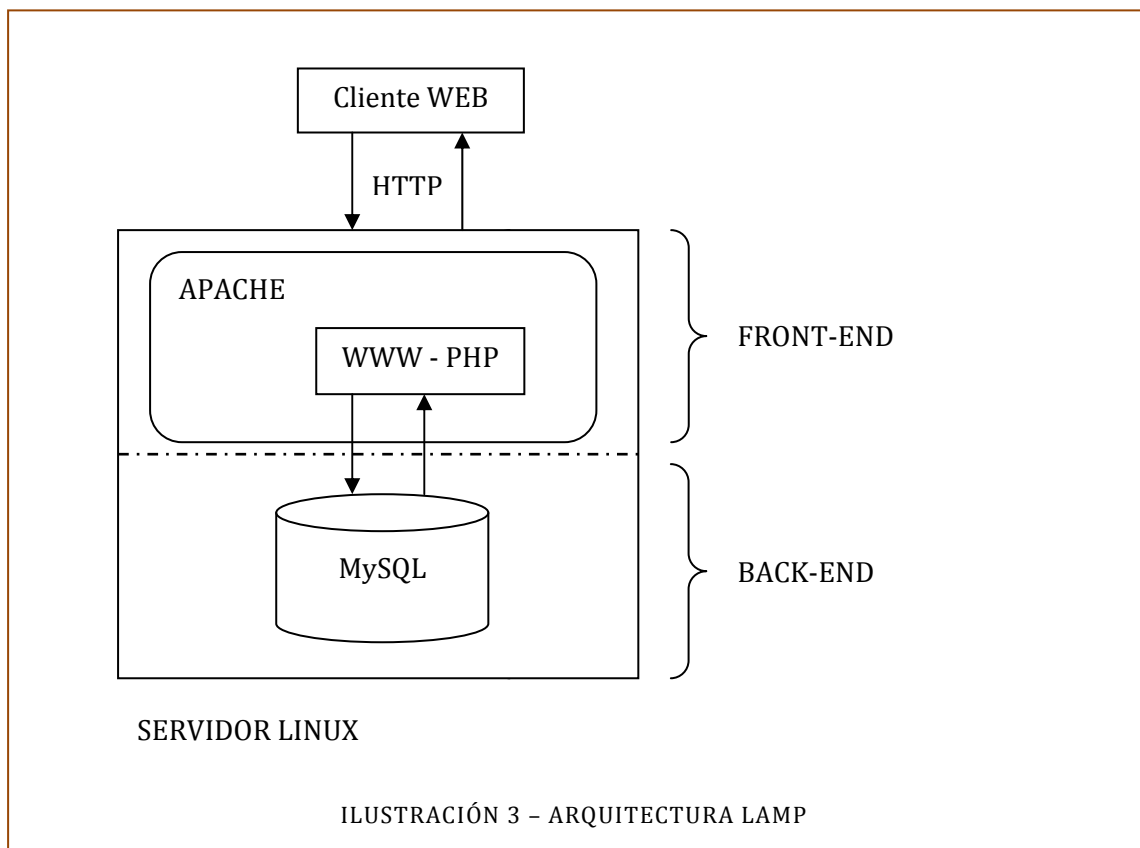
Debido a que se desea realizar el desarrollo de la aplicación de la forma más rápida, resulta lógico elegir una arquitectura con la cual ya están familiarizados los departamentos de arquitectura y desarrollo. Esta arquitectura, como se verá en el siguiente capítulo, se puede considerar adecuada para el desarrollo de la aplicación web.



ARQUITECTURA LAMP

La arquitectura LAMP (Linux como sistema operativo, Apache como servidor Web, MySQL como base de datos, y PHP como lenguaje de programación Web) es una plataforma de desarrollo web. Al tratarse de una arquitectura basada en capas tambien se suele usar el término pila LAMP para referirse a ella.

En este diagrama se puede ver como se integran los dintintos componentes, o capas, de la arquitectura LAMP:



Los componentes de la arquitectura LAMP no fueron diseñados inicialmente para trabajar juntos, pero se ha demostrado que se integran perfectamente unos con otros. Además comparten la característica de ser todos software libre de código abierto. Esta arquitectura se ha hecho muy popular como plataforma de desarrollo web, llegando a convertirse en un estándar de facto.

Esta es la descripción detallada de los distintos componentes:

LINUX

Linux es el sistema operativo donde se ejecuta el servidor web, y por lo tanto la aplicación web. Linux ofrece mucho más que una plataforma para aplicaciones web, pero en este ámbito concreto ofrece muchos beneficios: es un sistema operativo muy fiable, flexible y seguro.

El enfoque de Linux como aplicación libre de código abierto ha hecho que este sistema operativo esté disponible en un gran número de plataformas, desde PCs de escritorio a servidores mainframe. Al ser muy flexible y configurable ofrece la posibilidad de ser adaptado y optimizado para funcionar como plataforma web.

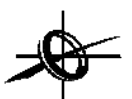
Apache HTTP

Apache es un servidor Web de código abierto, desarrollado y mantenido por la Apache Software Foundation. Resulta un producto relativamente simple, pero muy potente y capaz. Como uno de sus fuertes destaca la posibilidad de añadir módulos que amplíen sus capacidades básicas. Con el paso del tiempo se ha convertido en el servidor Web más utilizado.

Apache está perfectamente integrado con Linux, de tal forma que la mayoría de las distribuciones Linux incluyen Apache en su configuración básica.

MySQL

MySQL es un sistema de base de datos relacional. Se trata de un software sencillo de instalar y configurar, sin necesidad de grandes conocimientos. Como base de datos resulta una solución muy potente, ofreciendo la posibilidad de modificar la configuración para obtener el máximo rendimiento. Está considerada como la base de datos de código abierto más popular, sobre todo en entornos de desarrollo Web.



PHP

PHP es un lenguaje de programación creado para el desarrollo de contenido web dinámico. Corre en el lado de servidor, pudiendo incorporarse directamente en el documento HTML.



DESARROLLO

En este apartado se comenta la metodología utilizada para realizar el desarrollo de la aplicación web. También se habla de las distintas herramientas de desarrollo utilizadas.

METODOLOGÍA

Como se ha visto el objetivo final es desarrollar una aplicación web para realizar la gestión de incidencias. Esta aplicación web no resulta en sí misma un desarrollo extremadamente complejo, y además se va a utilizar una arquitectura con la que los equipos de arquitectura y desarrollo están familiarizados. Además hay que tener en cuenta que se quiere realizar el desarrollo de la forma más rápida posible. Respecto a los requisitos se dispone de unos requisitos principales, pero que no están totalmente detallados, y que pueden estar sujetos a cambios.

Teniendo en cuenta todo lo anterior se ha considerado que se debe utilizar una metodología ágil, ya que se ajusta más a las características del proyecto:

- Con metodologías ágiles es posible comenzar a disponer de entregas tempranas y continuas de valor, en cortos periodos de tiempo. El desarrollo del software se hace de forma incremental y con iteraciones muy cortas.
- El hecho de que los requisitos no estén plenamente definidos y sean cambiantes es algo aceptable.
- Las metodologías ágiles se centran en el factor humano. Le dan mayor valor al individuo, y a la colaboración del cliente. Se basan en las personas y no en los procesos.
- Resultan de muy fácil aplicación en proyectos pequeños.

De las distintas metodologías ágiles disponibles la elegida es Scrum.

SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible



de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Teniendo en cuenta que existe la necesidad de disponer de este desarrollo cuanto antes, y además que al tratarse de un desarrollo interno la participación activa del usuario final es posible, esta metodología resulta especialmente conveniente.

ITERACIONES – SPRINTS

En Scrum los proyectos se van desarrollando en iteraciones, que son bloques de tiempo habitualmente cortos, que pueden ir desde 5 días a unas semanas, habitualmente se establece que no más de 4 semanas. A estas iteraciones se las conoce como sprints. La idea es que en cada iteración se produce un resultado concreto, que sería una versión ampliada del producto final, que es susceptible de ser entregada y utilizada. Después de un análisis de los requisitos iniciales, y teniendo en cuenta la cantidad de personas que van a trabajar en el desarrollo y sus conocimientos, es posible estimar la duración adecuada de los sprints, de cara a que al final de cada sprint se haya podido producir una versión incremental de producto.

Inicialmente se define cuantos sprints se va a realizar, la duración de cada sprint, y que partes de los requisitos entrarán en cada sprint. Lógicamente se prioriza los requisitos de tal forma que en el primer Sprint se incluya los requisitos más básicos, aquellos que no dependan de otros requisitos y cuya implementación sea fundamental para obtener una versión funcional del desarrollo.

Estos son los pasos que se realiza para cada iteración o Sprint:

1. **Planificación:** Al comienzo de la iteración se realiza una reunión para hacer la planificación de la iteración. Esta tiene dos partes:
 - Selección de requisitos: a partir de una lista priorizada de requisitos, se discute que requisitos formarán parte de la iteración.
 - Planificación: a partir de los requisitos seleccionados se crea una lista de tareas necesarias.
2. **Ejecución de la iteración:** El equipo comienza la ejecución de las tareas. Cada día se realiza una breve reunión en la que cada persona detalla brevemente sus



avances desde la reunión anterior, que se plantea hacer, y las posibles dificultades que pueda encontrar.

3. Inspección: Al final de la iteración se hace una reunión que consta de dos partes:

- Demostración: se presenta al cliente el resultado final de la iteración.
- Retrospectiva: se analiza el proceso de la iteración, identificando aquellos impedimentos que hayan podido surgir, como se han resuelto, y si se pueden dar en el futuro.

SCRUM MANAGER

En la metodología Scrum existe la figura del Scrum Manager, que es una persona que se encarga de comprobar que aplica correctamente la metodología Scrum. Esta persona participa en las distintas reuniones que se realicen, realizando a veces de mediador entre el equipo de desarrollo y el usuario. También se encarga facilitar lo máximo posible el trabajo del equipo de desarrollo.

HERRAMIENTAS

Estas son las herramientas que se han utilizado durante el desarrollo de la aplicación:

Desarrollo:

- Eclipse Helios for PHP. Entorno de desarrollo integrado para el desarrollo web con PHP. Se ha utilizado para realizar el desarrollo del software PHP (codificación y depuración de errores).
- Notepad++ 7.3. Editor de texto.

Ejecución y pruebas:

- Wamp Server 2.4/2.5. Entorno WAMP (Windows/Apache/MySQL/PHP). Es el equivalente a la arquitectura LAMP en Windows. Se ha utilizado para poder disponer de un entorno de pruebas y depuración en aquellos equipos con Windows.
- Clientes web. Se ha hecho uso de los clientes web más habituales para garantizar la completa compatibilidad con todos ellos:
Firefox 52.0.2.



Chrome 55.0

Internet Explorer 8-10

Documentación:

- Microsoft Office 2011. Se ha utilizado para escribir los distintos documentos que se ha utilizado en el proyecto: documentación (requisitos, especificaciones) y manuales,

Producción:

- Filezilla 3.15.0.2. Cliente de FTP. Utilizado para realizar transmitir por FTP los paquetes desarrollados a los servidores de producción.
- Putty 0.59. Cliente de SSH. Utilizado para acceder vía SSH a los servidores de producción.





ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

REQUISITOS

Como se ha comentado el objetivo fijado es el desarrollo de una aplicación web para realizar la gestión centralizada de las incidencias de distintos proyectos de software.

Después de realizar reuniones con los jefes de desarrollo de las distintas áreas, y con el encargado del equipo de pruebas, se perfila una serie de requisitos que debe cumplir esta herramienta.

LISTA DE REQUISITOS

1. Control de acceso y perfiles de usuario

La aplicación deberá disponer de un control de acceso, de tal forma que sólo los usuarios registrados podrán acceder a la aplicación, después de realizarse una validación previa de sus credenciales. Esto implica que la aplicación deberá almacenar la información relativa a los usuarios, y así poder realizar la validación. Inicialmente los usuarios podrán tener asociados tres perfiles distintos:

- Administrador.
- Operario.
- Invitado.

A partir de ahora, haremos referencia a estos perfiles a la hora de describir el resto de pantallas/funcionalidades de la aplicación Web.

1.1. Validación de acceso de usuario a través de LDAP

Para facilitar al acceso a los usuarios, además de poder realizar una validación de acceso local, se ofrecerá la posibilidad de realizar el control de acceso contra un LDAP. Esto permitirá a los usuarios acceder con el mismo usuario y contraseña que utilizan en otras aplicaciones dentro de su área de trabajo.



2. Gestión de Incidencias

Una vez se haya hecho la validación del usuario, y se haya permitido el acceso a la aplicación web, se debe mostrar los menús que permitirán realizar el control de las incidencias registradas. Estos menús dependerán del perfil del usuario, habilitándose o deshabilitándose las opciones según corresponda con el perfil del usuario validado.

2.1. Consulta de Incidencias

Con este menú se accederá a una página en la que será posible realizar una consulta de aquellas incidencias que estén registradas en la aplicación. Este menú estará disponible para cualquier perfil

La página estará formada por tres bloques:

a) Filtros:

Se permitirá definir una serie de filtros para acceder a las incidencias registradas. Los filtros que se podrá aplicar serán los siguientes:

Filtro Identificador: Todas las incidencias creadas dispondrán de un identificador único. Esta opción permitirá filtrar por ese identificador.

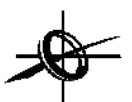
Filtro Proyecto: Ofrece filtrar por el entorno al cual está asociada la incidencia.

Filtro Estado: Será posible filtrar por el estado de incidencia. De forma inicial se definirá una serie de estados por los cuales podrán pasar las incidencias, acordes con la situación en la que se encuentren.

Filtro Criticidad: Este filtro permitirá seleccionar las incidencias que tengan una criticidad determinada.

Filtro Aplicación: Con este filtro se podrá seleccionar aquellas incidencias que pertenezcan a una aplicación determinada.

Filtro Descripción: Este filtro se utilizará para filtrar aquellas incidencias que contengan un texto determinado en su descripción.



Filtro Fecha: Se permitirá filtrar las incidencias creadas en una fecha o rango de fechas determinado.

b) Barra de opciones

Contendrá las posibles opciones a realizar, que serán:

1. Consulta

Con esta opción se aplicarán los filtros indicados, para mostrar las incidencias que los cumplan.

2. Exportar a Excel

Con esta opción será posible exportar a una hoja Excel todas aquellas incidencias que se esté visualizando.

3. Exportar a Excel (detalle)

Con esta opción será posible exportar a una hoja Excel todas aquellas incidencias que se esté visualizando.

c) Información

En este bloque se permitiría visualizar aquellas incidencias que cumplan todos los criterios de filtrado. Se mostrará una tabla que contendrá una serie de campos considerados principales:

Identificador de incidencia

Proyecto

Criticidad

Estado

Descripción

Fecha de Denuncia

En cualquier caso será posible seleccionar una incidencia concreta para obtener el detalle.

2.1.1. Volcado de incidencias a hoja Excel

Será posible volcar la información sobre las incidencias que se esté mostrando en la página de consulta de incidencias. Será posible obtener una hoja Excel con la información reducida y con el detalle de cada incidencia.



2.2. Creación de Incidencias

Este menú mostrará una página que permitirá realizar el registro de una incidencia. Estos son los campos de los que se compondrá una incidencia:

Obligatorios:

- Identificador:

Las incidencias tendrán un identificador único. Este identificador no podrá ser introducido por el usuario, sino que se asignará automáticamente, pero se indicará en la página para que el usuario pueda conocerlo.

- Proyecto:

Las incidencias creadas llevarán un proyecto asociado, que corresponderá a un área concreta. De forma inicial se definirá una serie de proyectos, correspondientes a las áreas ya existentes:

Proyecto	Descripción
TPV	Incidencias relacionadas con las aplicaciones de los terminales de punto de venta
Servidor	Incidencias relacionadas con las aplicaciones del servidor central de tienda
Aplicación WEB	Incidencias relacionadas con la aplicación web de gestión de tienda
Comprobadores de precios	Incidencias relacionadas con los comprobadores de precios

- Área resolutoria:

Este será el equipo que a priori deba encargarse de la resolución de la incidencia. Se definirá una serie de posibles valores acordes con los equipos existentes. La aplicación ofrecerá la posibilidad de modificarlos, añadir equipos nuevos, o borrar los ya existentes.

- Estado:

Este campo se corresponderá con el estado de la incidencia. Se define una serie de estados iniciales:

Estado	Descripción
Nueva	Incidencias recién creada
Asignada	La incidencia se encuentra asignada a un área resolutoria



En análisis	La incidencia se encuentra en análisis por el área resolutoria que la tiene asignada
Resuelta	La incidencia se ha resuelto
Pendiente de Prueba	La incidencia ha sido incluida en una versión y se debe comprobar si ya está resuelta
Verificada	La resolución de la incidencia ha sido verificada
Reabierto	La incidencia no se considera resuelta, y se ha vuelto a abrir
Duplicada	La incidencia se considera un duplicado de otra registrada previamente
Rechazada	La incidencia se considera incorrecta o inexistente y se rechaza
Cerrada	La incidencia se encuentra cerrada

Algunos estados llevarán asociada información extra. Por ejemplo al asignar una incidencia será posible indicar a quien se asigna. Cuando una incidencia está en análisis será posible introducir un texto con el análisis. También cuando una incidencia se resuelva será posible introducir un texto que indique como se ha hecho la resolución.

- **Criticidad:**

Se define una serie de valores que indican la gravedad de la incidencia: Baja, Media y Alta. Como en los casos anteriores será posible modificar los posibles valores para este campo.

- **Fecha de denuncia:**

Es la fecha en la cual se detecta esta incidencia.

- **Descripción corta:**

Breve descripción de la incidencia.

- **Descripción:**

Descripción completa de la incidencia.

- **Afectación:**

Detalle de que funciones concretas dentro del área o aplicación se ven afectadas por la incidencia.

- **Contingencia:**



Describirá una solución de contingencia, en caso de haberla.

- Bitácora:

Se trata de un campo que permitirá añadir información para controlar la evolución de la incidencia. Se compondrá de una fecha, y de un texto informativo.

3. Administración

Este menú sólo será visible para el perfil Administrador, ya que desde este menú se podrá modificar tanto la configuración de datos como los usuarios de la aplicación.

3.1. Gestión de Usuarios

Se crea una página para realizar la gestión de usuarios. En esta página es posible:

- Dar de alta un usuario.
- Modificar un usuario ya existente, cambiando su perfil, o su forma de acceso.
- Dar de baja un usuario.

3.2. Gestión de campos maestros

Existe una gran cantidad de elementos en la aplicación cuyos posibles valores pueden ser alterados, desde esta opción de menús se ofrece la posibilidad de hacer la modificación de estos campos definidos como maestros.

Las operaciones que se podrán realizar serán: listar los registros del campo maestros, modificarlos, borrarlos y crear registros nuevos. Los campos maestros que se podrán modificar son los siguientes:

Campo maestro	Descripción
Proyecto	Define el proyecto en el que se produce la incidencia.
Criticidad	Define como de crítica es una incidencia
Estado	Define los distintos equipos estados posibles para una incidencia.



PLANIFICACIÓN

Como se ha comentado ya, la aplicación web a desarrollar será para uso interno, y el desarrollo será realizado por los equipos ya existentes en la empresa. Debido a que existe cierta urgencia por disponer de la aplicación web se quiere que el desarrollo se realice lo más rápido posible. Al haberse elegido una arquitectura con la que los equipos de desarrollo y arquitectura están familiarizados, y de que se trata de una aplicación relativamente sencilla, el desarrollo lo realizará un equipo pequeño, inicialmente no más de tres personas del equipo de desarrollo, dedicadas en exclusividad. A este equipo se sumará una persona del equipo de arquitectura, que se encargará de montar los entornos de desarrollo y pruebas. Una vez se disponga de una versión inicial también participará una persona del equipo de pruebas y calidad que se encargará de ir probando la versión inicial, y los posteriores incrementos.

Como se ha comentado en la sección de metodología, se dispondrá de una persona más dedicada a las funciones de Scrum Manager, que se encargará de que se aplique correctamente la metodología Scrum, de coordinar y dirigir las distintas reuniones que se organicen, y en general de facilitar el trabajo del equipo.

Inicialmente se realizarán tres Sprints de 5 días hábiles cada uno. Los requisitos, ya enumerados en la sección de requisitos, se irán organizando de la siguiente manera:

SPRINT 1:

Equipo de desarrollo:

- Estructura básica de la aplicación Web: creación de las hojas CSS, modulo de acceso a base de datos.
- Requisito 1: Control de acceso y perfiles de usuario.
- Requisito 3.1: Gestión de usuarios.
- Requisito 3.2: Gestión de campos maestros.

Equipo de arquitectura:

- Instalación de servidores Linux, configuración de Apache y MySQL.
- Creación de base de datos.
- Preparación de paquete de instalación SPRINT 1.

Equipo de pruebas:

- Pruebas de la versión SPRINT 1.



SPRINT 2:

Equipo de desarrollo:

- Requisito 2.2: Creación de incidencias.
- Resolución de problemas detectados en versión SPRINT 1.

Equipo de arquitectura

- Preparación de paquete de instalación para SPRINT 2.

Equipo de pruebas:

- Pruebas de la versión SPRINT 2.

SPRINT 3:

Equipo de desarrollo:

- Requisito 2.1: Consulta de incidencias.
- Requisito 2.1.1: Volcado de incidencias a hoja Excel.
- Requisito 1.1: Validación de acceso de usuario a través de LDAP.

Equipo de arquitectura

- Preparación de paquete de instalación para versión SPRINT 2.

Equipo de pruebas:

- Pruebas de la versión SPRINT 3.

Debido a que los requisitos de la aplicación web no están cerrados, y a que el usuario final participa activamente en todo el proceso de creación de la aplicación web, no se puede descartar que surjan más Sprints para añadir funcionalidad a la aplicación, o modificar parte de la funcionalidad ya existente.



ANÁLISIS Y DISEÑO

En esta sección hablaremos de cómo se realiza el análisis y diseño de la aplicación web. La forma de hacer tanto el análisis como el diseño está condicionada por la metodología elegida, ya que en Scrum el análisis y diseño se va realizando sobre los requisitos incluidos en cada iteración. Esto implica que no hay una fase de análisis y diseño inicial que incluya todos los requisitos, como es habitual en otras metodologías. Por esto se ha decidido dividir esta sección entre los distintos sprints que se realiza, incluyendo el análisis y diseño concreto para cada uno.

La forma de hacer el análisis y diseño también se ve impactada por el uso de la metodología Scrum. En las metodologías ágiles lo que se pretende es que a partir de los requerimientos del cliente se realice un análisis funcional y diseño en equipo, con la idea de que cada miembro del equipo tenga la posibilidad de aportar ideas. El resultado de este trabajo en equipo es una serie de tareas a desarrollar.

Para cada sprint se realiza un análisis funcional de los requisitos: identificación de casos de uso, estudio del modelo de datos, y de los interfaces de usuario. Posteriormente se realiza un diseño técnico detallado basado en el análisis funcional.

ANÁLISIS Y DISEÑO DEL SPRINT 1

Como se ha comentado previamente estos son los requisitos incluidos en el sprint:

- Requisito 1: Control de acceso y perfiles de usuario.
- Requisito 3.1: Gestión de usuarios.
- Requisito 3.2: Gestión de campos maestros.

Primero se va a realizar el análisis de estos requisitos, y posteriormente se definirá el diseño técnico.

ANÁLISIS FUNCIONAL SPRINT 1

A partir de los requisitos contenidos en el sprint se va a ir especificando cuales son los posibles casos de uso.



CASOS DE USO

Requisito 1: Control de acceso y perfiles de usuario.

Requisito 3.1: Gestión de usuarios.

Los requisitos 1 y 3.1 se consideran parte de una misma funcionalidad. La aplicación web va a disponer de una serie de usuarios registrados. Estos usuarios se utilizarán tanto para realizar un control de acceso a la aplicación, como para saber el tipo de perfil que tienen, y a que opciones de la aplicación pueden acceder.

Los usuarios tendrán asociado un perfil de uso. Sólo aquellos usuarios con un perfil de uso de administrador podrán realizar la gestión de usuarios, pudiendo crear y borrar usuarios, o modificar los ya existentes.

Caso de uso 1.1 – Acceso a la aplicación

Este caso de uso define como se realiza el acceso a la aplicación web.

Precondiciones: Ninguna

Flujo:

- El usuario accede al servidor web, indicando la URL de la aplicación web.
- Se muestra la página de acceso de la aplicación, que permite al usuario introducir su usuario y contraseña.
- El usuario introduce los datos y se envían al servidor web.
- Se realiza la validación de los datos.

Resultado: En caso de que el usuario se haya validado correctamente se permitirá el acceso a la página principal de la aplicación web. En caso de que la validación falle, se procederá a mostrar de nuevo la página de acceso, indicando que ha habido un error en la validación.

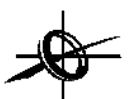
Caso de uso 1.2 – Acceso a la página de gestión de usuarios

Se define como se realiza el acceso a la página de gestión de usuarios

Precondiciones: El usuario debe haberse validado previamente en la aplicación, y además disponer de un perfil que permita realizar la gestión de usuarios.

Flujo:

- El usuario accede al menú de gestión de usuarios, enviándose al servidor web la petición.



Resultado: El servidor web muestra la página de gestión de usuarios. Esta página incluye una lista con los usuarios existentes, que incluye para cada usuario opciones de borrado o modificación de datos. También incluye una opción para crear usuarios.

Caso de uso 1.3 – Creación de usuario

Se define como se realiza la creación de un usuario.

Precondiciones: El usuario debe encontrarse en la página de gestión de usuarios.

Flujo:

- El usuario accede a la opción de creación de usuarios, enviándose al servidor web la petición.
- Se muestra la página de creación de usuarios, que permite al usuario introducir los datos del nuevo usuario.
- El usuario introduce los datos y se envían al servidor web.
- Se realiza la validación de los datos.

Resultado: Si el servidor web considera válidos los datos procede a crear el usuario, mostrando un mensaje de confirmación. Se vuelve a la página de gestión de usuarios, en la que ya aparece el usuario nuevo. Si los datos no son válidos, se muestra un error.

Caso de uso 1.4 – Borrado de usuario

Se define como se realiza el borrado de un usuario.

Precondiciones: El usuario debe encontrarse en la página de gestión de usuarios.

Flujo:

- El usuario pulsa en el botón de borrado asociado al usuario, enviándose al servidor web la petición.

Resultado: Se borra el usuario y se muestra un mensaje de confirmación. Se vuelve a mostrar la página de gestión de usuarios.

Caso de uso 1.5 – Modificación de un usuario

Se define como se modifica un usuario.

Precondiciones: El usuario debe encontrarse en la página de gestión de usuarios.

Flujo:

- El usuario pulsa el botón de modificación asociado al usuario, enviándose al servidor web la petición.
- Se muestra la página de datos de usuario, que permite al usuario visualizar los datos actuales del usuario, pudiendo modificarlos.
- El usuario introduce los datos y se envían al servidor web.

Resultado: Si el servidor web considera válidos los datos procede a modificar el usuario, mostrando un mensaje de confirmación. Se vuelve a la página de gestión de usuarios, en la que aparece el usuario modificado. Si los datos no son válidos, se muestra un error.

Requisito 3.2: Gestión de campos maestros.

Las incidencias disponen de una serie de atributos, algunos de los cuales tienen una serie de valores predefinidos. La gestión de campos maestros consiste en poder modificar los valores predefinidos de estos atributos, pudiendo añadir o eliminar los valores predefinidos.

Sólo aquellos usuarios con un perfil de uso de administrador podrán realizar la gestión de campos maestros.

Caso de uso 1.6 – Acceso a la página de gestión de campos maestros

Se define como se hace el acceso a la página de gestión de campos maestros:

Precondiciones: El usuario debe haberse validado previamente en la aplicación, y además disponer de un perfil que permita realizar la gestión de usuarios.

Flujo:

- El usuario accede al menú de gestión de campos maestros, enviándose al servidor web la petición.

Resultado: Se muestra una página de gestión de campos maestros. Esta página incluye una lista con todos los campos maestros. Desde esta lista es posible seleccionar un campo y pulsar un botón para realizar la modificación del campo seleccionado.

Caso de uso 1.7 – Acceso a página de consulta de valores de un campo maestro

Se define como se hace la consulta de los valores de un campo maestro.

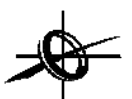
Precondiciones: El usuario debe encontrarse en la página de gestión de usuarios.

Flujo:

- El usuario selecciona un campo maestro y pulsa el botón de consulta, enviándose al servidor web la petición.

Resultado: Se muestra una página que contiene una lista con los valores predefinidos del campo maestro. Para cada valor se incluye una opción de borrado o modificación de datos. También se incluye una opción para crear nuevos valores.

Caso de uso 1.8 - Borrado de un valor predefinido de un campo maestro



Se define como hacer el borrado de un valor de un campo maestro

Precondiciones: El usuario debe encontrarse en la página de consulta de valores de un campo maestro.

Flujo:

- El usuario pulsa sobre la opción de borrado del valor predefinido que desea eliminar. La petición se envía al servidor web.

Resultado: El valor se elimina. Se vuelve a mostrar la página de consulta de valores del campo maestro, en la que ya no aparece el valor predefinido que ha sido eliminado.

Caso de uso 1.9 - Creado de un valor predefinido de un campo maestro

Se define como crear un valor predefinido de un campo maestro.

Precondiciones: El usuario debe encontrarse en la página de consulta de valores de un campo maestro.

Flujo:

- El usuario pulsa el botón de creado. La petición se envía al servidor web.
- Se muestra una página de creación de valores de un campo maestro, que permite introducir los datos del campo maestro.
- El usuario introduce los datos y se envían al servidor web.

Resultado: Si el servidor web considera válidos los datos procede a crear el valor, mostrando un mensaje de confirmación. Se vuelve a la página de consulta de valores del campo maestro, en la que ya aparece el nuevo valor. Si los datos no son válidos, se muestra un error.

Caso de uso 1.10 - Modificación de un valor predefinido de un campo maestro

Se define como hacer la modificación de un valor de un campo maestro

Precondiciones: El usuario debe encontrarse en la página de consulta de valores de un campo maestro.

Flujo:

- El usuario pulsa la opción de modificación del valor predefinido que desea modificar. La petición se envía al servidor web.
- Se muestra una página de modificación de valores de un campo maestro, que permite cambiar los datos del campo maestro.
- El usuario introduce los datos y se envían al servidor web.

Resultado: Si el servidor web considera válidos los datos procede a modificar el valor, mostrando un mensaje de confirmación. Se vuelve a la página de consulta de valores del

campo maestro, en la que ya aparece el nuevo valor. Si los datos no son válidos, se muestra un error.

ESTUDIO DE DATOS:

Para realizar la gestión de usuarios se hace patente que será necesario crear tablas para almacenar siguientes datos:

- Usuarios: se incluirá toda la información necesaria de los usuarios.
- Perfiles: se incluirá toda la información necesaria de los perfiles.

Respecto a la gestión de campos maestros, se optará por crear una tabla por cada campo maestro. En esta tabla se guardarán los valores predefinidos de los campos maestros.

El nombre de las tablas, estructura, y relaciones entre las tablas se discutirá en la sección de diseño técnico para este sprint.

INTERFACES DE SOFTWARE:

1) Acceso a la base de datos

Debido a que es necesario realizar accesos a la base de datos para consultar o almacenar información. En el diseño técnico se comentará como se realiza el acceso a la base de datos.

INTERFACES DE USUARIO:

Teniendo en cuenta que el objetivo de este proyecto es desarrollar una aplicación web, el interfaz que se ofrece al usuario son páginas web. A partir de los distintos casos de uso es posible detallar que páginas web será necesario crear para desarrollar la funcionalidad comprometida en los distintos requisitos.

Para cada página web se estudiará que elementos deben ofrecer para poder realizar su función.

P1.1 - Página de acceso



Esta es la página principal de acceso que se muestra al acceder a la URL de la aplicación desde el navegador.

Casos de uso que requieren esta página: 1.1.

Estructura:

La página contendrá dos campos de texto en los que el usuario podrá incluir su identificador de usuario y contraseña. Además contendrá un botón para realizar la petición de acceso.

P1.2 - Página principal de la aplicación

Esta página será la que se muestre cuando el usuario se haya validado exitosamente.

Casos de uso que requieren esta página: 1.1.

Estructura:

La página contendrá el menú principal de la aplicación, que según el perfil del usuario contendrá enlaces a las distintas funcionalidades a las que el usuario tenga acceso.

Bajo el menú se mostrará una barra de estado, que indicará el identificador del usuario, y contendrá un botón que permitirá al usuario salir de la aplicación.

P1.3 - Página de gestión de usuarios

Esta página será la que se muestre cuando el usuario quiera gestionar los usuarios.

Casos de uso que requieren esta página: 1.2, 1.4

Estructura:

Esta página contendrá una tabla para mostrar los usuarios existentes en la aplicación. Cada fila contendrá los datos del usuario, y unos botones para realizar el borrado o modificación del usuario. Además se incluirá un botón para poder crear nuevos usuarios.

P1.4 - Página de creación/modificación de usuarios

Esta página se utilizará para realizar la creación o modificación de usuarios.

Casos de uso que requieren esta página: 1.3, 1.5

Estructura:

Esta página contendrá campos de texto para los distintos datos del usuario. En caso de utilizarse para una modificación se cargarán los datos del usuario a modificar. La página dispondrá de un botón para realizar el creado o modificación, según corresponda.

P1.5 - Página de gestión de campos maestros

Esta página se utilizará para realizar la gestión de los campos maestros.

Casos de uso que requieren esta página: 1.6



Estructura:

Se mostrará un combo con los posibles campos maestros. También se mostrará un botón para poder consultar los valores del campo maestro seleccionado.

P1.6 - Página de consulta de valores de un campo maestro

Esta página se utilizará para consultar los valores predefinidos de un campo maestro.

Casos de uso que requieren esta página: 1.7, 1.8

Estructura:

Esta página contendrá una tabla para mostrar los valores existentes del campo maestro. Cada fila contendrá los datos valor, y unos botones para realizar el borrado o modificación del valor. Además se incluirá un botón para poder crear nuevos valores.

P1.7 - Página de creación/modificación de un valor predefinido del campo maestro

Esta página se utilizará para crear un valor predefinido de un campo maestro, o modificar uno ya existente.

Casos de uso que requieren esta página: 1.9, 1.10

Estructura:

Esta página contendrá campos de texto para los distintos datos del valor predefinido. En caso de utilizarse para una modificación se cargarán los datos del valor predefinido a modificar. La página dispondrá de un botón para realizar el creado o modificación, según corresponda.

DISEÑO TÉCNICO SPRINT 1

MODELO DE DATOS:

Como se vio al realizar el estudio de datos del análisis funcional, es necesario disponer de tablas para almacenar la información relativa a los usuarios y los perfiles:

Tabla Usuario

En esta tabla se almacenará la información sobre los usuarios de la aplicación. Esta es la estructura:

Campo	Sintaxis	Descripción	Atributos
ID_USUARIO	varchar(8)	Identificador único del usuario	Clave primaria. No nulo



PASSWORD	varchar(60)	Contraseña encriptada	No nulo
ID_PERFIL	varchar(3)	Perfil de usuario	No nulo

Datos iniciales: A la hora de crear la base de datos debe existir al menos un usuario, para que sea posible hacer el acceso a la aplicación. Este usuario además debe tener un perfil de administrador, para que con él se pueda crear al resto de usuarios según convenga. De esta manera se crea un usuario con los siguientes datos:

ID_USUARIO	PASSWORD	ID_PERFIL
'ADMIN'	'ADM1N'	'ADM'

Tabla Perfil

En esta tabla se almacenará la información sobre los perfiles de usuario. Esta es la estructura:

Campo	Sintaxis	Descripción	Atributos
ID_PERFIL	varchar(3)	Perfil de usuario	Clave primaria. No nulo
DESCRIPCIÓN	varchar(250)	Descripción	Valor por defecto NULL

Datos iniciales: A partir de los perfiles indicados en los requisitos de usuario se inicializa la tabla con los siguientes datos:

ID_PERFIL	DESCRIPCIÓN
'ADM'	'Administrador'
'OPE'	'Operario'
'INV'	'Invitado'

Respecto a los campos maestros, se decidió utilizar una tabla para cada campo maestro, en la que se guarde la información de cada uno de los valores predefinidos. En los requisitos se especifica que los campos maestros son:

Campo maestro	Descripción
Proyecto	Define el proyecto en el que se produce la incidencia
Criticidad	Define como de crítica es una incidencia
Área resolutoria	Define los distintos equipos encargados de la resolución de incidencias



De cara a poder utilizar las mismas páginas para realizar la gestión de los campos maestros se opta por que todas las tablas tengan los mismos campos. De esta forma estas son las tablas que se debe crear:

Tabla Proyecto

En esta tabla se almacenará los valores predefinidos para el campo maestro Proyecto:

Campo	Sintaxis	Descripción	Atributos
ID	int(11)	Identificador único del valor	Clave primaria. No nulo. Auto-incremento
NOMBRE	varchar(5)	Nombre del valor	No nulo
DESCRIPCION	varchar(45)	Descripción del valor	No nulo

Datos iniciales: Debido a que en los requisitos de usuario se especificaba ya una serie de proyectos, se añade los siguientes datos iniciales:

ID	NOMBRE	DESCRIPCION
1	'TPV'	'Terminal de Punto de Venta'
2	'SERVI'	'Servidor'
3	'WEB'	'Aplicación Web'
4	'PCK'	'Comprobador de Precio'

Tabla Criticidad

En esta tabla se almacenará los valores predefinidos para el campo maestro Criticidad:

Campo	Sintaxis	Descripción	Atributos
ID	int(11)	Identificador único del valor	Clave primaria. No nulo. Auto-incremento
NOMBRE	varchar(11)	Nombre del valor	No nulo
DESCRIPCION	varchar(10)	Descripción del valor	No nulo

Datos iniciales: Según los requisitos de usuario estos son los posibles valores:

ID	NOMBRE	DESCRIPCION
1	'BAJA'	'Baja'
2	'MEDIA'	'Media'



3	'ALTA'	'Alta'
---	--------	--------

ACCESO A BASE DE DATOS:

Como se comentó al describir la arquitectura de la aplicación, PHP soporta nativamente el acceso a bases de datos MySQL, es decir que es posible realizar el acceso a la base de datos utilizando funciones de PHP, incluidas en el módulo de MySQL.

Para facilitar al acceso a la base de datos se debe crear una capa intermedia con una serie de funciones dedicadas al acceso a la base de datos. El acceso a la base de datos se debe realizar siempre a través de esta capa.

ASPECTO Y DISEÑO DE LAS PÁGINAS WEB:

- **CSS** (Cascading Style Sheets)

Para darle a la aplicación web un aspecto homogéneo es conveniente utilizar CSS (Cascading Style Sheets). Con CSS lo que se hace es describir la presentación de una página HTML, definiendo el aspecto que tendrán los elementos contenidos por la página.

Como realizar todo el diseño de los CSS de la aplicación desde cero es un proceso costoso, y los resultados no son siempre satisfactorios, resulta conveniente recurrir a alguno de los muchos Framework CSS disponibles para PHP. Estos Frameworks permiten dar a la aplicación un aspecto atractivo de forma sencilla.

El framework elegido para la aplicación es el Skyblue CSS Framework (<https://stanko.github.io/skyblue/>). Se trata de un framework ligero, que sólo dispone de una funcionalidad muy básica, pero suficiente para las necesidades de la aplicación web.

Debido a que la aplicación va a manejar fechas, se decide utilizar un módulo de cara a realizar la gestión a la hora de realizar la selección de fechas

- **JavaScript**



JavaScript se considera el lenguaje de programación de HTML. Se utiliza para facilitar gran cantidad de procesos de las páginas web, con la idea de hacer las páginas más dinámicas.

Principalmente se utilizará para la validación de datos en los formularios. Debido a que se va a manejar datos basados en fechas, se va a utilizar alguno de los módulos de JavaScript disponibles para realizar la selección y validación de fechas. El módulo elegido es DhtmlxCalendar (<https://dhtmlx.com/docs/products/dhtmlxCalendar/>), que es un módulo con licencia libre GNU para realizar la gestión de fechas, ofreciendo un interfaz visual muy atractivo.

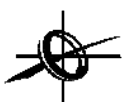
SEGURIDAD:

Como ya se ha visto de cara a proteger el acceso a la aplicación este se hace utilizando una validación por contraseña. Una vez se ha validado el usuario se crea una sesión de PHP. Esta sesión se anula una vez el usuario cierra la aplicación, ya sea cerrando el navegador o pulsando en la opción de cerrar la sesión.

- **Registro de accesos:** Para tener un registro de los accesos de los usuarios a la aplicación, cada vez que se realice un acceso o desconexión de un usuario de la aplicación, se volcará en un log de acceso toda la información (usuario que realiza el acceso, fecha y hora del acceso).
- **Encriptado de contraseñas:** De cara a proteger la contraseña almacenada en la tabla usuario esta debe encriptarse. De tal forma que aunque se disponga de acceso a la tabla usuario no sea posible obtener las contraseñas que utilizan los usuarios.

GESTIÓN DE LOGS:

De cara a tener un registro de los accesos que se realiza a la aplicación, y de los errores que se producen en la misma, se utiliza un sistema de log basado en ficheros. Por un lado se define un nivel de trazas, que indica la cantidad de información que se debe volcar en los logs, y por otro lado un conjunto de ficheros donde se volcará los distintos tipos de información que se desee almacenar en los logs.



CONFIGURACIÓN:

Dentro de la aplicación existen una serie de parámetros configurables:

- Nivel de trazas: indica la cantidad de trazas que volcará en los ficheros de log.
- Ficheros de trazas: ficheros donde se escribirá los logs de trazas.
- Datos de acceso a la base de datos MySQL
- Datos sobre la versión de la aplicación web.

Todos estos datos se almacenan en un fichero de configuración de PHP, definiendo una serie de variables.

ANÁLISIS Y DISEÑO DEL SPRINT 2

Como se ha comentado previamente estos son los requisitos incluidos en el sprint:

- Requisito 2.2: Creación de incidencias

Primero se va a realizar el análisis de estos requisitos, y posteriormente se definirá el diseño técnico.

ANÁLISIS FUNCIONAL SPRINT 2

A partir de los requisitos contenidos en el sprint se va a ir especificando cuales son los posibles casos de uso.

CASOS DE USO

Requisito 2.2: Creación de incidencias

La aplicación web va a disponer de la opción de crear incidencias. Para realizar el creado será necesario introducir toda la información necesaria referente a la incidencia. Esta información se almacenará en la base de datos para su posterior consulta.

Caso de uso 2.1 – Acceso a la página de creación de incidencias

Se define como se hace el acceso a la página de creación de incidencias.

Precondiciones: El usuario debe haberse validado previamente en la aplicación, y además disponer de un perfil que permita crear incidencias.



Flujo:

- El usuario accede al menú de creación de incidencias, enviándose al servidor web la petición.

Resultado: Se muestra una página de creación de incidencias. Esta página incluye campos de texto o combos, según corresponda, para los distintos atributos de la incidencia. Esta página incluye además un botón para enviar la petición de creado de incidencia al servidor.

Caso de uso 2.2 – Creación de una incidencia

Se define como se crea una incidencia.

Precondiciones: El usuario debe encontrarse en la página de creación de incidencias.

Flujo:

- El usuario rellena todos los valores obligatorios para crear la incidencia. Una vez hecho pulsa el botón para enviar la petición.

Resultado: Si se considera válidos los datos introducidos por el usuario la incidencia se crea. Se indica al usuario el identificador de la incidencia recién creada. Si no se considera los datos válidos se muestra un mensaje de error al usuario, y se vuelve página de creado de incidencias, marcando aquellos atributos que no se han considerado correctos.

ESTUDIO DE DATOS:

Para realizar la creación de incidencias se hace patente que será necesario crear tablas para almacenar siguientes datos:

- Incidencia: se incluirá toda la información necesaria de una incidencia.
- Bitácora: se incluirá toda la información de registro de la incidencia

INTERFACES DE USUARIO:

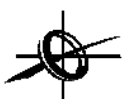
P2.1 - Página de creación de incidencias

Esta página se utilizará para crear una incidencia.

Casos de uso que requieren esta página: 2.1, 2.2

Estructura:

Está página contendrá campos de texto y combos para los distintos atributos de la incidencia. La página dispondrá de un botón para realizar el creado.



MODELO DE DATOS:

Es necesario disponer de una tabla en la que se guardará la información de las distintas incidencias. Esta será la estructura:

Tabla incidencia:

Campo	Sintaxis	Descripción	Atributos
ID_PROBLEMA	int(11)	Identificador único de incidencia	Clave primaria. No nulo. Auto-incremento
ID_PROYECTO	int(11)	Identificador del proyecto	No nulo
ID_CRITICIDAD	int(11)	Identificador de criticidad	No nulo
ID_ESTADO	int(11)	Identificador del estado de la incidencia	No nulo
DESCRIPCION_CORTA	varchar(100)	Descripción breve de la incidencia	No nulo
DESCRIPCION_LARGA	varchar(300)	Descripción larga de la incidencia	No nulo
FECHA_DENUNCIA	timestamp(6)	Fecha de detección de la incidencia	
APP_AFECTADAS	varchar(100)	Aplicaciones afectadas	
ANALISIS	varchar(500)	Análisis de la incidencia	
AFECTACION	varchar(500)	Descripción de las áreas afectadas	

Las incidencias llevan una bitácora asociada para mantener un registro de los cambios que se producen en la incidencia. Se crea una tabla para almacenar este registro:

Tabla bitácora:

Campo	Sintaxis	Descripción	Atributos
ID_BITACORA	int(11)	Identificador único de una entrada de registro	Clave primaria. No nulo. Auto-incremento
ID_PROBLEMA	int(11)	Identificador del problema	No nulo



		asociado	
DESCRIPCION	varchar(500)	Descripción del registro	No nulo
FECHA	timestamp(6)	Fecha de creación del registro	No nulo

ANÁLISIS Y DISEÑO DEL SPRINT 3

Como se ha comentado previamente estos son los requisitos incluidos en el sprint:

- Requisito 2.1: Consulta de incidencias
- Requisito 2.1.1: Volcado de incidencias a hoja Excel
- Requisito 1.1: Validación de acceso de usuario a través de LDAP

Primero se va a realizar el análisis de estos requisitos, y posteriormente se definirá el diseño técnico.

ANÁLISIS FUNCIONAL SPRINT 3

A partir de los requisitos contenidos en el sprint se va a ir especificando cuales son los posibles casos de uso.

CASOS DE USO

Requisito 2.1: Consulta de incidencias

La aplicación web va a disponer de la opción de consulta de incidencias. Como se indica en el requisito se dispondrá de una serie de filtros para poder seleccionar las incidencias que se va a visualizar.

Caso de uso 3.1 – Acceso a la página de listado de incidencias

Se define como se hace el acceso a la página de listado de incidencias.

Precondiciones: El usuario debe haberse validado previamente en la aplicación, y además disponer de un perfil que permita crear incidencias.

Flujo:

- El usuario accede al menú de listado de incidencias, enviándose al servidor web la petición.

Resultado: Se muestra una página de listado de incidencias. Esta página incluye los filtros que se ha designado para poder seleccionar que incidencias se quiere visualizar. También



incluye un área para mostrar las incidencias, que inicialmente estará vacía. Esta página incluye además un botón para enviar la petición de listado de incidencias al servidor.

Caso de uso 3.2 – Filtrado de incidencias

Se define como se hace el filtrado de las incidencias.

Precondiciones: El usuario debe encontrarse en la página de listado de incidencias.

Flujo:

- El usuario rellena los filtros que desee aplicar, suponiendo que quiera aplicar alguno.
- El usuario pulsa el botón de consulta para aplicar el filtro y visualizar las incidencias.

Resultado: Se vuelve a mostrarla página de filtrado de incidencias, pero en este caso la página mostrará el filtro seleccionado por el usuario. Además se mostrará todas las incidencias que cumplan el filtro.

Requisito 2.1.1: Volcado de incidencias a hoja Excel

Como se indica en los requisitos será posible volcar la información de las incidencias que se visualice a una hoja Excel. Será posible hacer un volcado reducido de la información, o de toda la información de la incidencia.

Será necesario disponer de librerías que faciliten la creación de un fichero Excel. Esto se estudiará cuando se realice el diseño técnico de este sprint.

Caso de uso 3.3 – Volcado de información de incidencias a hoja Excel

Se define como obtener un volcado de todas las incidencias que se esté visualizando a una hoja Excel.

Precondiciones: El usuario debe encontrarse en la página de listado de incidencias, y se debe estar mostrando al menos una incidencia.

Flujo:

- El usuario pulsa el botón de para exportar las incidencias que se esté visualizando a una hoja Excel, la petición se envía al servidor.

Resultado: Se genera una hoja Excel con la información resumida de las incidencias que se esté visualizando. El servidor envía la hoja al navegador del usuario.

Caso de uso 3.4 – Volcado de información detallada de incidencias a hoja Excel



Se define como obtener un volcado detallado de todas las incidencias que se esté visualizando a una hoja Excel.

Precondiciones: El usuario debe encontrarse en la página de listado de incidencias, y se debe estar mostrando al menos una incidencia.

Flujo:

- El usuario pulsa el botón de para exportar el detalle de las incidencias que se esté visualizando a una hoja Excel, la petición se envía al servidor.

Resultado: Se genera una hoja Excel con la información detallada de las incidencias que se esté visualizando. El servidor envía la hoja al navegador del usuario.

Requisito 1.1: Validación de acceso de usuario a través de LDAP

Lo que se pretende es que la validación de usuarios se realice a través de un servidor LDAP. Realmente no es necesario un caso de uso concreto para este requerimiento, se modificará el caso de uso 1.1 – Acceso a la aplicación, definido en el Sprint 1.

Será necesario disponer de librerías que faciliten el acceso al servidor LDAP. Esto se estudiará cuando se realice el diseño técnico de este sprint.

ESTUDIO DE DATOS:

En el caso del requisito 1.1 es necesario añadir un campo más a la tabla usuario para indicar si la validación de hace de forma local o por LDAP. En el diseño funcional se concreta este cambio.

INTERFACES DE SOFTWARE:

1) Creación de ficheros Excel XLS

La creación de hojas de cálculo Excel figura como parte del requisito 2.1.1: Volcado de incidencias a hoja Excel. En el diseño técnico se ha estudiado cómo realizarlo.

2) Acceso a servicio LDAP



Es necesario considerar como se realizará el acceso y validación con LDAP, de cara a cumplir con el requisito 1.1: Validación de acceso de usuario a través de LDAP. En el diseño técnico se ha estudiado cómo realizarlo.

INTERFACES DE USUARIO:

P3.1 - Página de consulta de incidencias

Esta página se utilizará para crear una incidencia.

Casos de uso que requieren esta página: 3.1, 3.2, 3.3, 3.4

Estructura:

Está página se divide en tres partes. Una primera parte donde se localizará los filtros de incidencias. Una segunda parte donde se visualizará la información de las incidencias. Y por último una parte donde se contendrá la barra de botones con las posibles opciones: consulta, exportar incidencias a Excel, y exportar detalle de incidencias a Excel.

DISEÑO TÉCNICO SPRINT 3

MODELO DE DATOS:

Según se comentó en el análisis funcional de este sprint es necesario añadir un campo a la tabla usuario para indicar si la validación se hace de forma local o por LDAP:

Tabla Usuario

Se añade el siguiente campo:

Campo	Sintaxis	Descripción	Atributos
ID_LDAP	tinyint(1)	Usar LDAP para validar	No nulo. Valor por defecto 0

Datos iniciales: Se mantiene los mismos datos iniciales, aplicándose el valor por defecto 0 para el nuevo campo.

ACCESO A SERVICIO LDAP:



Debido a que PHP soporta desde la versión 4 el acceso a LDAP no es necesario buscar n software de terceros para gestionar la conexión. Lo que se decide es simplemente implementar una serie de funciones para realizar el acceso al servidor LDAP y la validación de la contraseña del usuario, utilizando las funciones nativas de PHP.

EXPORTACIÓN DE INCIDENCIAS A EXCEL:

Para realizar la exportación de datos de incidencias a Excel será necesario disponer de una librería que permita construir la hoja Excel necesaria. El Excel que se va a construir es bastante básico: sólo se desea volcar datos a la hoja Excel (no es necesario soporte para formulas) y estos datos tienen un formato sencillo.

Lo que se busca es alguna solución GNU para PHP que resulte fácil de usar y que resulte compatible con el fichero Excel que se desea construir, que como se ha comentado no es muy complejo. Finalmente se elige utilizar la librería de licencia libre PHPEXcel (<https://github.com/PHPOffice/PHPEXcel>).

PHPEXcel consta de una serie de librerías para facilitar tanto la escritura como la lectura de ficheros Excel. Este proyecto se basa en el estándar de Windows OpenXML, y está escrito en código PHP.



IMPLEMENTACIÓN

En este apartado se va a explicar cómo se ha realizado la implementación de la aplicación web que se desea desarrollar. Al igual que se ha hecho en los apartados de análisis funcional y diseño técnico, se ha decidido dividir este apartado entre los distintos sprints.

IMPLEMENTACIÓN DEL SPRINT 1

ACCESO A BASE DE DATOS:

Como se comentó en el diseño funcional de este script es necesario crear una capa intermedia con una serie de funciones dedicadas al acceso a la base de datos. El acceso a la base de datos se debe realizar siempre a través de esta capa.

- Configuración

Es necesario almacenar en algún lugar la información necesaria para realizar el acceso a la base de datos. Se crea un fichero PHP llamado BBDD/configBBDD.php en el que se almacenará toda la información: usuario de base de datos, contraseña, IP o nombre del servidor de base de datos, etc. Esto se hace utilizando un array indexado de PHP.

- Funciones de acceso a base de datos

Se crea un fichero PHP llamado BBDD/BBDD.php en el que se almacenan las funciones de acceso a la base de datos. También se incluye funciones para realizar la comparación y conversión de datos.

- Consultas de base de datos

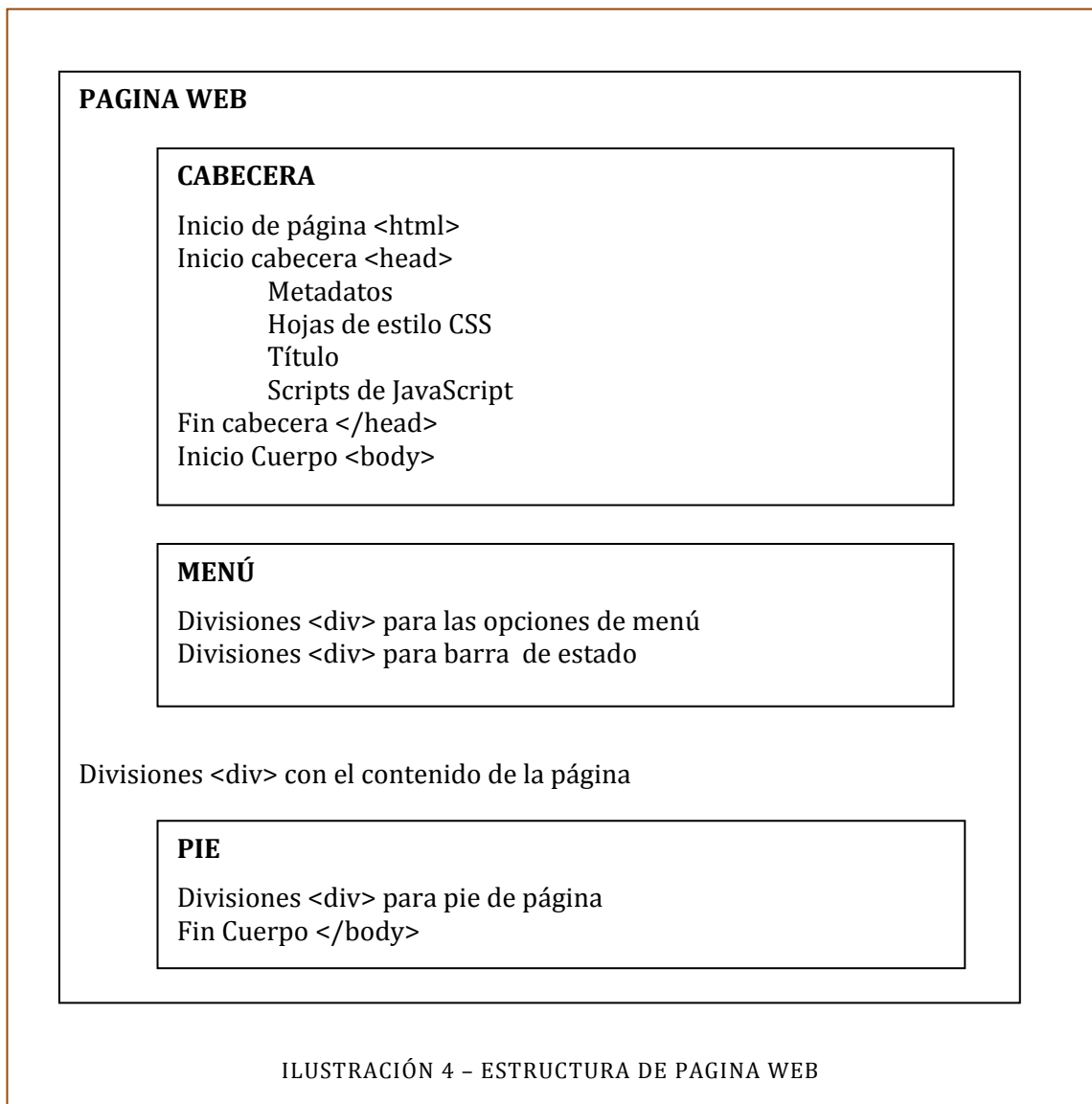
De cara a tener todas las consultas de base de datos agrupadas en un único fichero se crea el fichero BBDD/sql.php, que contiene un array indexado con todas las consultas que se realizan en la aplicación.



ESTRUCTURA DE LAS PÁGINAS WEB:

Como se comento en el diseño técnico del sprint 1, se va a utilizar el Framework CSS Skyblue, con la idea de darle un aspecto atractivo y homogéneo a las páginas de la aplicación web.

Las páginas web se van a estructurar de forma modular, para poder reutilizar el mismo código en distintas páginas. La estructura básica será la siguiente:



De esta forma existirá un fichero PHP para la página, que a su vez podrá incluir un fichero PHP con la cabecera, y otro con el menú. Los ficheros PHP de cabecera y menú pueden ser compartidos entre varios ficheros PHP de página. Las páginas podrán contener al final un pie.

Se creará en la estructura de PHP un directorio cabecera, donde se guardarán los ficheros PHP con las distintas cabeceras de las páginas web, y otro directorio en el que se incluirán los ficheros PHP con los distintos menús de las páginas web.

VALIDACIÓN DE USUARIOS:

Como se ha comentado el acceso a la aplicación se hace con un control de acceso por usuarios. La validación de estos usuarios puede ser local o por LDAP. Como se vio en el análisis funcional para realizar el acceso a la aplicación se define las siguientes páginas:

P1.1 - Página de acceso – index.php

Se crea el fichero index.php, que será la página de acceso de la aplicación.

- Incluye la cabecera cabecera.php
- No incluye ningún menú, ya que no es necesario.

Contiene los campos de texto para introducir usuario y contraseña, y un botón para realizar el acceso.

P1.2 - Página principal de la aplicación – inicio.php

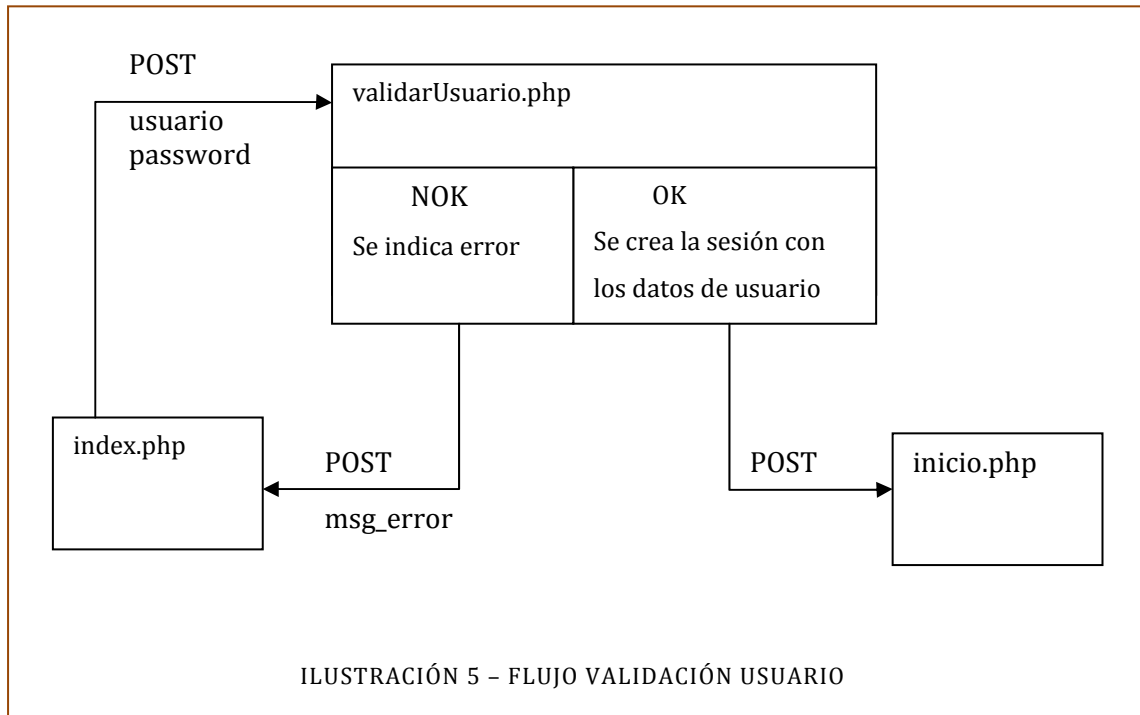
Se crea el fichero inicio.php, que será la página de inicio una vez se ha validado el usuario.

- Incluye la cabecera cabecera.php
- Incluye menú menu.php

No tiene contenido, sólo muestra las opciones de menú:

Este será el flujo para la validación:





controlAcceso.php

De cara a evitar el acceso a la aplicación sin haber realizado previamente una validación, se crea un fichero PHP que se encarga de ver que la sesión PHP contiene la información sobre el usuario, y comprobar que se ha realizado la validación. Si la sesión contiene los datos que se introduce cuando se realiza la validación se entiende que el usuario no está validado, y se le redirige a la página de inicio.

De esta forma todas las páginas PHP deben contener al comienzo el fichero controlAcceso.php, para validar que se ha realizado una validación antes de acceder a la página.

MENÚ DE ADMINISTRACIÓN:

Cuando se crea la barra de menús se consulta los datos de sesión, para revisar el perfil del usuario y saber si debe disponer de las opciones de administrador. Aunque inicialmente se decidió disponer de una página principal para la gestión de usuarios, y otra para la gestión de campos maestros, se va a crear una página de administración, que contendrá las funciones de la página principal de gestión de campos maestros, y un acceso a la página principal de usuarios.



De esta forma se cambia la página P1.5, para que en lugar de ser la página de gestión de campos maestros sea la página principal de administración:

P1.5 – Página principal de administración – menuAdministrador.php

Se crea el fichero menuAdministrador.php, que será la página principal de administración.

- Incluye la cabecera cabeceraAdministrador.php
- Incluye menú menu.php

Se divide en dos partes. Por un lado para realizar la gestión de campos maestros contendrá un combo para seleccionar el campo maestro que se quiere seleccionar., y un botón para realizar la modificación del campo maestro seleccionado. Por otro lado contendrá un botón para ir a la gestión de usuarios.

GESTIÓN DE USUARIOS:

Como se vio en el análisis funcional y diseño técnico es necesario crear las siguientes páginas para realizar la gestión de usuarios:

P1.3 - Página de gestión de usuarios – administrarUsuarios.php

Se crea el fichero administrarUsuarios.php.

- Incluye la cabecera cabeceraAdministrador.php
- Incluye el menú menu.php

Como se indicó en el análisis funcional la página contiene una tabla con la lista de usuarios existentes en la aplicación. En la última columna de la tabla se habilitan dos botones para poder realizar la modificación o borrado del usuario. Además se incluye un botón para poder crear nuevos usuarios.

P1.4 - Página de creación/modificación de usuarios – registroUsuario.php

Se crea el fichero registroUsuario.php.

- Incluye la cabecera cabeceraAdministrador.php
- Incluye el menú menu.php

Se comprueba si se está creando o modificando un usuario. Si se trata de una modificación en los campos de la página se introduce la información del usuario a modificar.

GESTIÓN DE CAMPOS MAESTROS:



Debido a que la ventana principal de gestión de campos maestros se ha integrado en la página principal de administración, ya sólo son necesarias estas páginas:

P1.6 - Página de consulta de valores de un campo maestro – administrarCampoMestro.php

Se crea la página administrarCampoMaestro.php para visualizar los valores de un campo maestro, y dar opción a modificarlos.

- Incluye la cabecera cabeceraAdministrador.php
- Incluye el menú menu.php

En esta página se muestra una tabla con los valores del campo maestro. Para cada fila se incluye una opción de modificación o borrado. Se incluye también un botón para crear un nuevo valor.

P1.7 - Página de creación/modificación de un valor predefinido del campo maestro – registroCampoMaestro.php

Se crea el fichero registroCampoMaestro.php, para realizar la modificación de un valor de un campo maestro.

- Incluye la cabecera cabeceraAdministrador.php
- Incluye el menú menu.php

Se incluye un campo de texto para el nombre del valor, y otro para la descripción. En caso de utilizarse para una modificación se cargarán los datos del valor a modificar. Se añade un botón para realizar la modificación, y otro para volver a la página de consulta.

IMPLEMENTACIÓN DEL SPRINT 2

Este sprint se basa en la creación de incidencias.

ACCESO A BASE DE DATOS:

Se añaden todas las funciones y consultas de SQL necesarias para realizar la creación de la incidencia.

PÁGINA DE CREADO DE INCIDENCIAS:

Durante el análisis funcional se definió que sólo es necesario añadir una página:



P2.1 - Página de creación de incidencias – menuCreacionIncidencias.php

Se crea el fichero menuCreacionIncidencias.php, desde esta página se realizará la creación de una incidencia.

- Incluye la cabecera cabeceraRegistro.php
- Incluye el menú menu.php

Se incluyen campos de texto o combos, según corresponda, para incluir la información de la incidencia. Aquellos atributos que no se puedan introducir al crear una incidencia se dejarán deshabilitados, y los que se obligan a incluir se marcarán en rojo. El estado de la incidencia sólo mostrará el valor "Nueva". La página contendrá al final un botón para realizar la creación de la incidencia.

IMPLEMENTACIÓN DEL SPRINT 3

En este sprint se incluye todo lo necesario para realizar la consulta de incidencias.

ACCESO A BASE DE DATOS:

Se incluyen todas las funciones y consultas de SQL necesarias para realizar el filtrado de incidencias.

PÁGINA DE CONSULTA DE INCIDENCIAS:

P3.1 - Página de consulta de incidencias – menuConsultaIncidencias.php

Se crea el fichero menuConsultaIncidencias.php para realizar la consulta de incidencias.

- Incluye la cabecera cabeceraConsulta.php
- Incluye el menú menu.php

Como se comentó se divide la página en tres partes. La superior contiene los distintos filtros para elegir las incidencias. En esta se incluye una serie de combos o campos de texto para introducir los filtros.

La parte media de la página se utilizará para mostrar una tabla con las incidencias que cumplan con las opciones de filtrado. Si no hay ningún filtro aplicado no se mostrará la tabla.

La parte inferior contendrá el botón para aplicar el filtro, y además los botones para realizar la exportación de las incidencias filtradas a una hoja Excel. Sólo se podrá realizar



la exportación si hay algún filtro aplicado, si no lo hay se mostrará los botones, pero deshabilitados.



PRUEBAS Y CERTIFICACIÓN

Al seguir la metodología de desarrollo Scrum las pruebas se realizan al final de cada sprint, para probar la funcionalidad incluida en la iteración que se ha concluido. De esta forma para cada iteración se define un plan de pruebas básico, relativo a los requisitos incluidos.

Como se comentó al no disponerse de una herramienta para realizar el seguimiento de las incidencias detectadas, que es justo el objetivo de la aplicación que se desarrolla, se utiliza un sistema rudimentario de hojas Excel para registrar las incidencias que se va localizando en la aplicación, y su evolución.

Para este desarrollo en concreto se da la situación de que las personas dedicadas a realizar las pruebas son al mismo tiempo los usuarios finales de la aplicación. Esto hace que la certificación de la aplicación se pueda ir realizando conjuntamente con las pruebas.



MANUALES Y DOCUMENTACIÓN

Dado que la idea de los sprints de Scrum es generar una iteración de la aplicación a desarrollar que sea funcional, se incluye dentro de la iteración la necesidad de generar todos los documentos necesarios:

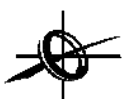
- Documentos de análisis y diseño
Incluyen el análisis funcional y diseño técnico.
- Manual de usuario
Manual para el usuario final de la aplicación.
- Manual de administración
Manual de administrador, incluye como hacer la instalación y configuración inicial de la aplicación.
- Plan de pruebas
Plan con las distintas pruebas a realizar

Los documentos se construyen de forma incremental, de tal forma que cuando se concluye un sprint se parte de los documentos del sprint anterior, y se añade todo lo que se ha realizado durante el sprint.

La implementación del código incluye también realizar una correcta documentación. Al desarrollarse en PHP se incluye comentarios compatibles con PHPDoc, que es el estándar oficial de documentación para PHP.

MANUAL DE USUARIO

Se incluye en esta sección el manual de usuario completo de la aplicación web:



APLICACIÓN WEB: GESTIÓN DE INCIDENCIAS (SGINC)

Manual de usuario

Índice

I.	Introducción.....	2
II.	Acceso a la aplicación.....	2
III.	Página de inicio.....	3
IV.	Creación de incidencias	4
V.	Consulta de incidencias	5
V-I.	Filtrado de incidencias.....	6
V-II.	Exportar datos a una hoja Excel.....	7
VI.	Administración.....	8
VI-I.	Gestión de tablas maestras.....	8
VI-II.	Gestión de usuarios.....	10



I. Introducción

Este manual explica cómo utilizar la aplicación web sgINC. Esta aplicación permite realizar una gestión centralizada de las incidencias detectadas en una serie de sistemas de software. Con ella es posible registrar incidencias y gestionarlas. También proporciona herramientas para generar informes sobre las distintas incidencias registradas y su estado.

II. Acceso a la aplicación

El acceso a la aplicación se hace a través de unas credenciales de acceso (usuario y contraseña). En caso de que el usuario no tenga credenciales de acceso debe ponerse en contacto con el administrador de la aplicación para solicitar la creación de las credenciales de acceso. El portal de entrada de la aplicación está disponible en la URL:

<http://sginc/>

Esta es la página de acceso:



sgINC

Usuario

Contraseña

Acceder



El usuario debe introducir su identificador de usuario y contraseña, y pulsar el botón <Acceder>, teniendo en cuenta que la contraseña distingue entre mayúsculas y minúsculas. Si las credenciales del usuario se consideran correctas, se accederá a la página de inicio. En caso negativo se mostrará un error indicando cual ha sido el problema.

III. Página de inicio

Es la página principal de la aplicación. En ella se dispone de los menús para acceder a las distintas áreas de la aplicación.

Esta es la página de inicio:



En la parte superior se muestra una barra de menú con las siguientes opciones:

- Menú <Consulta de Incidencias>
Acceso a la página de consulta de indecencias
- Menú <Creación de Incidencias>
Acceso a la página de creación de incidencias
IMPORTANTE: Sólo usuarios con perfil de administrador u operario dispondrán de esta opción de menú
- Menú <Administración>
Acceso a la página de administración
IMPORTANTE: Sólo usuarios con perfil de administrador dispondrán de esta opción de menú

(3)



- Menú <Acerca De>

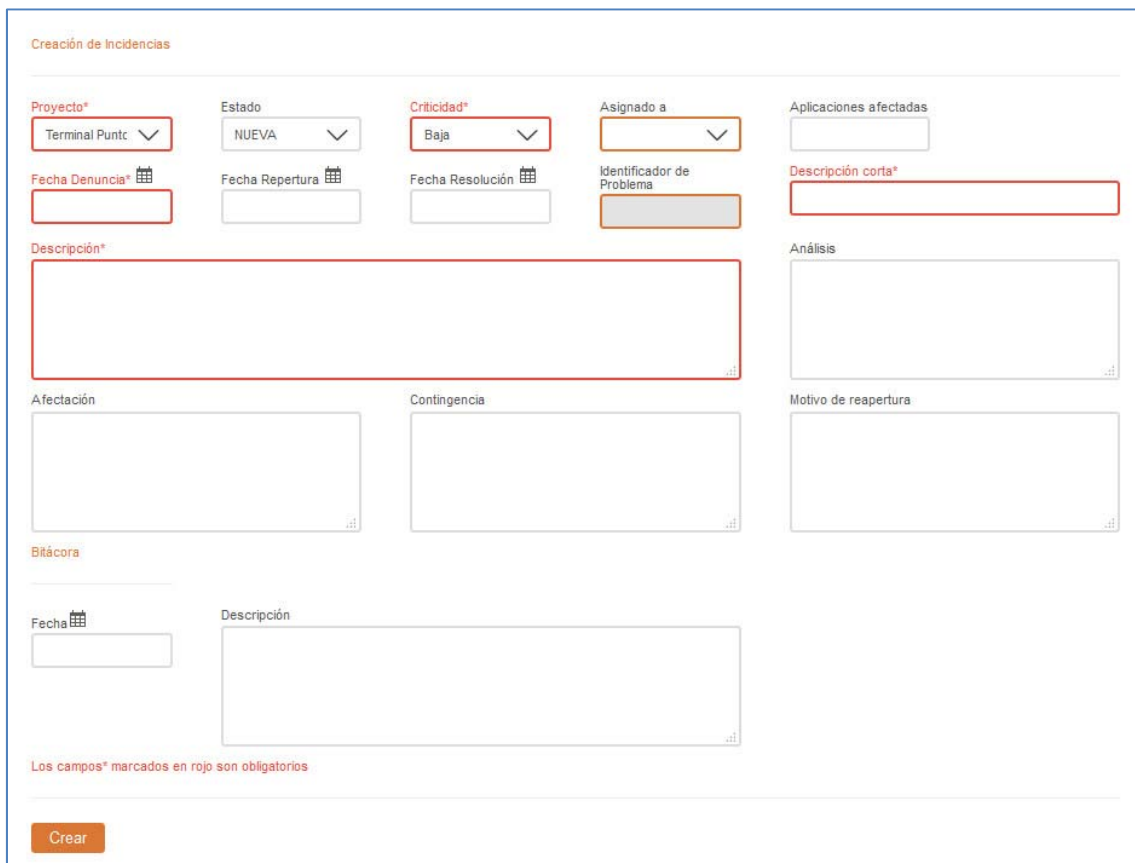
Muestra información sobre la versión de la aplicación web

La parte inferior aparece vacía, hasta el momento que el usuario elige una opción.

IV. Creación de incidencias

Para acceder a esta funcionalidad ese debe pulsar sobre el botón <Creación de Incidencias> del menú. La funcionalidad de de creación de incidencias está accesible únicamente para aquellos usuarios con perfil administrador u operario.

Esta es la página de creación de incidencias:



Formulario de Creación de Incidencias:

Campos obligatorios (marcados en rojo):

- Proyecto* (lista desplegable: Terminal Punc)
- Estado (lista desplegable: NUEVA)
- Criticidad* (lista desplegable: Baja)
- Asignado a (lista desplegable)
- Fecha Denuncia* (fecha)
- Fecha Repertura (fecha)
- Fecha Resolución (fecha)
- Identificador de Problema (campo gris)
- Descripción corta* (campo de texto)
- Descripción* (campo de texto grande)
- Afectación (campo de texto)
- Contingencia (campo de texto)
- Motivo de reapertura (campo de texto)
- Fecha (fecha)
- Descripción (campo de texto grande)

Campos opcionales:

- Aplicaciones afectadas (campo de texto)
- Análisis (campo de texto)

Botón: Crear

Nota: Los campos* marcados en rojo son obligatorios

En esta página el usuario debe incluir la información de la incidencia que desea crear. Aquellos campos que es obligatorio rellenar aparecerán marcados en rojo.



Una vez incluida toda la información se debe pulsar el botón <Crear> para iniciar la creación de la incidencia. Los datos introducidos se verificarán, y si todo se considera correcto se procederá a crear la incidencia, y una vez creada se mostrará un mensaje al usuario indicando que la incidencia se ha creado:

El nuevo problema se ha creado correctamente.

Aceptar

Las incidencias creadas podrán ser revisadas en la página de consulta de incidencias.

V. Consulta de incidencias

Para acceder a esta funcionalidad ese debe pulsar sobre el botón <Consulta de Incidencias> del menú. Esta funcionalidad está disponible para cualquier perfil de usuario.

Esta es la página de consulta de incidencias:

Consulta de Incidencias

Identificador de problema	Proyecto	Estado	Criticidad	Campo	Texto de búsqueda
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha Desde	Fecha Hasta				
<input type="text"/>	<input type="text"/>				
Consulta	Exportar a Excel	Exportar a Excel (detalle)			

(5)



V-I. Filtrado de incidencias:

En esta página se ofrece la posibilidad de visualizar aquellas incidencias que cumplen unos determinados filtros. En la parte superior se muestra los distintos filtros que es posible aplicar:

- Identificador de problema
Filtrar por un identificador de problema dado
- Proyecto
Filtrar por el proyecto al que está asignada la incidencia
- Estado
Filtrar por el estado de la incidencia
- Criticidad
Filtrar por la criticidad de la incidencia
- Campo / Texto de búsqueda
Permite buscar un texto en alguno de los siguientes campos de la incidencia:
Aplicaciones afectadas
Descripción
- Fecha desde / Fecha hasta
Filtrar por la fecha de denuncia, seleccionando un intervalo de fechas

Es posible aplicar un único filtro o varios filtros a la vez, en cuyo caso se mostrará solamente aquellas incidencias que cumplan todos los filtros.

Una vez seleccionado el filtro que se desea aplicar se debe pulsar el botón <Consulta>. Existe la posibilidad de no introducir ningún filtro, en cuyo caso al pulsar el botón se mostrará todas las incidencias registradas.

Una vez pulsado el botón, si no se encuentra ninguna incidencia que cumpla el filtro se muestra un mensaje indicando que ninguna incidencia cumple el filtro. Si se ha encontrado al menos una incidencia que cumple el filtro se procede a mostrar una lista con las incidencias que cumplen el filtro seleccionado.



Este es el aspecto de la página cuando se muestra la lista de incidencias:

Consulta de Incidencias

Identificador de problema

Proyecto

Estado

Criticidad

Campo

Texto de búsqueda







Fecha Desde

Fecha Hasta

Consulta

Exportar a Excel

Exportar a Excel (detalle)

ID	Proyecto	Criticidad	Estado	Descripción	F. Denuncia	A. Afectadas	F. Resolución	F. Reapertura	
5	Terminal Punto de Venta	Alta	NUEVA	No arranca la caja	11/05/2017 15:05	TODAS			 
4	Aplicacion Web	Alta	ANALISIS	No aparece el menú de cierre de tienda	09/05/2017 17:14				 
3	Servidor	Baja	ASIGNADA	No se procesan los tickets con descuento de empleado	08/05/2017 17:12	MOVI			 

Para cada incidencia mostrada en la lista, en la última columna, se ofrece la posibilidad de editar la incidencia o borrarla, en ambos casos sólo si el usuario tiene perfil de administrador u operario:

- Edición: Para realizar la edición de la incidencia se debe pulsar sobre el icono del lápiz. Esto abrirá la página de edición de la incidencia, que es idéntica en formato a la de creación de incidencias.
- Borrado: Para realizar el borrado de la incidencia se debe pulsar sobre el icono del aspa. Se mostrará una página para pedir confirmación. Si se confirma el borrado se borrará la incidencia, y se volverá a mostrar la página de consulta de incidencias.

V-II. Exportar datos a una hoja Excel

Es posible exportar las incidencias que se está visualizando a una hoja Excel. Para ello existen dos botones, que se encuentran bajo la lista de filtros. Estos botones sólo estarán activos si se está visualizando al menos una incidencia.

A la hora de exportar los datos hay dos posibilidades:

(7)



Exportar datos resumidos:

Se exporta a una hoja Excel las incidencias que se está visualizando. En esta opción sólo se muestra una lista reducida de los campos de la incidencia. Para ello se debe pulsar el botón <Exportar a Excel>

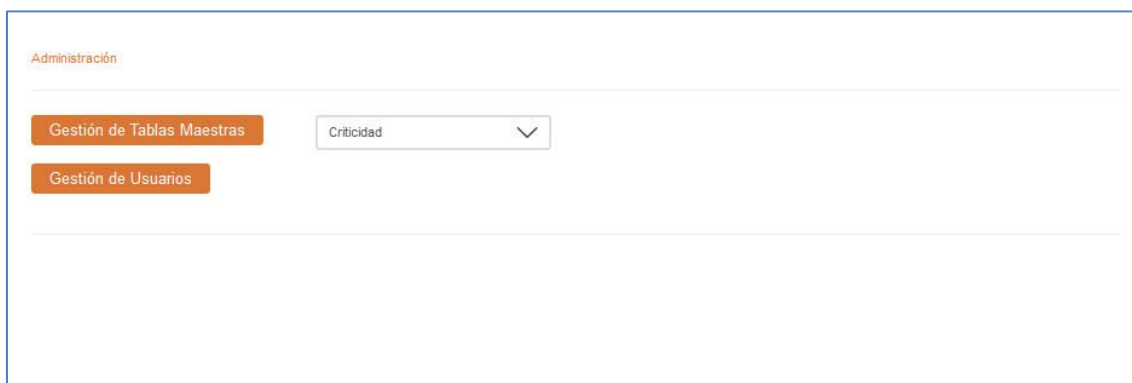
Exportar datos completos:

Se exporta a una hoja Excel las incidencias que se está visualizando. En esta opción se incluye en la hoja todos los campos de la incidencia. Para ello se debe pulsar el botón <Exportar a Excel (detalle)>

VI. Administración

Para acceder a esta funcionalidad ese debe pulsar sobre el botón <Administración > del menú. Esta funcionalidad sólo está disponible para aquellos usuarios con perfil de administrador.

Esta es la página de administración:



Se dispone de estas dos opciones:

VI-I. Gestión de tablas maestras

Se utiliza para revisar los valores de las tablas maestras, que indican los posibles valores de algunos de los campos de las incidencias. Es posible modificar los valores ya existentes, borrarlos, o dar de alta valores nuevos.

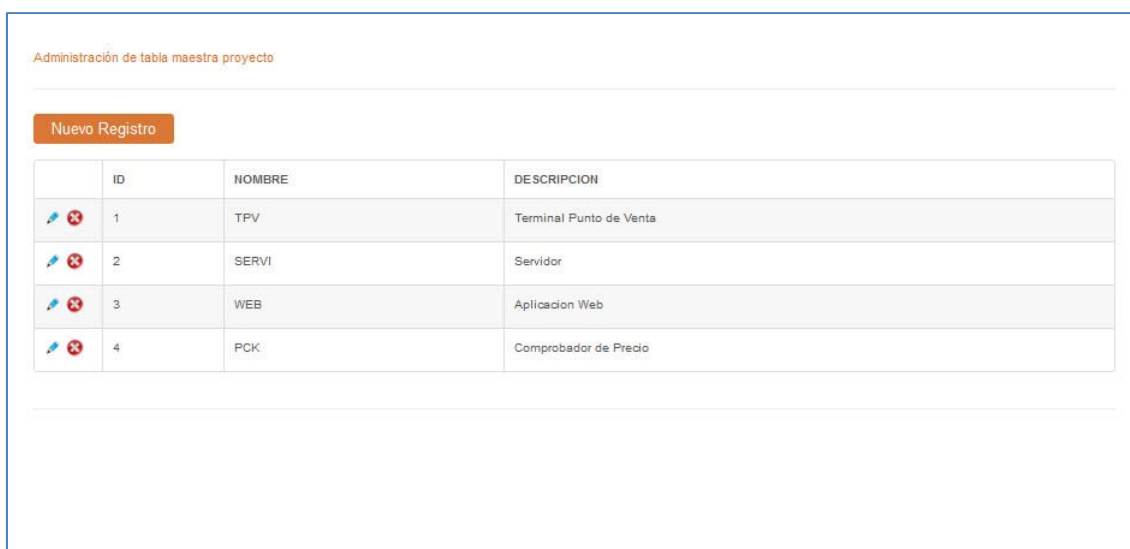










Los campos maestros que es posible modificar son los siguientes:

1. Criticidad: Campo de la incidencia que indica cómo de crítica es.
2. Proyecto: Campo de la incidencia que indica a que proyecto se debe asignar.
3. Estado: Estado de la incidencia

Se debe seleccionar la tabla maestra que se desea modificar y pulsar el botón <Gestión de tablas maestras>

Esto abre la página de modificación de campos maestros. Este es un ejemplo de la página cuando se modifica el maestro Proyecto:

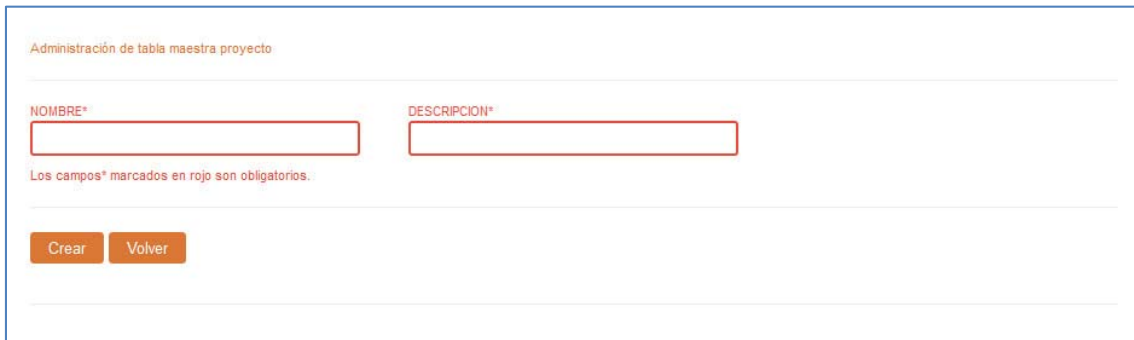


	ID	NOMBRE	DESCRIPCION
 	1	TPV	Terminal Punto de Venta
 	2	SERVI	Servidor
 	3	WEB	Aplicacion Web
 	4	PCK	Comprobador de Precio

La página contiene un botón para crear nuevos valores, y una tabla con los valores existentes. Para cada valor en la primera columna se habilita unos iconos para modificar o borrar el valor.

Creación de un nuevo valor:

Para crear un nuevo valor, se debe pulsar sobre el botón <Nuevo Registro>. Con esto se abrirá una página en la que se introducirá la información del valor a crear. Este es un ejemplo de la página cuando se va a crear un nuevo valor en el maestro Proyecto:



Administración de tabla maestra proyecto

NOMBRE* DESCRIPCION*

Los campos* marcados en rojo son obligatorios.

Crear Volver

El usuario debe introducir el nombre del nuevo valor, y una breve descripción. Una vez introducidos los valores debe pulsar sobre el botón <Crear>.

Borrado de un valor:

Para borrar un valor se debe pulsar sobre el icono del aspa del valor a borrar. Se mostrará una página para pedir confirmación. Si se confirma el borrado se borrará el valor, y se volverá a mostrar la página de consulta de incidencias.

Edición de un valor:

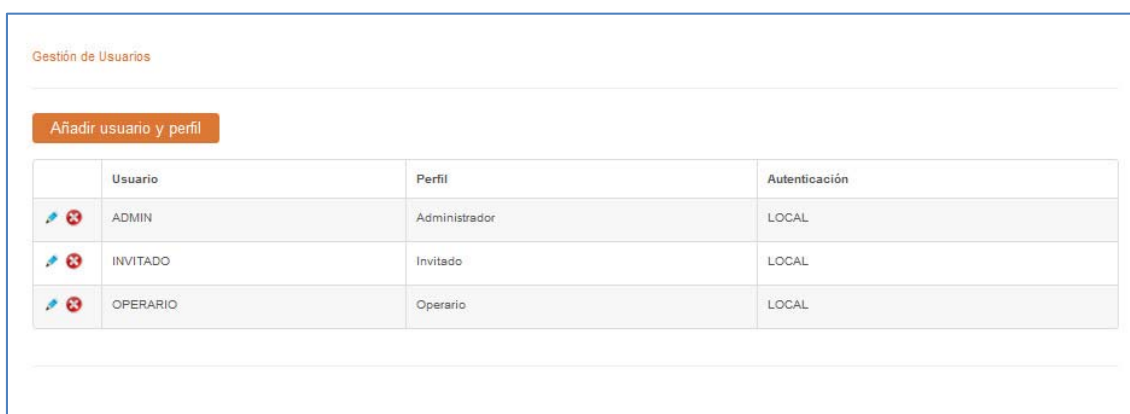
Para editar un valor se debe pulsar sobre el icono del lápiz del valor a editar. Se mostrará una página idéntica a la que se muestra cuando se crea un nuevo valor, sólo que en los campos de la página aparecerán los datos actuales del valor. El usuario debe modificar los datos, y pulsar el botón <Modificar>.







VI-II. Gestión de usuarios:

Se utiliza para revisar los usuarios registrados en la aplicación. Es posible modificar los datos de los usuarios ya existentes, borrar usuarios, y crear usuarios nuevos. Se debe pulsar el botón <Gestión de Usuarios>

Esto mostrará la página de Gestión de usuarios:



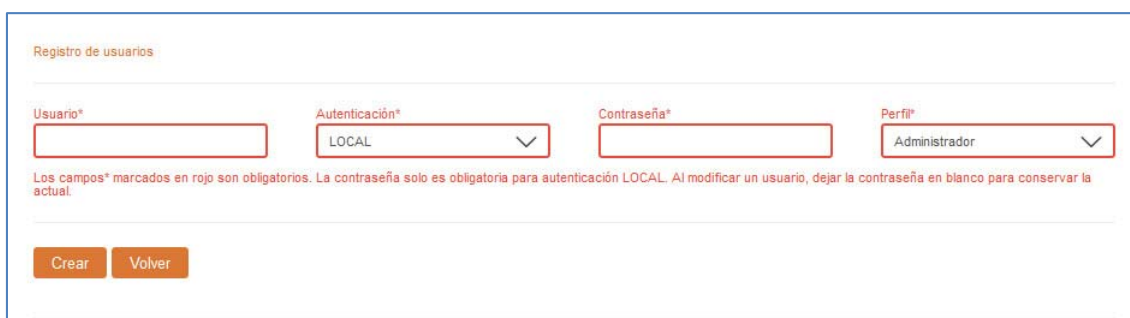


	Usuario	Perfil	Autenticación
 	ADMIN	Administrador	LOCAL
 	INVITADO	Invitado	LOCAL
 	OPERARIO	Operario	LOCAL

La página contiene un botón para crear valores, y una lista con los usuarios existentes.

Creación de un nuevo usuario:

Se debe pulsar el botón <Añadir usuario y perfil>. Esto abrirá la página de creación de usuarios, en la que se introducirá la información del usuario:



Registro de usuarios

Usuario* Autenticación* Contraseña* Perfil*

LOCAL Administrador

Los campos* marcados en rojo son obligatorios. La contraseña solo es obligatoria para autenticación LOCAL. Al modificar un usuario, dejar la contraseña en blanco para conservar la actual.

Crear Volver

Los campos obligatorios están marcados en rojo. Si el usuario tiene validación por LDAP no es necesario poner la contraseña, ya que no es necesario almacenarla. Una vez introducidos los datos se debe pulsar el botón <Crear>

Modificación de un usuario:

Para modificar un usuario se debe pulsar sobre el icono del lápiz del usuario a modificar. Se mostrará una página idéntica a la que se muestra cuando se crea un nuevo usuario, sólo que en los campos de la página aparecerán los datos actuales del usuario. El usuario debe modificar los datos, y pulsar el botón <Modificar>.

(11)



Borrado de un usuario:

Para borrar un usuario se debe pulsar sobre el icono del aspa del usuario a borrar. Se mostrará una página para pedir confirmación. Si se confirma el borrado se borrará el usuario, y se volverá a mostrar la página de gestión de usuarios.





CONCLUSIONES

Como se comentó al inicio, el objetivo de este trabajo era el diseñar y construir una aplicación web para facilitar la gestión de incidencias en una determinada empresa, que hasta el momento estaba utilizando unos sistemas muy básicos para realizar dicha gestión. Este objetivo se considera alcanzado, ya que la aplicación desarrollada supone una importante mejora con respecto a los métodos que se utilizaba inicialmente.

Debido a que había una cierta urgencia por disponer de esta herramienta, se diseñó inicialmente con la funcionalidad básica que se ha descrito en este trabajo, pero con la idea futura de ir añadiendo nuevas funcionalidades. Precisamente por esto se eligió una metodología ágil como Scrum, ya que facilitaba la tarea de disponer de una versión funcional de forma rápida, y además permitía funcionar con requisitos cambiantes. A medida que vayan surgiendo nuevas ideas para añadir funcionalidad a la aplicación se irán implementando nuevas iteraciones en las cuales se añadirá dicha funcionalidad. Como ejemplos de mejoras de la funcionalidad del sistema se podrían citar los siguientes: posibilidad de hacer envíos de correos para notificar los cambios en las incidencias, añadir nuevos filtros a la hora de consultar las incidencias, añadir nuevos campos a las incidencias, etc.

En todo el proceso de creación de esta aplicación web se han ido enfrentando una serie de dificultades y retos:

El proyecto que se ha desarrollado es para uso interno, y va a ser utilizado por distintos departamentos de la empresa: Arquitectura, Desarrollo, Pruebas y Calidad. Por un lado, el disponer de distintos puntos de vista sobre el uso que se debe dar a la aplicación, e incluso qué requisitos debe incluir, hace que se enriquezca la herramienta desarrollada. Pero por otro lado, también puede ser una fuente de conflictos al considerar cada parte sus necesidades como lo más prioritario.

La urgencia a la hora de disponer de la aplicación web hace más complicado todo el proceso de análisis, diseño e implementación. Además al tratarse de un proyecto interno los recursos tienden a ser más limitados, ya que no existe un beneficio directo por el desarrollo obtenido. Por esta misma razón, otros desarrollos no internos que se puedan estar desarrollando en paralelo resultan más prioritarios, y en momentos de conflicto tienen más posibilidades de hacerse con los recursos que necesiten.



La elección de una arquitectura con la que estén familiarizados los distintos participantes en el análisis, diseño e implementación suponen un gran beneficio. No solo facilitan el cumplir con los plazos, al no tener que realizar una formación previa, sino que además reducen la posibilidad de que surjan imprevistos en las distintas fases, como incompatibilidades entre los distintos componentes de la arquitectura, o limitaciones imprevistas. Algunos de los requisitos de la aplicación requerían realizar labores algo complejas, como la conexión a un servidor LDAP, o la exportación de datos a hojas Excel. Esto en principio podía haber supuesto un impacto a nivel de tiempo y de coste, pero debido a que la arquitectura elegida dispone de un gran número de módulos con licencia libre GNU, se ha podido elegir soluciones sencillas y además gratuitas.

Con esta herramienta se demuestra que es posible realizar desarrollos potentes, con múltiples funcionalidades e interfaces, con software libre. Y que ni la calidad ni el rendimiento tienen porqué verse afectados por esta elección.





REFERENCIAS

Documentación sobre la gestión de Incidencias:

AXOSOFT

https://www.axosoft.com/downloads/axowp_importance_of_defect_tracking.pdf

Información general sobre la gestión de incidencias.

ISTQB Exam Certification

<http://istqbexamcertification.com/what-is-a-defect-life-cycle/>

Información sobre el ciclo de vida de las incidencias.

Metodología SCRUM – SCRUM Manager

ProyectosAgiles.org

<https://proyectosagiles.org/que-es-scrum/>

Información sobre la metodología Scrum.

Curso SCRUM Manager - Estratecno

<http://www.estrategno.es/>

Información sobre Scrum Manager.

Arquitectura LAMP:

Wikipedia

<http://es.wikipedia.org/wiki/LAMP/>

Información sobre arquitectura LAMP.

Php.net

<http://www.php.net>

Información sobre PHP: funciones, sintaxis, etc.

Mysql.com

<http://www.mysql.com>

Información sobre MySQL: descarga de productos, documentación, etc.

Módulos externos CSS/PHP/JavaScript:

Skyblue CSS Framework

<https://stanko.github.io/skyblue/>

Framework CSS para definir estilos de página.

DhtmlxCalendar

<https://dhtmlx.com/docs/products/dhtmlxCalendar/>

Módulo JavaScript para gestión de fechas

PHPExcel

<https://github.com/PHPOffice/PHPExcel>

Módulo PHP para la creación de ficheros Excel XLS.

