

**UNIVERSIDAD POLITÉCNICA
DE VALENCIA**

ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA



**INGENIERÍA TÉCNICA DE INFORMÁTICA
DE GESTIÓN**

***MEMORIA
PROYECTO FINAL DE CARRERA***

***APLICACIÓN WEB
BEETRACKER***

**Autor: Alberto Sánchez Gómez
Tutor: Sergio Sáez Barona
Código P.F.C.: DISCA-222
Septiembre de 2011**

ÍNDICE

1.- INTRODUCCIÓN.....	1
1.1.- Presentación del proyecto.....	2
1.2.- Motivación del proyecto.....	2
1.3.- Descripción del entorno del proyecto.....	2
1.4.- Objetivos del proyecto.....	3
1.5.- Organización del proyecto.....	3
2.- ESPECIFICACIÓN DE REQUISITOS.....	4
2.1.- Introducción.....	5
2.1.1.- Propósito.....	5
2.1.2.- Ámbito.....	5
2.1.3.- Definiciones, acrónimos y abreviaturas.....	5
2.1.4.- Referencias.....	5
2.1.5.- Visión global.....	5
2.2.- Descripción general.....	6
2.2.1.- Perspectiva del producto.....	6
2.2.2.- Funciones del producto.....	7
2.2.3.- Características del usuario.....	7
2.2.4.- Restricciones generales.....	7
2.2.5.- Suposiciones y dependencias.....	7
2.3.- Requisitos específicos.....	8
2.3.1.- Requisitos de interfaces externos.....	8
2.3.2.- Requisitos funcionales.....	8
2.3.3.- Requisitos de eficiencia.....	9
2.3.4.- Restricciones de diseño.....	9
2.3.5.- Atributos.....	9
3.- ANÁLISIS.....	11
3.1.- La petición.....	12
3.2.- Diagrama de contexto.....	13
3.3.- Plantillas de casos de uso.....	14
3.3.1.- Iniciar sesión.....	14
3.3.2.- Recuperación de usuario y contraseña.....	15
3.3.3.- Listar BeeTrackers.....	15
3.3.4.- Ver información de BeeTracker.....	16
3.3.5.- Cambiar estado de BeeTracker.....	17
3.3.6.- Configurar BeeTracker.....	18
3.3.7.- Deshabilitar BeeTracker.....	19
3.3.8.- Listar alarmas activas.....	20
3.3.9.- Ver alarmas de seguimiento.....	20
3.3.10.- Ver alarmas técnicas.....	21
3.3.11.- Cerrar alarma.....	22
3.3.12.- Listar alarmas cerradas.....	23
3.3.13.- Ver alarmas de seguimiento cerradas.....	24
3.3.14.- Ver alarmas técnicas cerradas.....	24
3.3.15.- Geolocalización BeeTrackers.....	25
3.3.16.- Gestión de usuario.....	26
3.3.17.- Gestión del cliente.....	27
3.3.18.- Creación de incidencias.....	28
3.3.19.- Visualización de informes.....	29
3.3.20.- Salir sesión.....	29

4.- DISEÑO	30
4.1.- Arquitectura de la aplicación.....	31
4.2.- Diseño de la aplicación.....	32
4.2.1.- Diseño de la interfaz de usuario.....	32
4.2.2.- Diseño de la Base de Datos.....	36
4.2.3.- Diseño de la capa de negocio.....	39
5.- IMPLEMENTACIÓN	40
5.1.- Tecnologías utilizadas.....	41
5.1.1.- Navegadores web.....	44
5.2.- Estrategia para el desarrollo de la aplicación.....	44
5.3.- Planificación del proyecto.....	45
5.4.- Archivos creados y desarrollo de la aplicación.....	47
5.4.1.- Diagrama navegacional.....	47
5.4.2.- Estructura de directorios.....	48
5.4.3.- Uso de principales ficheros intrabeetracker.....	53
5.5.- Seguridad de la aplicación.....	69
5.5.1.- Acceso a la aplicación.....	69
5.5.2.- Validación de usuario.....	69
5.5.3.- Permisos de usuario.....	69
5.5.4.- Bloqueo de información.....	70
6.- PRUEBAS	71
6.1.- Pruebas funcionales.....	72
6.1.1.- Seguridad de acceso.....	72
6.1.2.- Error conexión a la Base de Datos.....	74
6.1.3.- Bloqueo de información crítica.....	74
6.1.4.- Acceso a múltiples servidores.....	75
6.1.5.- Alarmas técnicas.....	76
6.1.6.- Alarmas de seguimiento.....	77
6.1.7.- Logs servidores.....	78
6.1.8.- Ayuda en línea.....	78
6.1.9.- Tareas automáticas de generación de informes.....	79
6.1.10.- Proceso completo.....	81
6.2.- Pruebas de validación.....	85
6.2.1.- Validación W3C.....	85
7.- CONCLUSIÓN	86
8.- BIBLIOGRAFÍA	87
8.1.- Páginas Web.....	87
8.2.- Libros.....	87

1.- INTRODUCCIÓN

En este apartado se va a proceder a elaborar una breve introducción del proyecto. Se considerarán la funcionalidades, objetivos ,motivaciones y entorno implicados en el desarrollo.

-
- 1.1.- Presentación del proyecto
 - 1.2.- Motivación del proyecto
 - 1.3.- Descripción del entorno del proyecto
 - 1.4.- Objetivos del proyecto
 - 1.5.- Organización del proyecto
-

1.1.- Presentación del proyecto

BeeTracker es un sistema que permite la monitorización remota de las colmenas con el fin de tener conocimiento detallado de las condiciones en las que se encuentran y optimizar al máximo la producción reduciendo costes de desplazamiento.

El dispositivo se encuentra en el interior de un panal, quedando totalmente camuflado, y gracias a diversos sensores que contiene registra periódicamente parámetros de temperatura, humedad, sonido y peso que son transmitidos al Centro de Proceso de Datos (CPD) para que puedan ser consultados mediante informes detallados de los datos recogidos en cada colmena.

BeeTracker también está dotado de un sistema de localización GPS lo que permite tener localizada su posición en cualquier momento. Esto permite que en caso de sustracción el dispositivo envíe al CPD su posición de forma constante para que la colmena pueda ser posteriormente localizada y recuperada.

La gran novedad que conlleva el BeeTracker es la gran cantidad de datos que provee a los apicultores, muchos de los cuales eran desconocidos, para que así puedan mejorar sus producciones.

Para facilitar el uso de la información anteriormente citada y controlar los BeeTrackers en el día a día o en caso de robo, se hace necesario la creación de una aplicación de gestión accesible para todos desde cualquier lugar, con una interfaz de usuario sencilla y accesible a un público con bajos conocimientos de informática, capaz de sintetizar y condensar la basta información, que sea compatible con la mayoría de navegadores y sistemas del mercado y con una agilidad que permita un acceso cómodo desde una conexión a Internet modesta como la que puede encontrarse en el entorno rural.

Aquí nace el proyecto "Aplicación Web BeeTracker".

1.2.- Motivación del proyecto

El proyecto BeeTracker tiene el gran desafío de introducir en la web la gestión telemática de todos los dispositivos, así como la gestión de la geolocalización en tiempo real de los dispositivos sustraídos, la representación de toda la telemetría capturada y la discriminación de la información en función de los usuarios y privilegios.

En la actualidad, la mayoría de las aplicaciones que salen al mercado se están enfocando hacia el mundo web, por la gran facilidad de acceso, así como a la geolocalización, la necesidad de tener localizado todo. Este proyecto junta estos dos aspectos actualmente tan de moda, lo que hacía que fuera bastante apetecible y difícil de rechazar.

1.3.- Descripción del entorno del proyecto

El dispositivo BeeTracker es el buque insignia de la empresa LEPTON Ingenieros S.L. y se registra como patente mundial.

LEPTON Ingenieros S.L. Es una joven y pequeña empresa que cuenta con 3 empleados dedicada a la gestión remota y la eficiencia energética.

El proyecto BeeTracker surge dentro de un mercado bastante artesanal, como es el de la apicultura, en el que se intenta realizar una mejora en la forma de trabajo que tiene los apicultores para la gestión de los distintos asentamientos de colmenas que tiene en su propiedad. El motivo por el que surge el proyecto, es que los apicultores sufrían robos de los asentamientos y no tenían consciencia de ello hasta que iban a visitarlos, no tenían idea de lo que ocurría en tiempo real, y viendo que no se encontraba nada en el mercado con dichas características aparece BeeTracker.

Debido al carácter industrial y por la seguridad de la aplicación, los pormenores que impliquen una difusión no serán revelados.

1.4.- Objetivos del proyecto

Los objetivos que se quieren obtener con este proyecto es la correcta gestión vía web de todos los BeeTracker. Esta gestión va desde la visualización de los datos de temperatura, humedad, peso y sonido recogidos por cada dispositivo, pasando por la geolocalización en tiempo real de un dispositivo que ha detectado movimiento dentro de la colmena hasta la consulta de la posición global de los distintos dispositivos que están en los asentamientos.

Finalmente, se proporcionarán informes detallados de todas las operaciones, eventos y datos capturados por los dispositivos.

1.5.- Organización del proyecto

La estructura del proyecto a partir de este punto será:

- 2.- Especificación de requisitos, donde expondremos cuales son los requisitos del sistema necesarios para su desarrollo, usando el estándar IEEE 830/98.
- 3.- Análisis, en este apartado realizaremos un análisis completo del sistema, exponiendo cual es la petición del sistema y los casos de uso del mismo.
- 4.- Diseño, en que abordaremos todo lo relativo al diseño de las 3 capas de las que está compuesto nuestro sistema.
- 5.- Detalles de implementación, en que se realizará un estudio sobre cada una de las secciones de las que está compuesta nuestra aplicación.
- 6.- Pruebas, donde expondremos las pruebas que se han realizado para confirmar el correcto funcionamiento de nuestra aplicación.
- 7.- Conclusiones, donde expondré cuales han sido las conclusiones y experiencias obtenidas a partir de la realización del proyecto que a continuación se va a presentar.
- 8.- Bibliografía

2.- ESPECIFICACIÓN DE REQUISITOS

Este apartado se va a realizar la especificación de requisitos. Los contenidos que a continuación se van a exponer, pueden ser tratados como una memoria dentro de este proyecto y están basados en el estándar IEEE 830/98.

-
- 2.1.- Introducción
 - 2.1.1.- Propósito
 - 2.1.2.- Ámbito
 - 2.1.3.- Definiciones, acrónimos y abreviaturas
 - 2.1.4.- Referencias
 - 2.1.5.- Visión global
 - 2.2.- Descripción general
 - 2.2.1.- Perspectiva del producto
 - 2.2.2.- Funciones del producto
 - 2.2.3.- Características del usuario
 - 2.2.4.- Restricciones generales
 - 2.2.5.- Supuestos y dependencias
 - 2.3.- Requisitos específicos
 - 2.3.1.- Requisitos de interfaces externos
 - 2.3.2.- Requisitos funcionales
 - 2.3.3.- Requisitos de eficiencia
 - 2.3.4.- Restricciones de diseño
 - 2.3.5.- Atributos
-

2.1.- Introducción

2.1.1.- Propósito

El propósito de la Especificación de Requisitos del Software de este documento, es la de presentar de forma detallada el proyecto BeeTracker, desde el propio dispositivo (BeeTracker) hasta de su sistema de gestión mediante una aplicación web, explicando cual es su propósito y las características, así como las interfaces del sistema.

Este documento va dirigido a las distintas partes interesadas, a los desarrolladores del sistema y a los miembros de la empresa LEPTON Ingenieros S.L..

2.1.2.- Ámbito

La Aplicación Web BeeTracker es un sistema de gestión remota de un dispositivo, denominado BeeTracker, destinada a apicultores con el que se permite tener un control remoto del estado de una colmena sin tener que desplazarse hasta a ella, gracias al constante flujo de datos que son transmitidos por el dispositivos desde el interior de la colmena.

Las grandes ventajas que trae consigo este dispositivo para los apicultores, es que reciben y registran datos hasta ahora casi desconocidos de temperaturas, humedades, peso y sonido de dentro de la colmena, así como la posibilidad de rastrearlo en caso de que se produzca una sustracción de la colmena.

Las metas a las que se quieren llegar con el proyecto es obtener una clara y completa presentación de la información además de ofrecer una interfaz de usuario sencilla de utilizar.

2.1.3.- Definiciones, acrónimos y abreviaturas

BeeTracker	BT
Aplicación Web BeeTracker	AWBT
Especificación de Requisitos del Software	ERS
Centro de Proceso de Datos	CPD
Data Slot	DS

2.1.4.- Referencias

Las referencias que se realizan dentro de este documento son:

[1] IEEE. *IEEE Std. 830-1998 IEEE práctica recomendada para los requisitos de software Especificaciones*. IEEE Computer Society, 1998.

2.1.5.- Visión global

A partir de este punto dentro de la ERS vamos a pasar a analizar:

2.- Descripción general

En este apartado se pasará a valorar todos aquellos factores que afectan a la Aplicación Web BeeTracker y a sus requisitos.

3.- Requisitos específicos

En este apartado se desarrollaran todos los requisitos de la Aplicación Web BeeTracker para que sea posible el diseño del producto.

2.2.- Descripción general

2.2.1.- Perspectiva del producto

AWBT es una aplicación web que permite la gestión remota y la monitorización de los datos que son transmitidos por un dispositivo, por lo que únicamente es una pequeña parte dentro de todo el proyecto BeeTracker.

En el siguiente diagrama observamos en su conjunto el proyecto BeeTracker:

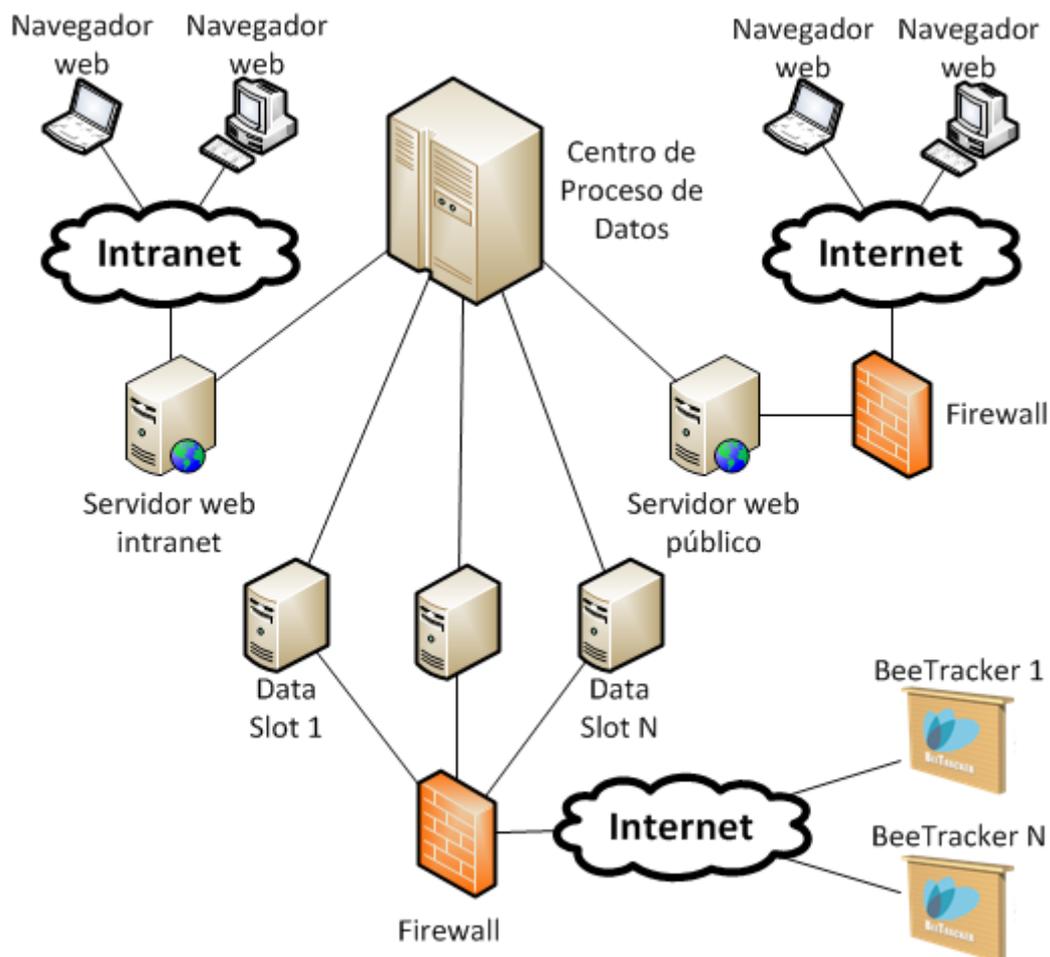


Diagrama estructura sistema BeeTracker

Como acabamos de ver, el proyecto BeeTracker consta de un Centro de Proceso de Datos (CPD) encargado de la gestión de todo el sistema. Desde él, se controlan los dispositivos y las aplicaciones web públicas y las privadas.

AWBT es la herramienta que permite la gestión del BT, que es un dispositivo que se encuentra en el interior de una colmena de abejas y permite recoger datos de temperatura, humedad, sonido y peso. El BT funciona de forma autónoma, ya que cada hora va registrando todos estos datos y cada cierto tiempo, como le sea establecido en su configuración, transmite los datos para que estos puedan ser consultados a posterior. El hecho de ser un dispositivo autónomo provoca que debe de tener una gran eficiencia energética para que su duración sea la correcta, esto hace que las comunicaciones siempre las inicie el propio dispositivo, así el resto del tiempo puede estar en modo hibernación.

A parte de estos sensores anteriormente descritos, el dispositivo también contiene un sensor de movimiento que permite controlar el más ligero movimiento que se produce en el exterior de la colmena enviando, desde el momento que es detectado dicho movimiento, una trama de datos indicando la posición actual del dispositivo, bien por posición GPS (mediante latitud o longitud) o bien mediante coberturas móviles, lo que nos permite en caso de no tener una posición GPS dar una posición aproximada de donde se encuentra el dispositivo.

Un dispositivo BeeTracker se comunica con el CPD mediante un Data Slot, estos servidores se van a encargar de procesar y almacenar toda la información que los BT transmiten.

Por último, nos encontramos con la aplicación web de la Intranet, con la cual los administradores del sistema pueden controlar todo el sistema del proyecto BeeTracker.

2.2.2.- Funciones del producto

La AWBT permitirá la gestión de los distintos BT que un cliente tenga contratados, así como la posibilidad de visionar los distintos parámetros que han sido recogidos por el dispositivo.

También podrá realizar consultas sobre incidencias que se hayan producido por el dispositivo y consultar todas aquellas que ya hayan tenido lugar.

Por último, tendrá un sistema de comunicación con los administradores del sistema para poderles plasmar problemas o dudas que le surjan con la aplicación o con el dispositivo.

2.2.3.- Características del usuario

Los usuarios a los que va destinado la utilización de la AWBT son los apicultores, por lo que el nivel de complejidad de la aplicación no tiene que ser muy alto y todas las tareas y acciones que se realicen deben ser muy intuitivas.

2.2.4.- Restricciones generales

El desarrollo de la aplicación se realizara con Aptana y en algunos momentos IDE NetBeans .

Los lenguajes de programación para la AWBT son: PHP, HTML, XAJAX, SQL y JavaScript.

También es necesario el lenguaje de programación Java para el desarrollo de las aplicaciones de gestión de informes.

2.2.5.- Suposiciones y dependencias

La AWBT deberá ser compatible con los exploradores ofertados en el mercado, como Internet Explorer, Mozilla Firefox, Google Chrome y Safari, aunque las pruebas se realizarán sobre Mozilla Firefox 4.

2.3.- Requisitos específicos

2.3.1.- Requisitos de interfaces externas

No tiene requisitos de interfaces externas.

2.3.2.- Requisitos funcionales

Las principales funciones que debe realizar la AWBT son:

- Inicio de sesión: El usuario para poder acceder al sistema debe de introducir un usuario y contraseña correcto para poder dar los privilegios necesarios para acceder al sistema. Solo un usuario con permisos sobre un grupo de asentamiento o un asentamiento será redirigido a nuestra página.

En caso de que el usuario no recuerde su usuario o contraseña, deberá introducir un correo electrónico que se encuentra en sus datos personales, donde se le enviara el usuario con una nueva contraseña para poder acceder de nuevo al sistema.

- Listado de los dispositivos BT de un cliente. Aquí podrá visualizar los dispositivos que se encuentren activos en el sistema y comprobar cual es su estado (encendido, apagado, en alarma,...), cual es su posición o ver más detalles los datos de un dispositivo, cada usuario sólo podrá visualizar los dispositivos a los que tenga acceso acorde a los privilegios obtenidos en el inicio de sesión.

Debe tener la posibilidad de realizar un filtrado para la búsqueda concreta de un dispositivo ya sea por su nombre, estado, tipo, BT o asentamiento en el que se encuentra el dispositivo

- Gestión de estado BT. En cualquier fase en que se encuentre el dispositivo podremos comprobar cual es su estado (encendido, apagado, en alarma, actualizando, iniciando, apagándose,...).

En el caso de que el dispositivo se encuentre en estado Encendido y que el usuario tenga los privilegios suficientes se podrá apagar el dispositivo de forma remota.

- Gestión de avisos de un BT. Este es un punto únicamente accesible si el usuario que está registrado en el sistema tiene los permisos necesarios, en este caso podrá configurar los avisos de alarmas que tiene el dispositivo (temperatura, humedad, peso, sonido, sensibilidad de movimiento, estado de la batería) contemplando si el tipo es básico o premium, así como también podrá deshabilitar los sensores de alarmas durante un periodo de 24 horas.

- Visión de los parámetros recogidos por el BT. En forma de gráfica en el que se podrán contemplar los datos de temperatura, humedad, peso, sonido o batería, según el tipo de dispositivo que sea, de las últimas 24 horas, de la última semana, del último mes y del último año. Estos datos podrán ser impresos o guardados en diferentes formatos para su posterior gestión por parte del usuario.

- Gestión de alarmas activas. Se visualizarán las alarmas que el dispositivo haya detectado debido al haber sobrepasado alguno de los parámetros que tiene en su configuración. Aquí se distinguirán 2 tipos de alarmas: las técnicas y las de seguimiento.

- Gestión de alarmas técnicas. Estas alarmas son las producidas por la temperatura, humedad, sonido, peso o batería. Estas incidencias serán tratadas como meras transportadoras de información para el usuario mostrando una gráfica para que se vea la evolución de las últimas 48 horas en las que se ha producido dicha alarma.
- Gestión de alarmas de seguimiento por robo. Estas alarmas se activan por la detección de un movimiento, lo que puede conllevar una sustracción de la colmena, por lo que es necesario la posibilidad de poder realizar un seguimiento en tiempo real, marcando la posición en la que se encuentra en todo momento el dispositivo gracias a los datos que son transmitidos por el BT.
- Gestión de alarmas históricas. Una vez se considere que una alarma puede ser cerrada, estas pasarán a un histórico para que puedan ser consultadas con posterioridad ofreciendo todos los datos que se hayan recogido durante las mismas.

El usuario con los suficientes permisos, podrá eliminar incidencias que ya no considere necesarias y tendrá una posibilidad de filtrado para poder localizar incidencias de manera más sencilla.

- Geolocalización de los BT. Todos los dispositivos deben marcar una posición en cualquier estado en que se encuentren.
- Gestión de informes. Se debe poder visionar cualquier tipo de informe que se haya producido, ya sean por incidencias de robo, informes mensuales teledados capturados por el BT, facturas emitidas, informes de registro/acciones por usuario o la visualización de dispositivos dados de baja por los administradores del sistema.
- Gestión de usuarios/cliente. El usuario podrá personalizar sus datos personales, de contacto y actualizar su contraseña.
Si tuviera los permisos necesarios, podrá visualizar el listado de usuarios que existen para dicho cliente y actualizar los datos de contacto del cliente.
- Comunicación con los administradores. Se permitirá a los usuarios la posibilidad de comunicarse de manera privada por medio de un foro, en el cual podrán exponer sus dudas o sugerencias tanto de la AWBT como de los dispositivos BT.

2.3.3.- Requisitos de eficiencia

El sistema deberá permitir el acceso a la misma información por varios clientes o administradores simultáneamente, sin que la información que contiene se vea afectado por las acciones de alguno de los usuarios.

2.3.4.- Restricciones de diseño

El sistema tendrá que ser validado por los estándares de la W3C para el código generado HTML como para las hojas de estilos CSS y compatible con todos los navegadores actuales del mercado.

2.3.5.- Atributos

El sistema únicamente permite el acceso a personas autorizadas por los administradores, por lo que el registro se realizara mediante usuario y contraseña.

Toda acción que se realice sobre el sistema por los distintos usuarios de la aplicación quedará registrada en un log , para tener constancia de las acciones que se realizan.

Como medida de seguridad, la página tendrá una caducidad si sobrepasa el tiempo de navegación previamente establecido, por lo que un usuario que lo sobrepase deberá iniciar de nuevo sesión en la aplicación web.

El sistema puede estar compuesto por más de un servidor, por lo que la AWBT debe estar preparada para ello.

3.- ANÁLISIS

En este apartado vamos a pasar a realizar el análisis de la aplicación Web BeeTracker, desarrollando la petición para el desarrollo del proyecto y mostrando todos los casos de uso que nos hemos encontrado en el sistema.

-
- 3.1.- La petición
 - 3.2.- Diagrama de contexto
 - 3.3.- Plantillas de casos de uso
 - 3.3.1.- Iniciar sesión
 - 3.3.2.- Recuperar usuario y contraseña
 - 3.3.3.- Listar BeeTrackers
 - 3.3.4.- Mostrar información de BeeTracker
 - 3.3.5.- Cambiar estado de BeeTracker
 - 3.3.5.1.- Diagrama de secuencia
 - 3.3.6.- Configurar BeeTracker
 - 3.3.7.- Deshabilitar BeeTracker
 - 3.3.8.1.- Diagrama de secuencia
 - 3.3.8.- Listar alarmas activas
 - 3.3.9.- Ver alarmas de seguimiento
 - 3.3.10.- Ver alarmas técnicas
 - 3.3.11.- Cerrar alarma
 - 3.3.11.1.- Diagrama de secuencia
 - 3.3.12.- Listar alarmas cerradas
 - 3.3.13.- Ver alarmas de seguimiento cerradas
 - 3.3.14.- Ver alarmas técnicas cerradas
 - 3.3.15.- Geolocalización BeeTrackers
 - 3.3.16.- Gestión de usuario
 - 3.3.17.- Gestión del cliente
 - 3.3.18.- Creación de incidencias
 - 3.3.19.- Visualización de informes
 - 3.3.20.- Salir sesión
-

3.1.- La petición

La empresa LEPTON Ingenieros S.L. necesita desarrollar una aplicación web para el control del dispositivo BeeTracker.

El sistema debe ser capaz de permitir que usuarios registrados accedan al sistema, otorgándoles los privilegios acordes a su perfil, para así realizar el control de los distintos dispositivos que pertenecen al cliente.

El usuario podrá observar de un vistazo el estado de todos sus dispositivos, cual es su localización y el tipo de dispositivos que son y ver con más detalles cada uno de estos dispositivos. Aquí se podrá visionar todos los datos transmitidos por el BeeTracker desde la fecha actual hasta un año hacia atrás y según los permisos del usuario podrá actualizar la configuración del dispositivo para la gestión de las alarmas que se puedan producir, deshabilitar el dispositivo en caso de que el cliente lo desee o incluso si se encuentra en estado encendido podrá desconectarlo de forma remota.

Todos los dispositivos podrán lanzar dos tipos de alarmas: técnicas o de seguimiento. Estas alarmas se encontrarán en estado activo hasta que un usuario con los privilegios necesario realice su cierre. De las alarmas técnicas, nos interesa saber cual es el motivo de dicha alarma mostrando toda la información necesaria para explicar el motivo de dicha alarma técnica, mientras que en el caso de las alarmas de seguimiento lo importante es poder realizar un seguimiento a partir de los datos que el dispositivo nos ha ido transmitiendo desde el inicio de la misma, permitiendo trazar una ruta por la que se está desplazando el dispositivo en tiempo real.

Una vez las alarmas son cerradas por el usuario privilegiado, que son los usuarios supervisores o cualquier usuario con permiso de escritura sobre el dispositivo, estas pasan a un histórico donde el usuario puede comprobar todas estas alarmas ya cerradas. Como en el caso anterior la información es mostrada según el tipo de alarma, lo distinto es que las alarmas de seguimiento muestran todos los puntos recibidos en dicha alarma.

El usuario también tendrá la posibilidad de localizar sus dispositivos en cualquier momento gracias a la información que los BeeTrackers van enviando diariamente.

También se podrá realizar la gestión de datos de usuario, donde cada usuario podrá modificar sus datos personales, siempre y cuando el recurso no este siendo utilizado por algún administrador para así mantener siempre los datos de forma correcta. Del mismo modo, y teniendo los privilegios necesarios, se podrán modificar los datos de contacto relativos al cliente con las mismas características que con el usuario.

Para una mayor comunicación con los administradores, el sistema tiene que permitir crear incidencias donde el usuario pueda exponer sus dudas o problemas con el dispositivo para que los administradores les faciliten una solución, esto se tratara como un pequeño foro privado con los administradores, en donde cualquier usuario del cliente pueda exponer sus consultas.

Por último, los usuarios con privilegios tendrán acceso a una serie de informes relativos a las alarmas de seguimiento cerradas, los informes de telemidas mensuales, el registro de usuario mensual del cliente y así como las diferentes facturas emitidas.

3.2.- Diagrama de contexto

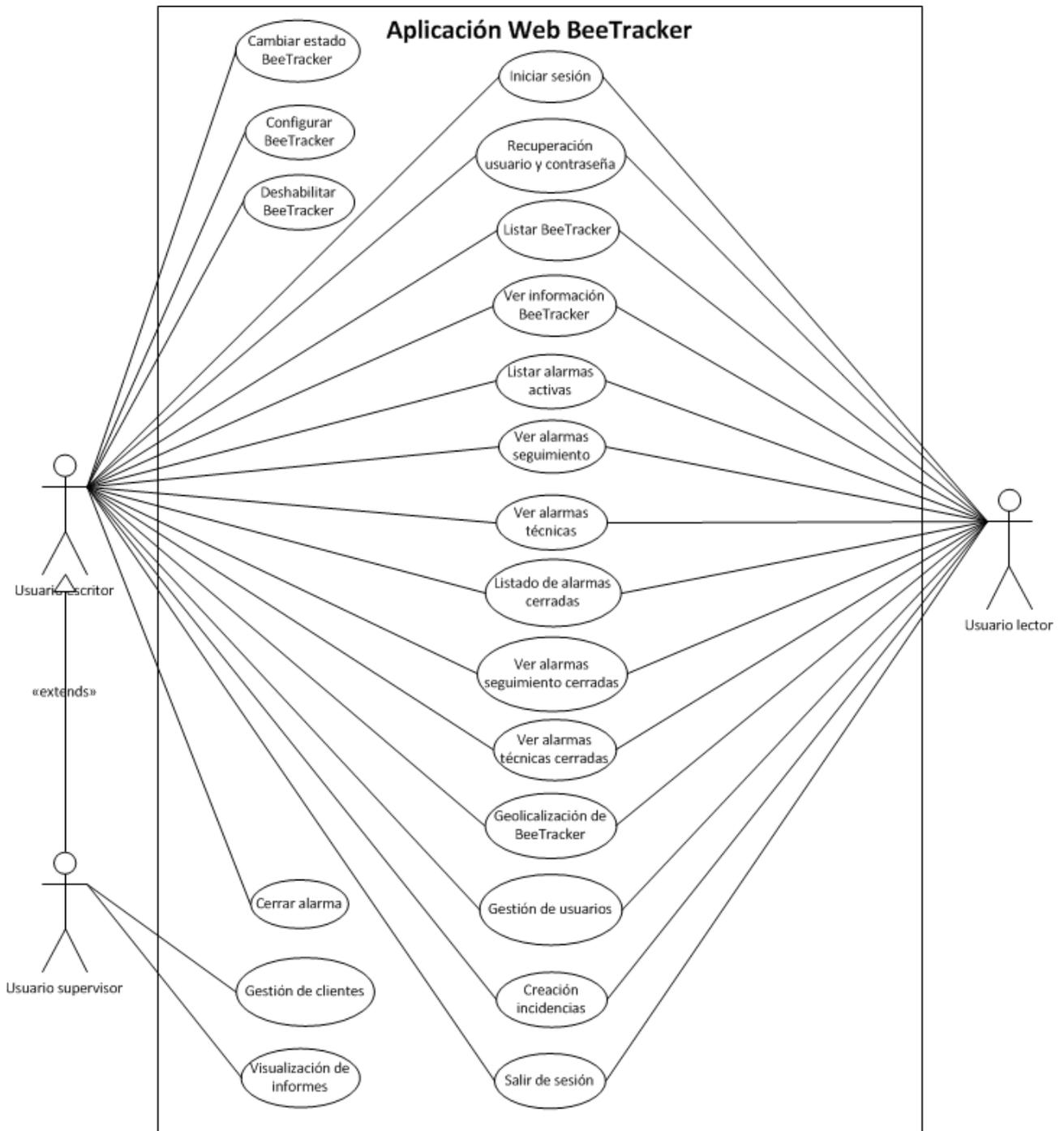


Diagrama de casos de uso

3.3.- Plantillas de casos de uso

3.3.1.- Iniciar sesión

Este caso de uso contiene un caso de uso normal y cuatro casos alternativos que se describen a continuación.

Caso de uso: Iniciar sesión

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Controla el acceso a la aplicación de los usuarios, se encarga de comprobar que el usuario existe en la Base de Datos y le entrega los privilegios asociados a dicha cuenta.

Requerimientos: Ninguno

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario entra en la página de inicio de la aplicación web, entra los datos de cuenta y contraseña e ingresa en la aplicación.	
	2.- El sistema comprueba que el usuario existe y le otorga los derechos de acceso.
	3.- Finalizada la entrada de datos del usuario y la comprobación del sistema, muestra el escritorio de la aplicación.

Curso alternativo de acontecimientos. Error en la entrada de datos:

2.- El sistema comprueba que los datos introducidos son incorrectos y lo devuelve a la pagina de acceso de la aplicación mostrando un mensaje acorde al error. Al cuarto acceso se bloquea la cuenta y es reportado a los administradores.

Curso alternativo de acontecimientos. Usuario bloqueado por el administrador:

2.- El sistema comprueba que los datos introducidos son correctos, pero su estado se encuentra en un estado no activo para el sistema y lo devuelve a la pagina de acceso a la aplicación mostrando un mensaje acorde al error.

Curso alternativo de acontecimientos. Actualización de contraseña:

3.- Finalizada la entrada de datos de usuario y comprobación del sistema, y muestra la página de actualización de contraseña, donde el usuario puede cambiar la contraseña.

Curso alternativo de acontecimientos. Usuario activo pero sin permisos:

3.- Finalizada la entrada de datos de usuario y comprobación del sistema, se muestra una pantalla de contacto con los administradores ya que el usuario no tiene asignado ningún tipo de permisos en el sistema.

3.3.2.- Recuperación de usuario y contraseña

Este caso de uso contiene un caso de uso normal y dos casos alternativos que se describen a continuación.

Caso de uso: Recuperación de usuario y contraseña

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Permite al usuario recuperar su nombre de usuario y contraseña en caso de no recordarlo, enviándole a la cuenta de correo introducido el usuario y una nueva contraseña.

Requerimientos: Ninguno

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario entra en la página de inicio de la aplicación web, selecciona recuperación de contraseña e introduce el correo electrónico y solicita su recuperación.	
	2.- El sistema comprueba que el correo existe en la Base de Datos y envía a la cuenta facilitada la información de usuario asociada a dicho correo. Le envía a la página de inicio de la aplicación donde le muestra un mensaje para que compruebe el correo introducido.

Curso alternativo de acontecimientos. Error en la entrada de datos:

2.- El sistema comprueba que los datos introducidos son incorrectos y lo devuelve a la página de introducción de correo electrónico mostrando un mensaje acorde al error.

Curso alternativo de acontecimientos. Usuario bloqueado por el administrador:

2.- El sistema comprueba que los datos introducidos son correctos, pero su estado se encuentra en un estado no activo para el sistema y lo devuelve a la página de introducción de correo electrónico mostrando un mensaje acorde al error.

3.3.3.- Listar BeeTrackers

Este caso de uso contiene un caso de uso normal y un caso alternativo que se describen a continuación.

Caso de uso: Listar BeeTrackers

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Se muestran todos los dispositivos BeeTracker activos asociados al cliente del usuario registrado

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita el listado de BeeTrackers del cliente	
	2.- El sistema comprueba que dispositivos están activos en el sistema y se corresponden con el cliente del usuario introducido y son mostrados, acorde a los privilegios que el usuario ha adquirido al acceder al sistema.

Curso alternativo de acontecimientos. Búsqueda de dispositivos:

2.- El sistema comprueba que dispositivos activos concuerdan con los datos de búsqueda introducidos por el usuario y son mostrados. En caso de que no exista ninguno que cumpla con esas condiciones muestra un mensaje de información.

3.3.4.- Ver información de BeeTracker

Este caso de uso contiene un caso de uso normal y tres casos alternativos que se describen a continuación.

Caso de uso: Ver información de un BeeTracker

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Se muestran todos los dispositivos BeeTracker activos asociados al cliente del usuario registrado

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita ver la información de un BeeTracker.	
	2.- El sistema comprueba que el dispositivo solicitado corresponde al cliente del usuario introducido y son mostrados, siempre y cuando el usuario tenga acceso a dicho dispositivo

Curso alternativo de acontecimientos. Dispositivo no perteneciente al cliente del usuario:

2.- El sistema comprueba que dispositivos no corresponde al cliente del usuario, el sistema lo devuelve al listado de BeeTrackers del usuario.

Curso alternativo de acontecimientos. Dispositivo no activo en el sistema:

2.- El sistema comprueba que dispositivos no está activo en el sistema y lo devuelve al listado de BeeTrackers del usuario.

Curso alternativo de acontecimientos. Dispositivo no accesible por el usuario:

2.- El sistema comprueba que dispositivos no es accesible con los permisos del usuario y lo devuelve al listado de BeeTrackers del usuario.

3.3.5.- Cambiar estado de BeeTracker

Este caso de uso contiene un caso de uso normal y dos casos alternativos que se describen a continuación.

Caso de uso: Cambiar estado de un BeeTracker

Actores: Usuario supervisor y usuario con permisos de escritura.

Descripción: El usuario puede cambiar el estado a apagado del dispositivo cuando se encuentre encendido de forma remota sin tener que desplazarse hasta el dispositivo.

Requerimientos: Haber iniciado sesión. Estado BeeTracker encendido.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita desconectar un BeeTracker en particular siempre que se encuentre en estado encendido.	
	2.- El sistema solicita confirmación de la operación
3.- El usuario confirma la operación de apagado	
	4.- El sistema autentifica que el dispositivo pertenece al cliente del usuario, y si tiene los privilegios necesarios lo desconecta

Curso alternativo de acontecimientos. Cancelación petición de apagado:

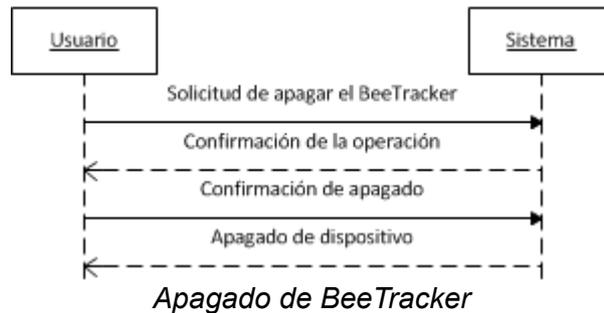
3.- El usuario cancela la petición de apagado del dispositivo.

Curso alternativo de acontecimientos. El usuario no tiene suficientes privilegios:

4.- El sistema comprueba que dispositivos pertenece al cliente del usuario pero no tiene suficientes privilegios para poder desconectarlo, así que cancela la operación.

3.3.5.1.- Diagrama de secuencia

A continuación se muestra el diagrama de secuencia para el caso de uso normal del cambio de estado de un BeeTracker.



3.3.6.- Configurar BeeTracker

Este caso de uso contiene un caso de uso normal y dos casos alternativos que se describen a continuación.

Caso de uso: Configurar un BeeTracker

Actores: Usuario supervisor y usuario con permisos de escritura..

Descripción: El usuario puede cambiar la configuración de los parámetros de temperatura, humedad, peso, sonido, sensibilidad de movimiento y estado de la batería de un dispositivo.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario configura los distintos parámetros de un dispositivo BeeTracker	
	2.- El sistema valida los parámetros introducidos.
	3.- El sistema comprueba que el usuario tiene los privilegios correctos, actualiza los datos y confirma que los cambios se han realizado con éxito.

Curso alternativo de acontecimientos. El usuario introduce los datos de forma incorrecta:

2.- El sistema valida que los parámetros introducidos son incorrectos y muestra un mensaje informando de ello al usuario.

Curso alternativo de acontecimientos. El usuario no tiene suficientes privilegios:

3.- El sistema comprueba que dispositivos pertenece al cliente del usuario pero no tiene suficientes privilegios para poder desconectarlo, así que cancela la operación.

3.3.7.- Deshabilitar BeeTracker

Este caso de uso contiene un caso de uso normal y cuatro casos alternativos que se describen a continuación.

Caso de uso: Deshabilitar un BeeTracker

Actores: Usuario supervisor y usuario con permisos de escritura.

Descripción: El usuario puede configurar el dispositivo para que en un periodo de 24 horas no reaccione ante ninguna alarma que se pueda detectar.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario configura un BeeTracker para que quede deshabilitado en un periodo máximo de 24 horas.	
	2.- El sistema valida los parámetros introducidos.
	3.- El sistema solicita confirmación para las deshabilitación.
4.- El usuario confirma la deshabilitación del dispositivo BeeTracker	
	5.- El sistema comprueba que el usuario tiene los privilegios correctos, actualiza los datos y confirma que los cambios se han realizado con éxito.

Curso alternativo de acontecimientos. El usuario cancela una deshabilitación existente:

1.- El usuario cancela un periodo de deshabilitación previamente introducido.

2.- El sistema comprueba que el usuario tiene los privilegios correctos, actualiza los datos y confirma que los cambios se han realizado con éxito.

Curso alternativo de acontecimientos. El usuario introduce los datos de forma incorrecta:

2.- El sistema valida que los parámetros introducidos son incorrectos y muestra un mensaje informando de ello al usuario.

Curso alternativo de acontecimientos. El usuario cancela la deshabilitación:

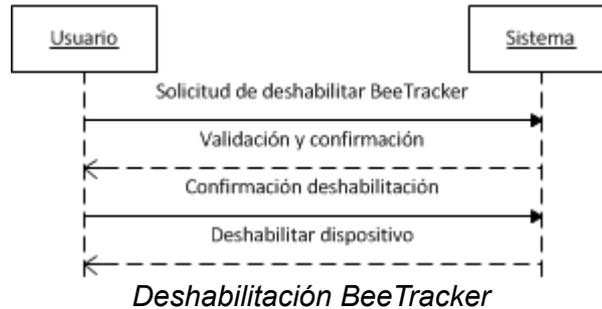
4.- El usuario cancela la deshabilitación.

Curso alternativo de acontecimientos. El usuario no tiene suficientes privilegios:

5.- El sistema comprueba que dispositivos pertenece al cliente del usuario pero no tiene suficientes privilegios para poder desconectarlo, así que cancela la operación.

3.3.7.1.- Diagrama de secuencia

A continuación se muestra el diagrama de secuencia para el caso de uso normal de la deshabilitación de un BeeTracker.



3.3.8.- Listar alarmas activas

Este caso de uso contiene un caso de uso normal que se describe a continuación.

Caso de uso: Listar alarmas activas.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Se muestra todas las alarmas activas que hayan producido los dispositivos del cliente.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita el listado de alarmas activas del usuario	
	2.- El sistema comprueba que alarmas se encuentran activas se corresponden con el cliente del usuario introducido y son mostrados, acorde a los privilegios que el usuario ha adquirido al acceder al sistema

3.3.9.- Ver alarmas de seguimiento

Este caso de uso contiene un caso de uso normal y un caso alternativo que se describen a continuación.

Caso de uso: Ver alarmas de seguimiento.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: El usuario puede ver los detalles de las alarmas de seguimiento pudiendo ver en tiempo real cual es la posición en la que se encuentra durante el periodo de tiempo que el dispositivo se encuentre en alarma.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la información de una alarma de seguimiento	
	2.- El sistema muestra la información del último punto transmitido desde el dispositivo
	3.- El sistema recupera información en forma de gráfica de todos los puntos recuperados hasta el momento.
4.- El usuario solicita ampliar la información de la alarma que se está realizando el seguimiento	
	5.- El sistema muestra toda la información transmitida por el dispositivo desde que inicio su estado de alarma.

Curso alternativo de acontecimientos. El usuario no tiene privilegios suficientes:

2.- El sistema comprueba que el usuario que solicita la acción tiene permisos de para poder visualizar la información solicitada, si no fuera así, se le devolvería al listado de alarmas.

3.3.10.- Ver alarmas técnicas

Este caso de uso contiene un caso de uso normal y dos caso alternativo que se describen a continuación.

Caso de uso: Ver alarmas técnicas.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: El usuario puede ver los detalles de las alarmas técnica según el tipo de la misma.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la información de una alarma de técnica	
	2.- El sistema comprueba si el tipo dispositivo que realizo la alarma técnica y recupera los datos que ocasionaron la alarma.

3.- El sistema recupera información relativa del dispositivo para comprobar de forma más sencilla el porque se ha producido la alarma.

Curso alternativo de acontecimientos. El usuario no tiene privilegios suficientes:

2.- El sistema comprueba que el usuario que solicita la acción tiene permisos de para poder visualizar la información solicitada, si no fuera así, se le devolvería al listado de alarmas.

Curso alternativo de acontecimientos. Tipo de dispositivo:

2.- El sistema comprueba si el tipo dispositivo que realizo la alarma técnica y si el tipo de alarma es superior al tipo de dispositivo muestra un mensaje de información de que no se puede visionar la alarma.

3.3.11.- Cerrar alarma

Este caso de uso contiene un caso de uso normal y dos casos alternativos que se describen a continuación.

Caso de uso: Cerrar alarma.

Actores: Usuario supervisor y usuario con permisos de escritura.

Descripción: Cuando el usuario considere que la alarma ha finalizado, puede realizar el cierre para que sea almacenada como un alarma histórica.

Requerimientos: Haber iniciado sesión. Estado BeeTracker encendido.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita el cierre de una alarma.	
	2.- El sistema comprueba que la alarma todavía este activa.
	3.- El sistema solicita confirmación para realizar el cierre de la alarma
4.- El usuario confirma que desea realizar el cierre de la alarma.	
	5.- El sistema comprueba que el usuario tiene los privilegios correctos, modifica los datos para indicar que la alarma se ha cerrado y se lo confirma al usuario.

Curso alternativo de acontecimientos. El usuario cancela el cierre de la alarma:

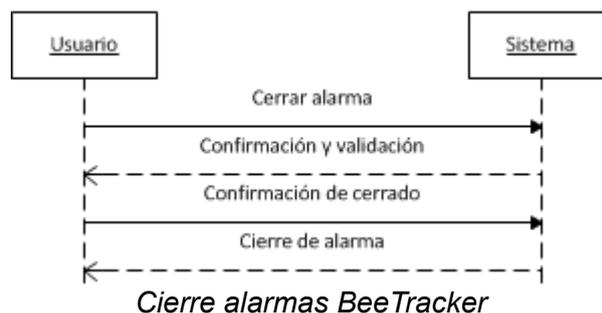
4.- El usuario cancela una petición de cierre de alarma.

Curso alternativo de acontecimientos. El usuario no tiene suficientes privilegios:

5.- El sistema comprueba el usuario pero no tiene suficientes privilegios para poder cerrar incidencias, así que cancela la operación.

3.3.11.1.- Diagrama de secuencia

A continuación se muestra el diagrama de secuencia para el caso de uso normal de la cierre de alarmas de un BeeTracker.

**3.3.12.- Listar alarmas cerradas**

Este caso de uso contiene un caso de uso normal que se describe a continuación.

Caso de uso: Listar alarmas cerradas.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Se muestran todas las incidencias cerradas de los dispositivos del cliente

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita el listado de alarmas cerradas del usuario	
	2.- El sistema comprueba que alarmas se encuentran cerradas se corresponden con el cliente del usuario introducido y son mostrados.

3.3.13.- Ver alarmas de seguimiento cerradas

Este caso de uso contiene un caso de uso normal se describe a continuación.

Caso de uso: Listar alarmas de seguimiento cerradas.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: El usuario puede ver los detalles de las alarmas de seguimiento cerrada pudiendo ver todo el recorrido y los datos recibidos desde el dispositivo.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la información de una alarma de seguimiento cerrada	
	2.- El sistema muestra la información del último punto transmitido desde el dispositivo
	3.- El sistema recupera información en forma de gráfica de todos los puntos recuperados durante el seguimiento.
4.- El usuario solicita ampliar la información de la alarma que se realizó el seguimiento	
	5.- El sistema muestra toda la información que el dispositivo envió durante el periodo de tiempo que estaba en alarma.

Curso alternativo de acontecimientos. El usuario no tiene privilegios suficientes:

2.- El sistema comprueba que el usuario que solicita la acción tiene permisos de para poder visualizar la información solicitada, si no fuera así, se le devolvería al listado de alarmas.

3.3.14.- Ver alarmas técnicas cerradas

Este caso de uso contiene un caso de uso normal se describe a continuación.

Caso de uso: Listar alarmas técnicas cerradas.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: El usuario puede ver los detalles de las alarmas técnica cerradas según el tipo de la misma.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la información de una alarma de técnica cerrada	
	2.- El sistema comprueba si el tipo dispositivo que realizo la alarma técnica y recupera los datos que ocasionaron la alarma.
	3.- El sistema recupera información relativa del dispositivo para comprobar de forma más sencilla el porque se produjo la alarma.

Curso alternativo de acontecimientos. El usuario no tiene privilegios suficientes:

2.- El sistema comprueba que el usuario que solicita la acción tiene permisos de para poder visualizar la información solicitada, si no fuera así, se le devolvería al listado de alarmas.

Curso alternativo de acontecimientos. Tipo de dispositivo:

2.- El sistema comprueba si el tipo dispositivo que realizo la alarma técnica y si el tipo de alarma es superior al tipo del dispositivo muestra un mensaje de información de que no se puede visionar la alarma.

3.3.15.- Geolocalización BeeTrackers

Este caso de uso contiene un caso de uso normal se describe a continuación.

Caso de uso: Geolocalización de BeeTrackers.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Se visualizará la posición de los dispositivos localizados del cliente del usuario.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la localización de todos los dispositivos que pertenecen a su cliente.	
	2.- El sistema muestra la información relativa a la posición de todos los dispositivos del cliente, a los cuales el usuario que solicita la acción tiene acceso.

3.3.16.- Gestión de usuario

Este caso de uso contiene un caso de uso normal y tres casos alternativos que se describen a continuación.

Caso de uso: Gestión de usuario.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: El usuario puede personalizar sus datos de usuario, así como actualizar su contraseña.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la modificación de sus datos personales de usuario.	
	2.- El sistema comprueba que los datos no se encuentren bloqueados por algún motivo.
	3. El sistema facilita al usuario la posibilidad de cambiar los datos
4.- El usuario modifica los datos y confirma el cambio	
	5.- El sistema valida los datos introducidos
	6.- El sistema vuelve a comprobar que los datos del usuario no se encuentren bloqueados para poder realizar el cambio
	7.- El sistema realiza los cambios y confirma el cambio.

Curso alternativo de acontecimientos. Datos bloqueados, en 2 casos:

2.- El sistema comprueba que los datos no se encuentren bloqueados por el sistema, si se encuentran muestran un mensaje acorde al usuario.

6.- El sistema comprueba que los datos no se encuentren bloqueados por el sistema, si se encuentran muestran un mensaje acorde al usuario sin realizar los cambios.

Curso alternativo de acontecimientos. El usuario cancela los cambios:

4.- El usuario cancela la modificación de los datos.

5.- El sistema restaura los valores por defecto de la información de usuario.

Curso alternativo de acontecimientos. Los datos introducidos son incorrectos:

5.- El sistema comprueba que los datos introducidos no son correctos y muestra un mensaje acorde al error al usuario.

3.3.17.- Gestión del cliente

Este caso de uso contiene un caso de uso normal y cuatro casos alternativos que se describen a continuación.

Caso de uso: Gestión del cliente.

Actores: Usuario supervisor.

Descripción: El usuario puede visualizar y modificar los datos del cliente, así como ver el listado de usuarios del cliente.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita la modificación de sus datos del cliente.	
	2.- El sistema comprueba que el usuario tiene permisos para realizar el cambio.
	3.- El sistema comprueba que los datos no se encuentre bloqueados por algún motivo.
	4.- El sistema facilita al usuario la posibilidad de cambiar los datos
5.- El usuario modifica los datos y confirma el cambio	
	6.- El sistema valida los datos introducidos
	7.- El sistema comprueba que el usuario tiene permisos para realizar el cambio.
	8.- El sistema vuelve a comprobar que los datos del usuario no se encuentre bloqueados para poder realizar el cambio
	9.- El sistema realiza los cambios y confirma el cambio.

Curso alternativo de acontecimientos. Usuario sin privilegios, en 2 casos:

2.- El sistema comprueba que el usuario no tiene privilegios para modificar los datos del cliente y cancela la operación mostrando un mensaje acorde.

7.- El sistema comprueba que el usuario no tiene privilegios para modificar los datos del cliente y cancela la operación mostrando un mensaje acorde.

Curso alternativo de acontecimientos. Datos bloqueados, en 2 casos:

3.- El sistema comprueba que los datos no se encuentren bloqueados por el sistema, si se encuentran muestran un mensaje acorde al usuario.

8.- El sistema comprueba que los datos no se encuentren bloqueados por el sistema, si se encuentran muestran un mensaje acorde al usuario sin realizar los cambios.

Curso alternativo de acontecimientos. El usuario cancela los cambios:

5.- El usuario cancela la modificación de los datos.

6.- El sistema restaura los valores por defecto de la información de usuario.

Curso alternativo de acontecimientos. Los datos introducidos son incorrectos:

6.- El sistema comprueba que los datos introducidos no son correctos y muestra un mensaje acorde al error al usuario.

3.3.18.- Creación de incidencias

Este caso de uso contiene un caso de uso normal y un caso alternativo que se describen a continuación.

Caso de uso: Creación de incidencias.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: El usuario puede mantener un foro privado con los administradores del sistemas en el cual podrá gestionar tanto incidencias del dispositivo, como dudas sobre la aplicación.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita un listado de las incidencias creadas.	
	2.- El sistema comprueba las incidencias del cliente de usuario y se la muestra.
3.- El sistema solicita información de una incidencia en concreto.	
	4.- El sistema muestra la incidencia solicitada por el usuario

Curso alternativo de acontecimientos. El usuario solicita la creación de una incidencia:

3.- El usuario solicita la creación de una nueva incidencia introduciendo los datos.

4.- El sistema valida los datos introducidos por el usuario.

5.- El sistema almacena los datos introducidos por el usuario.

3.3.19.- Visualización de informes

Este caso de uso contiene un caso de uso normal y un caso alternativo que se describen a continuación.

Caso de uso: Visualización de informes.

Actores: Usuario supervisor.

Descripción: El usuario podrá gestionar distintos informes que se vayan generando relacionados con los dispositivos BeeTracker.

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita el listado de los informes del cliente al que pertenece.	
	2.- El sistema comprueba que el usuario tiene suficientes privilegios para visionar informes.
	3.- El sistema muestra la información de informes solicitada por el usuario

Curso alternativo de acontecimientos. El usuario no dispone de privilegios:

2.- El usuario comprueba que el usuario no tiene privilegios para la operación que se está solicitando y cancela la petición.

3.3.20.- Salir sesión

Este caso de uso contiene un caso de uso normal se describe a continuación.

Caso de uso: Salir sesión.

Actores: Usuario supervisor, usuario con permisos de escritura y usuario con permisos de lectura.

Descripción: Cierra la sesión del usuario

Requerimientos: Haber iniciado sesión.

Curso normal de los acontecimientos

Acciones de los actores	Acciones del sistema
1.- El usuario solicita salir de la aplicación.	
	2.- El sistema cierra la sesión y le devuelve al inicio de la aplicación.

4.- DISEÑO

En este apartado vamos a hablar del Diseño, explicando cual va a ser la arquitectura de la aplicación y los diferentes diseños de la aplicación, como son los de la interfaz de usuario, el diseño de la Base de Datos y el diseño de la capa de negocio.

4.1.- Arquitectura de la aplicación

4.2.- Diseño de la aplicación

4.2.1.- Diseño de la interfaz de usuario

4.2.1.1.- Diseño pantalla de registro

4.2.1.2.- Diseño básico de pantalla

4.2.1.3.- Diseño de formularios

4.2.1.4.- Diseño de listados

4.2.1.5.- Diseño de alarmas

4.2.1.6.- Diseño de gráficas

4.2.2.- Diseño de la Base de Datos

4.2.2.1.- Diseño de la Base de Datos intraLEPTON

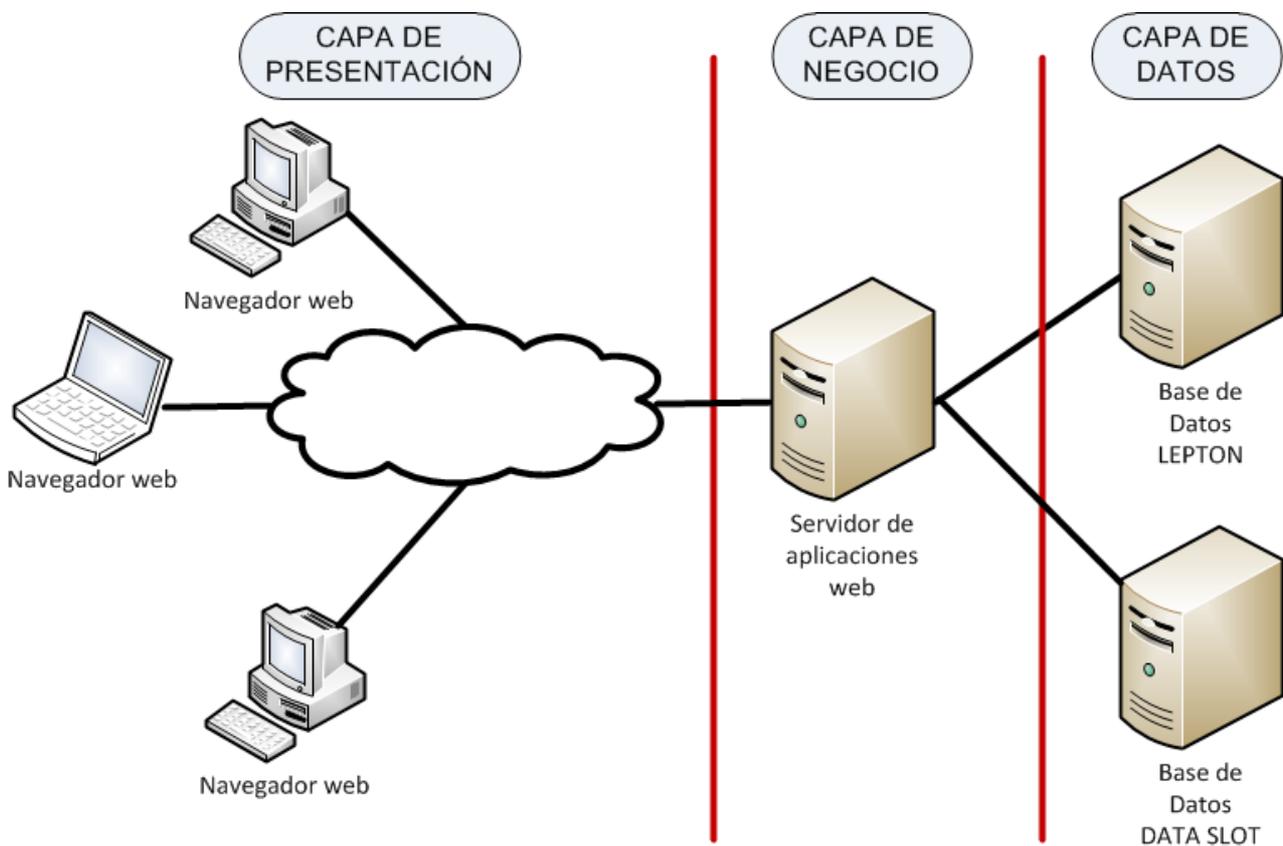
4.2.2.2.- Diseño de la Base de Datos DATA_SLOT

4.2.3.- Diseño de la capa de negocio

4.1.- Arquitectura de la aplicación

El diseño del proyecto sigue el modelo de arquitectura a 3 capas. Integrando la interfaz Web y modelo en un mismo servidor aunque conservando su independencia funcional. Ésta es la distribución en capas más común en las aplicaciones Web y es la que se ha utilizado en el proyecto.

- 1.- **Capa de presentación:** es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
- 2.- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- 3.- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



Esquema arquitectura a 3 capas

4.2.- Diseño de la aplicación

En esta parte mostrare de qué forma he creado la aplicación así como los detalles de todas las opciones que se pueden encontrar en la Web.

4.2.1.- Diseño de la interfaz de usuario

4.2.1.1.- Diseño pantalla de registro

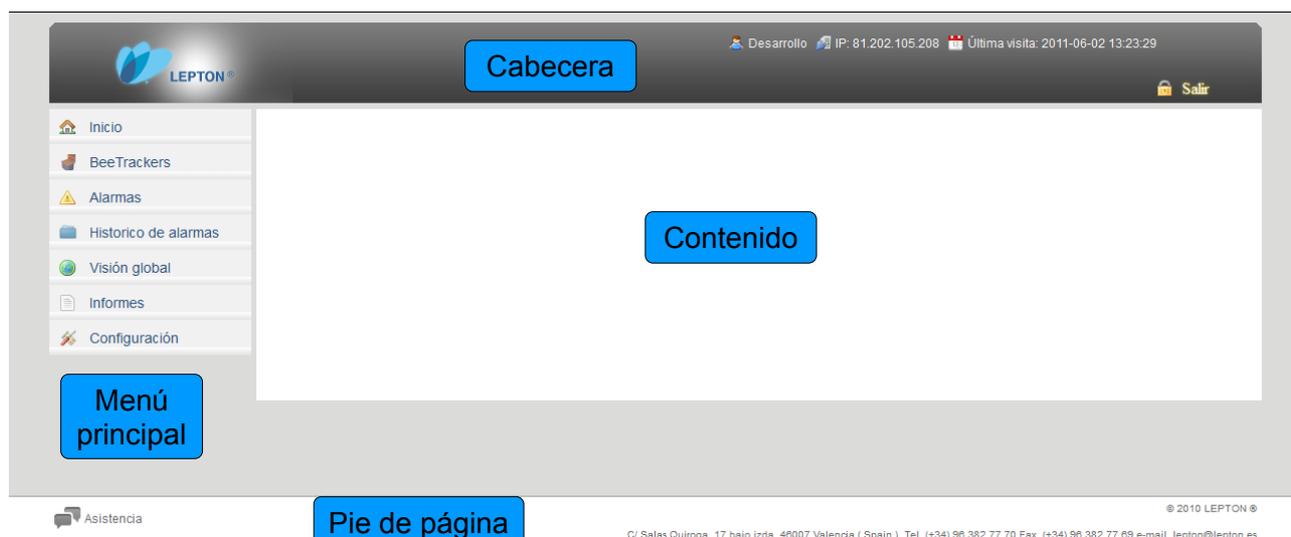
La página de registro, la única información que contiene es referente a la posibilidad de iniciar sesión o la de recuperación de contraseña, ya que al ser no se permite el registro en la misma si no es por parte de los administradores.



Registro de usuario

4.2.1.2.- Diseño básico de pantalla

La estructura básica que contendrán todas las páginas del proyecto, excepto en la asistencia en línea, es la siguiente:



Diseño de página básica

Cabecera. En esta sección nos encontraremos el logo de la empresa realizado mediante PhotoShop, información referente al último registro del usuario, mostrando la fecha y IP de último acceso, y así como el enlace para cerrar la sesión.

Menú principal. Se encuentra situado a la izquierda de la pantalla y en ella encontraremos todos los enlaces a las secciones más importantes de la aplicación. Dependiendo de que tipo de usuario que haya accedido, se mostrará un menú diferente.

Contenido. Es en la parte donde se mostrará toda la información de nuestra aplicación.

Pie de página. En la que encontraremos los datos de contacto de la empresa. Además en la parte izquierda se encontrará el enlace para la gestión de incidencias online.

4.2.1.3.- Diseño de formularios

A continuación se puede apreciar cual ha sido el estilo utilizado para los distintos formularios que nos pueden aparecer dentro de nuestra aplicación.

Nos encontramos con dos estilos de formularios, el primero de ellos, vemos que contiene un botón en el que se va a editar la información que se muestra, ya que para poder editar deberemos solicitar permiso y bloquear el recurso para conseguir que haya un estado de consistencia siempre en los datos que se están visionando.

Datos del usuario: Desarrollo

Datos personales

Nombre: * Apellidos:

Datos de contacto

E-mail: * Telf.: Móvil:

Botón editar

Zona horaria

Usted tiene seleccionada la zona horaria **Europe/Madrid** ¿Desea cambiarla?

Seleccione la nueva zona horaria

Formulario

En el segundo estilo de formularios, nos encontramos únicamente con un botón de guardar con el que se actualizarán los datos de forma automática, a partir de lo que se está viendo.

Avisos

Alarma de temperatura Temp. mín. : °C Temp. máx. : °C

Alarma de humedad H. mín. : % H. máx. : %

Alarma de sonido S. mín. : dB S. máx. : dB

Alarma de peso Peso máx. : Kg

Alarma de movimiento %

Alarma de batería % **Slider**

Botón guardar

Formulario de configuración

4.2.1.4.- Diseño de listados

Todos los listados que nos encontramos en la aplicación, nos encontraremos un título que nos indicará el listado que estamos viendo en cada momento.

Siempre contendrán unas cabeceras informando de cual es el tipo de información que contienen en dicha columna.

Los listados, que en algunos casos pueden ser bastante extensos, pueden encontrarse divididos mediante paginación o mediante un scroll lateral.

HISTÓRICO DE ALARMAS

⚠ | 12 Alarmas en total 🛠 | 0 Alarmas técnicas 📄 | 12 Alarmas por robo

Referencia	Tipo de alarma	Número de serie	Asentamiento	Fecha inicio	Fecha fin	Ver
00001395	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-30 09:21:41	2011-06-01 09:22:59	→
00001392	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-26 10:31:52	2011-05-26 11:16:10	→
00001391	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-18 07:34:44	2011-05-18 11:03:23	→
00001390	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-17 17:54:53	2011-05-18 02:09:55	→
00001389	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-16 19:54:36	2011-05-16 20:26:55	→
00001388	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-16 19:47:11	2011-05-16 19:48:34	→
00001387	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-14 21:17:46	2011-05-16 08:50:04	→
00001386	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-12 09:55:15	2011-05-14 14:35:52	→
00001383	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-11 11:16:25	2011-05-11 12:51:18	→
00001379	Robo de colmena	BT0000002	Pruebas LEPTON	2011-05-10 09:53:09	2011-05-10 12:44:48	→

✖ Borrar alarmas seleccionadas



Listado de alarmas

4.2.1.5.- Diseño de alarmas

Los listados de alarmas activas son un tipo de diseño bastante importante y del que cabe mencionar su funcionamiento.

LISTADO DE ALARMAS

⚠ | 7 Alarmas en total 🛠 | 7 Alarmas técnicas 📄 | 0 Alarmas por robo

Referencia	Tipo de alarma	Número de Série	Asentamiento	Fecha inicio	Cerrar	Atender
00001357	Temperatura	BT0000002	Pruebas LEPTON	2011-04-18 11:31:09	✖	→
00001356	Temperatura	BT0000002	Pruebas LEPTON	2011-04-16 17:22:19	✖	→
00001355	Temperatura	BT0000002	Pruebas LEPTON	2011-04-16 16:17:39	✖	→
00001339	Temperatura	BT0000002	Pruebas LEPTON	2011-04-14 17:52:25	✖	→
00001326	Temperatura	BT0000002	Pruebas LEPTON	2011-04-11 18:50:54	✖	→
00001325	Temperatura	BT0000002	Pruebas LEPTON	2011-04-11 18:50:54	✖	→
00001324	Temperatura	BT0000002	Pruebas LEPTON	2011-04-11 18:50:54	✖	→

Botón cerrar alarma

Botón atender alarma

Listado de alarmas activas en el sistema

Como se ve en la imagen, en los listado de alarmas podemos encontrar información acerca de las alarmas que se han producido, así como realizar acciones sobre ellas. Las acciones que se pueden realizar sobre una alarma son: cerrar una alarma y atender una alarma. La primera de ellas solo es accesible si el usuario que solicita la acción tiene privilegios en el sistema y conlleva que la alarma pase a almacenarse en el histórico. La segunda de ellas, nos permite ampliar la información de la alarma, visionando así toda la información recogida por el BeeTracker.

4.2.1.6.- Diseño de seguimiento

Los seguimientos son el punto más complicado de la aplicación, ya que recoge los datos en tiempo real que son transmitidos por el dispositivo. El diseño de estas página contendrá información sobre el último punto de información enviado por el dispositivo y teniendo la posibilidad de ampliar a todos los datos transmitidos para así trazar la ruta que ha llevado el dispositivo desde el momento en el que arranco la alarma.

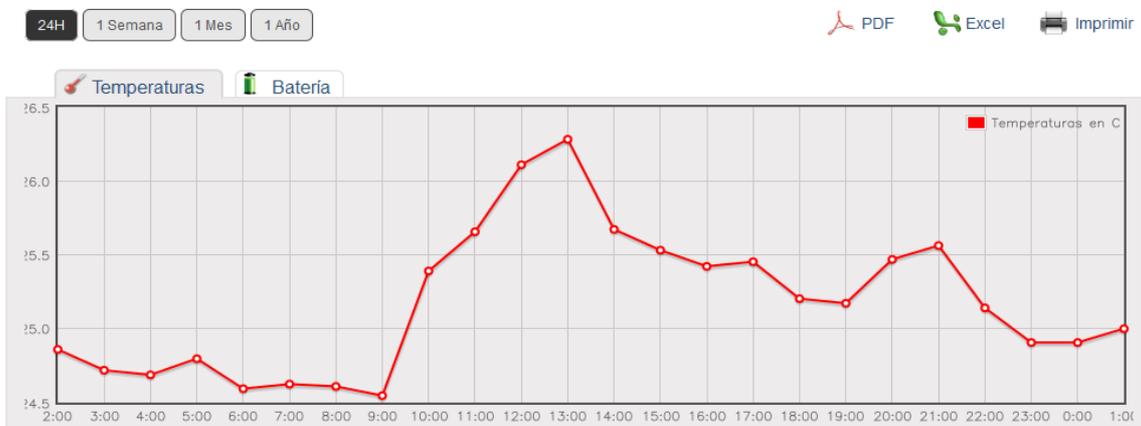


En este tipo de pantalla, nos encontraremos un área en la que se situará en un mapa el dispositivo y unos botones que nos permitirán manipularlo. También nos mostrarán toda la información de parámetros de temperatura, humedad, sonido y batería que se hayan recogido en forma de gráficas.

4.2.1.6.- Diseño de gráficas

Por último, otro de los diseños que cabe comentar es el diseño de las gráficas.

Las gráficas en la aplicación Web BeeTracker son bastante importante ya que es la forma más sencilla de mostrar toda la información recogida por los dispositivos. De este modo, nos podemos encontrar con gráficas de temperatura, sonido, humedad, peso y batería de los datos recogidos. Si no se produce ningún fallo en el sistema, los BeeTracker recogen los datos de hora en hora por lo que se puede ver una evolución diaria de todos los datos recogidos.



Diseño de gráficas

4.2.2.- Diseño de la Base de Datos

Dentro de la Aplicación Web BeeTracker, nos encontramos ante dos Bases de Datos de las cuales hay que hacer mención. La primera de ellas sería intraLEPTON, que es en aquella donde radican todos los datos referentes a los clientes de la aplicación, mientras que en la segunda, denominada DATA_SLOT es donde se encuentran los datos concretos de cada dispositivo BeeTracker.

4.2.2.1.- Diseño de la Base de Datos intraLEPTON

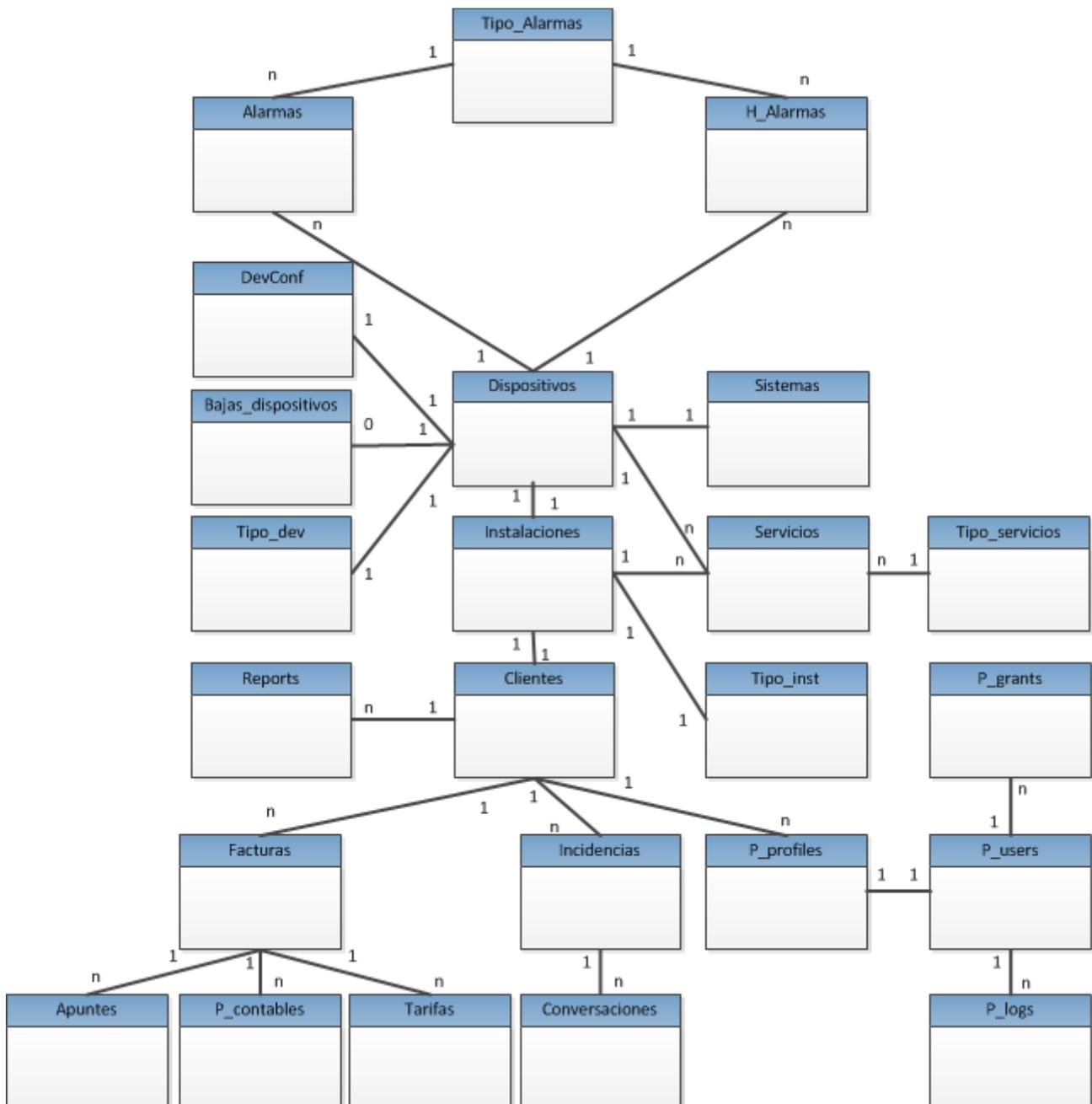


Diagrama de tablas de intraLEPTON

A continuación vamos a explicar cada una de las tablas que nos encontramos en el diagrama anterior:

Clientes: En esta tabla nos encontramos con los datos de todos los clientes que usan esta aplicación. Nos encontramos desde los datos identificativos de los clientes, el estado en el que se encuentran, el logo e incluso la persona de contacto con la que hay que contactar si nos encontramos con algún problema en alguno de sus dispositivos.

P_profiles: Es donde se encuentran los datos de usuario de cada uno de los clientes que hay en el sistema. Nos encontramos con los datos personales de cada uno de los usuarios que tiene acceso a nuestro sistema.

P_users: Es la tabla en la que se almacenan los usuarios y contraseña para acceder dentro de nuestra aplicación web.

P_grants: Todo usuario que desee acceder al sistema debe de estar reflejado en esta tabla. Es donde nos indica a que tipo el tipo de acceso que tiene al sistema, así como a que instalaciones tiene acceso y con que permisos accede.

Instalaciones: Dentro del sistema, todo dispositivo debe pertenecer a una instalación, por lo que nos encontramos que todo cliente debe tener una instalación asociada. Se mantiene información de la localización de la instalación

Tipo_inst: Es la tabla en la que nos indica los tipos de instalaciones que existen en el sistema.

Servicios: Es la tabla donde se nos indica los servicios asociados a las instalaciones o a los dispositivos. Esto es que tipo de características van a contener cada uno de nuestras instalaciones o dispositivos.

Tipo_servicios: Es la tabla en la que nos indica los tipos de servicios que existen en el sistema.

Sistemas: Es la información relativa a todos los sistemas que existen dentro de LEPTON, informándonos de como poder acceder a cada uno de ellos.

Dispositivos: Es una de las principales tablas de nuestro sistema, ya que en ella se contiene cada uno de los dispositivos de cada cliente. Contiene los datos en que sistema se encuentra el dispositivo, el estado en el que se encuentra el dispositivo y cual fue su fecha de alta en el sistema.

DevConf: Es una copia de la tabla Conf situada en la Base de Datos DATA_SLOT que más adelante comentaremos.

Baja_dispositivos: Es donde se encuentran todos los dispositivos que han sido dados de baja con el motivo por el cual fueron dados de baja.

Tipo_dev: Al igual que ocurre con las instalaciones, en esta tabla se nos indica los tipos de dispositivos existentes en el sistema. Dentro de un instalación concreta solo existen un tipo de dispositivo.

Alarmas: Es donde se nos indica la información básica sobre todas las alarmas activas actualmente en el sistema. Nos indica que dispositivo ha dado la alarma, el tipo de alarma que ha saltado y a que sistema debemos de conectarnos para comprobar los datos más detallados de la alarma.

H_Alarmas: Es donde se nos indica la información básica sobre todas las alarmas cerradas en el sistema. Nos indica que dispositivo ha dado la alarma, el tipo de alarma que ha saltado y a que sistema debemos de conectarnos para comprobar los datos más detallados de la alarma.

Tipo_Alarmas: Es la tabla en la que están asociados los tipos de alarma con sus descripciones.

Reports: En esta tabla es donde son recogidos los diferentes informes generados de forma automática por el sistema. Nos encontramos desde los informes de las alarmas de seguimiento de los dispositivos, a los informes mensuales de todas las telemidas recogidas y los datos de acceso al sistema clasificados por clientes.

Facturas: Es la tabla donde se almacenan todas las facturas generados a los diferentes clientes.

Tarifas: Es donde nos vamos a encontrar la información de cada una de las tarifas existentes a la hora de poder facturar a los clientes.

Apuntes: Es donde está registrada cada uno de los apuntes contables que se le realizan a un cliente a partir de todos los dispositivos que tiene asignados.

P_contables: Es la tabla donde se registra el IVA actual del sistema para poder generar las facturas, así como los distintos anuncios que se deseen introducir en las mismas.

Incidencias: Es donde se almacena la información de una incidencia de consulta creada por un usuario.

Conversaciones: Es la tabla en la que estas las conversaciones asociada a cada una de las incidencias que un usuario ha generado.

P_logs: Es donde se produce el registro de todas las acciones realizadas por los clientes que conlleve un cambio de estado en el sistema.

4.2.2.2.- Diseño de la Base de Datos DATA_SLOT

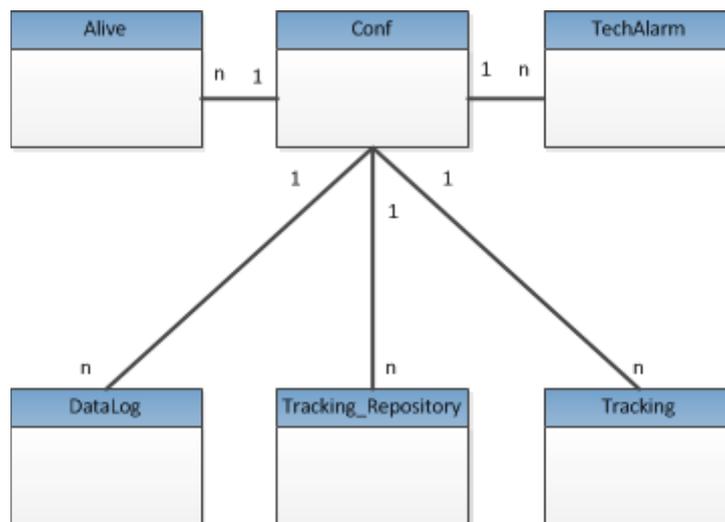


Diagrama de tablas de DATA_SLOT

A continuación vamos a explicar cada una de las tablas que nos encontramos en el diagrama anterior:

Alive: Es donde se almacena la información que transmite el dispositivo diariamente para indicarnos que su funcionamiento está siendo correcto.

Conf: Esta tabla contiene toda la información de configuración de cada dispositivo. Contiene desde la configuración de transmisión, a los datos por los cuales deben saltar alarmas. Como habíamos mencionado en el punto anterior, esta tabla Conf tiene una copia dentro de intraLEPTON denominada DevConf.

TechAlarm: Es la tabla que contiene toda la información de las denominadas alarmas técnicas.

DataLog: En esta tabla es donde se inserta todos los datos recogidos por los dispositivos a lo largo del tiempo, es decir, aquí está toda la información de las distintas telemidas recogidas por los dispositivos.

Tracking: Es la tabla de los seguimientos en activo de los dispositivos, nos encontramos con información relativa a la posición del dispositivo, todos los parámetros de telemidas recogidos por los dispositivos y cuales son las coberturas de telefonía que detecta. Una de las condiciones de esta tabla, es que todo dato de un mismo dispositivo en esta tabla se corresponden al mismo seguimiento.

Tracking_Repository: En esta última tabla del sistema, nos encontramos la misma información que en Tracking pero de los seguimientos ya han sido cerrados. Aquí ya tienen asociado un número de alarma de seguimiento.

4.2.3.- Diseño de la capa de negocio

En la capa de negocio, es donde reside toda la lógica de aplicación. Aquí nos encontramos con los ficheros que hacen que toda la aplicación Web BeeTracker funcione. Estos están formados con lenguajes de programación como PHP, HTML o JavaScript.

Como se ha dicho en la definición es la capa de unión entre la capa de datos, donde reside toda la información de nuestro sistema con lo que el usuario ve posteriormente. En ella se realiza todos los cálculos, procesos necesarios para mostrar una información coherente, sencilla y simple para el usuario.

5.- IMPLEMENTACIÓN

En este apartado, es donde vamos a exponer todos el desarrollo que se ha realizado en el nuestro proyecto. Vamos a abordar, desde las tecnologías utilizadas para su realización, así como la estrategia que se ha decido seguir, la planificación del mismo y haciendo especial mención a la utilización de todos los ficheros de la aplicación y a todo lo relacionado con la seguridad de nuestra aplicación.

-
- 5.1.- Tecnologías utilizadas
 - 5.1.1.- Navegadores web
 - 5.2.- Estrategia para el desarrollo de la aplicación
 - 5.3.- Planificación del proyecto
 - 5.4.- Archivos creados y desarrollo de la aplicación
 - 5.4.1.- Diagrama navegacional
 - 5.4.2.- Estructura de directorios
 - 5.4.1.- Diagrama navegacional
 - 5.4.2.- Estructura de directorios
 - 5.4.3.- Uso de principales ficheros intrabeetracker
 - 5.4.3.1.- Comienzo del proyecto
 - 5.4.3.2.- Permisos de usuario
 - 5.4.3.3.- Sección BeeTrackers
 - 5.4.3.4.- Sección BeeTracker
 - 5.4.3.5.- Sección alarmas
 - 5.4.3.6.- Sección seguimiento
 - 5.4.3.7.- Sección seguimiento histórico
 - 5.4.3.8.- Sección visión global
 - 5.4.3.9.- Sección informes
 - 5.4.3.10.- Sección configuración
 - 5.4.3.11.- Sección ayuda en linea
 - 5.5.- Seguridad de la aplicación
 - 5.5.1.- Acceso a la aplicación
 - 5.5.2.- Validación de acceso
 - 5.5.3.- Permisos de usuario
 - 5.5.4.- Bloqueo de información
-

5.1.- Tecnologías utilizadas

Como se comentó en el punto 2.4 las tecnologías utilizadas serán los lenguajes de programación PHP, HTML, JavaScript, la tecnología de la librería XAJAX, el sistema gestor de bases de datos MySQL y usando un servidor web Apache.

La aplicación web se ha desarrollado el sistema operativo Ubuntu Linux 10.04, con la aplicación APTANA Studio 2.0.

Cabe mencionar, que para la realización de los distintos informes que se muestran en la aplicación se han desarrollado con Java y el entorno IDE NetBeans.

APACHE 2.2.9 (Unix)

Es un servidor de programación libre y es el que aplican la mayoría de proveedores de páginas Web, la configuración del servidor permite tanto la integración de módulos para extender su funcionalidad como la asignación de alías o la ejecución de scripts CGI's para la ejecución de programas.

Un servidor que está siempre en ejecución, manteniéndose a la espera de recepción de peticiones por parte de los clientes (navegadores).

MYSQL 5.0

MySQL es un sistema gestor de base de datos de programación libre, Es uno de los más utilizados debido a su facilidad de uso y a su rapidez.

Permite el uso del lenguaje de programación SQL.

Utiliza un motor de guardado llamado MyISAM de forma predeterminada.

PHP 5.2.9

PHP es un lenguaje interpretado muy utilizado para el desarrollo de páginas Web de forma dinámica, está diseñado para realizar dicha labor tomando el código en PHP como su entrada y creando páginas Web como salida.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle,...

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos.

Es un lenguaje fácil de aprender y de programación libre.

APTANA STUDIO

Aptana Studio es un entorno de desarrollo integrado gratuito basado en eclipse y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux y provee soporte para lenguajes como: Php, Python, Ruby, CSS, AJAX, HTML y Adobe AIR. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades.

Sus principales características son:

- Asistente de código para HTML y Javascript.
- Librerías AJAX (jQuery, prototype...).
- Conexión vía FTP, SFTP y FTPS.
- Herramientas para trabajo con base de datos.
- Marcado de sintaxis mediante colores.
- Compatible con extensiones para Eclipse.

LIBRERÍA XAJAX 0.2

Xajax es una biblioteca de código abierto para PHP que permite crear de manera fácil y simple aplicaciones Web basadas en AJAX usando además HTML, CSS, y Javascript. Las aplicaciones desarrolladas con Xajax pueden comunicarse asincrónicamente con funciones que se encuentran del lado del servidor y así actualizar el contenido de una página sin tener que recargarla nuevamente, su última versión es la 0.5 Final que cambia ligeramente comparado con las versiones anteriores 2.5. y anteriores.

Por otro lado, la biblioteca Xajax es una biblioteca puramente centrada en PHP. Con la introducción de la misma el manejo de AJAX en PHP se hace mucho más sencillo y sobre todo solo se necesita escribir sentencias muy cortas y simples en Javascript, lo que supone, según sus creadores, una ventaja frente a otras soluciones.

Otras de las características son:

- Xajax es compatible con Firefox, Mozilla, Internet Explorer, Opera y Safari
- Xajax puede ser usado para actualizar estilos, clases CSS, botones de selección, casillas de verificación y botones de radio o cualquier otro atributo de un elemento
- Cada función registrada para ser accesible a través de Xajax puede tener distintos tipos de petición.

HTML

HTML, HyperText Markup Language, es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

CSS

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

NETBEANS

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

5.1.1.- Navegadores web

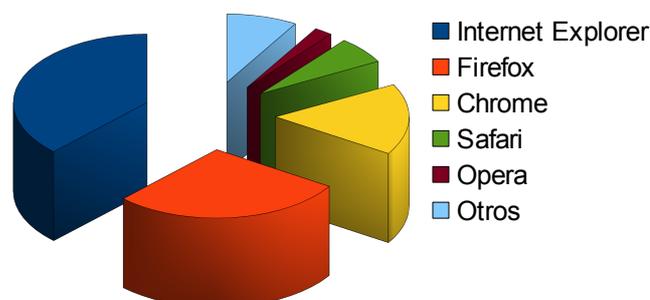
Un navegador web es un programa que permite visualizar la información que contiene una página web. El navegador interpreta el código, XHTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

El proyecto Aplicación Web BeeTracker puede ser interpretado por cualquiera de los navegadores que existen en el mercado, no obstante es preferible el uso del Mozilla Firefox, Chrome, Opera o de Internet Explorer a partir de su versión 8, aunque se recomienda el uso de Internet Explorer 9 por las nuevos soportes que aporta. Esto se debe para evitar problemas de incompatibilidad en las vistas de las páginas, debido a que ciertos objetos no son representados correctamente en versiones anteriores de Internet Explorer 8.

A continuación, se muestra las estadísticas de cuotas de mercado en mayo de 2011 obtenidas de la página w3counter.com.

1	Internet Explorer 8	22,29%
2	Firefox 4	14,82%
3	Chrome 11	14,13%
4	Firefox 3.6	11,03%
5	Internet Explorer 7	8,73%
6	Safari 5	5,43%
7	Internet Explorer 9	3,76%
8	Internet Explorer 6	2,91%
9	Firefox 3.5	1,48%
10	Chrome 10	1,25%

Cuota de mercado (mayo 2011)



De estas estadísticas podemos obtener las siguientes conclusiones:

- Uno de nuestras recomendaciones establecidas anteriormente es el uso de Internet Explorer a partir de su versión 8, esto supone que del 37,7% de cuota de mercado que tiene Internet Explorer estamos abarcando el 26,05%.
- De los exploradores incluidos en el en los 10 más usado se está dando soporte a todos ellos, excepto a los 2 comentados anteriormente que son Internet Explorer 6 e Internet Explorer 7, por lo que estamos abarcando el 74,19% de un total del 85,83% de la cuota de mercado que alcanzan estos 10 navegadores.

5.2.- Estrategia para el desarrollo de la aplicación

Para la realización del proyecto Aplicación Web BeeTracker se ha utilizado el modelo de desarrollo denominado Extreme Programming, es una metodología de desarrollo ágil en el que los requisitos cambian durante el tiempo que dura el ciclo de vida de la aplicación, en el que cualquier cambio de requisito hace que se adapte la aplicación.

La aplicación ha sido desarrollada de forma iterativa e incremental, de forma que conforme se iban acabando las partes funcionales de la aplicación, se iban corrigiendo sus errores.

5.3.- Planificación del proyecto

Dentro de todo proyecto debe existir una planificación por la cual tanto el desarrollador como los responsables del proyecto conozcan cual es el estado del mismo, si se están produciendo retrasos, si se van cumpliendo los objetivos o si se están produciendo adelantes en el mismo.

A continuación se muestra cual ha sido la planificación inicial propuesta y consensuada por desarrolladores y los responsables del proyecto BeeTracker para la realización del mismo.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	Aplicación web BeeTracker	70 días	jue 13/01/11	vie 29/04/11		Desarrollador
2	Registro en el sistema	25 hrs	jue 13/01/11	vie 21/01/11		Desarrollador
3	BeeTrackers	30 hrs	vie 21/01/11	mar 01/02/11	2	Desarrollador
4	BeeTracker	30 hrs	mar 01/02/11	vie 11/02/11	3	Desarrollador
5	Alarmas	35 hrs	vie 11/02/11	mié 23/02/11	4	Desarrollador
6	Seguimiento	30 hrs	jue 24/02/11	lun 07/03/11	5	Desarrollador
7	Historico	20 hrs	lun 07/03/11	lun 14/03/11	6	Desarrollador
8	Seguimiento historico	20 hrs	lun 07/03/11	lun 14/03/11	6	Desarrollador
9	Vision global	20 hrs	lun 14/03/11	jue 24/03/11	7;8	Desarrollador
10	Configuracion	10 hrs	jue 24/03/11	lun 28/03/11	9	Desarrollador
11	Informes	30 hrs	mar 29/03/11	jue 07/04/11	10	Desarrollador
12	Asistencia on-line	20 hrs	jue 07/04/11	mar 19/04/11	11	Desarrollador
13	Permisos de usuarios	30 hrs	mar 19/04/11	vie 29/04/11	12	Desarrollador

Listado de tareas

El proyecto está dividido en un total de 12 tareas que han de durar un periodo 70 días laborables con una jornada laboral de 4 horas. A continuación en el diagrama de Gantt vemos con más claridad las fases de desarrollo al completo.

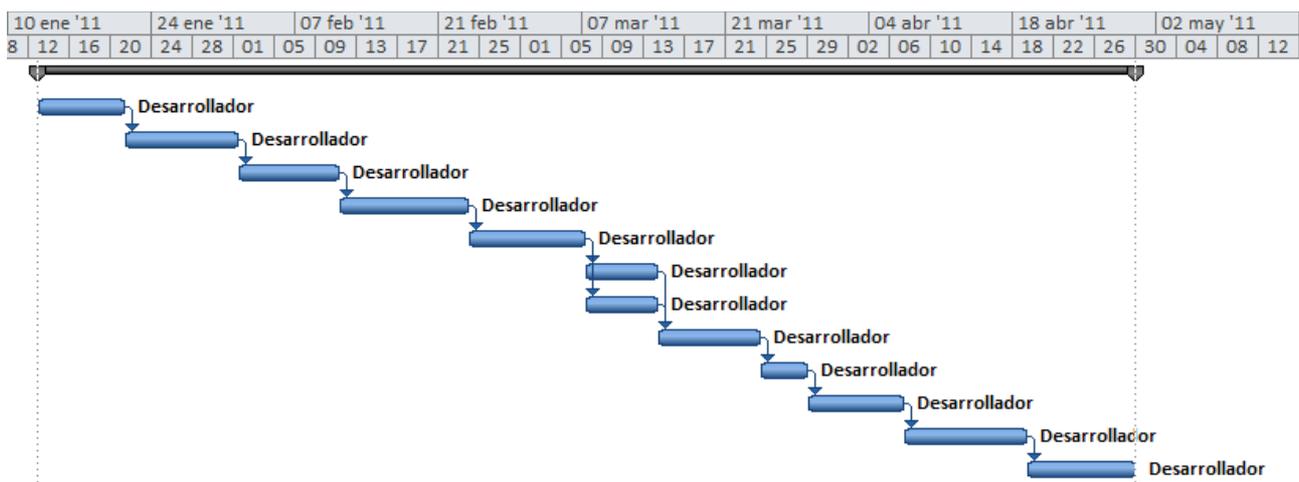


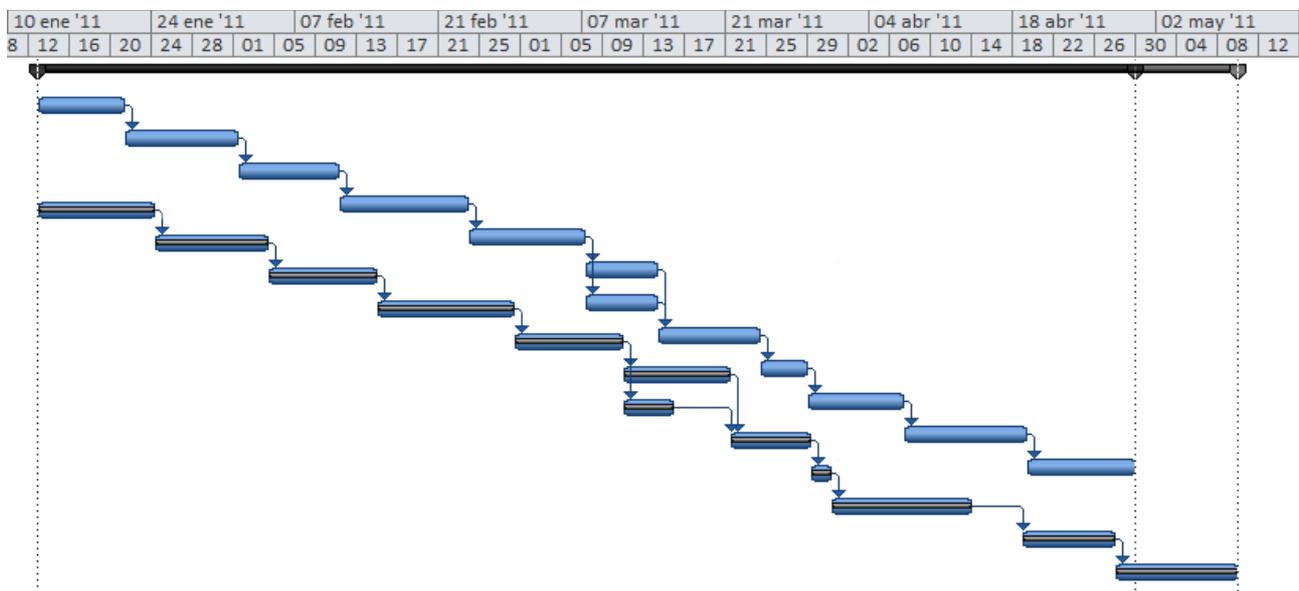
Diagrama de Gantt

Una vez concluido el proyecto, vamos a realizar una comparación con la planificación inicial que se estimo del proyecto con los datos reales del mismo para así conseguir abordar cuales han sido los principales puntos de dificultad dentro del mismo.

	Nombre de tarea	Duración	Comienzo	Fin	Duración	Comienzo	Fin
1	Aplicación web BeeTracker	70 días	jue 13/01/11	vie 29/04/11	76 días	jue 13/01/11	lun 09/05/11
2	Registro en el sistema	25 hrs	jue 13/01/11	vie 21/01/11	30 hrs	jue 13/01/11	lun 24/01/11
3	BeeTrackers	30 hrs	vie 21/01/11	mar 01/02/11	35 hrs	lun 24/01/11	vie 04/02/11
4	BeeTracker	30 hrs	mar 01/02/11	vie 11/02/11	27 hrs	vie 04/02/11	lun 14/02/11
5	Alarmas	35 hrs	vie 11/02/11	mié 23/02/11	38 hrs	mar 15/02/11	lun 28/02/11
6	Seguimiento	30 hrs	jue 24/02/11	lun 07/03/11	34 hrs	lun 28/02/11	jue 10/03/11
7	Historico	20 hrs	lun 07/03/11	lun 14/03/11	15 hrs	vie 11/03/11	lun 21/03/11
8	Seguimiento historico	20 hrs	lun 07/03/11	lun 14/03/11	12 hrs	vie 11/03/11	mar 15/03/11
9	Vision global	20 hrs	lun 14/03/11	jue 24/03/11	23 hrs	lun 21/03/11	mar 29/03/11
10	Configuracion	10 hrs	jue 24/03/11	lun 28/03/11	8 hrs	mar 29/03/11	jue 31/03/11
11	Informes	30 hrs	mar 29/03/11	jue 07/04/11	38 hrs	jue 31/03/11	mié 13/04/11
12	Asistencia on-line	20 hrs	jue 07/04/11	mar 19/04/11	24 hrs	mar 19/04/11	mié 27/04/11
13	Permisos de usuarios	30 hrs	mar 19/04/11	vie 29/04/11	32 hrs	jue 28/04/11	lun 09/05/11

Comparación de tareas

Por último, se muestra un nuevo diagrama de Gantt integrando tanto el diagrama inicial como el diagrama final para mostrar cual a sido la planificación final del proyecto.



Comparación diagramas de Gantt

Como conclusiones de estos diagramas, podemos sacar el mayor retraso se produjo al principio del mismo con las 5 primeras tareas, pero posteriormente se han ido cumpliendo los tiempos de ejecución o incluso mejorando en alguno de los casos. Todo esto ha llevado a que el proyecto haya tenido un retraso de 6 días laborables respecto a la planificación inicial del proyecto.

5.4.- Archivos creados y desarrollo de la aplicación

5.4.1.- Diagrama navegacional

Dentro de nuestra aplicación nos podemos encontrar 3 tipos de usuario:

Usuario supervisor: Será aquel que tenga un acceso completo a nuestro sistema, podrá realizar todas las tareas sin ninguna excepción.

Este sería el diagrama navegacional de un usuario supervisor:

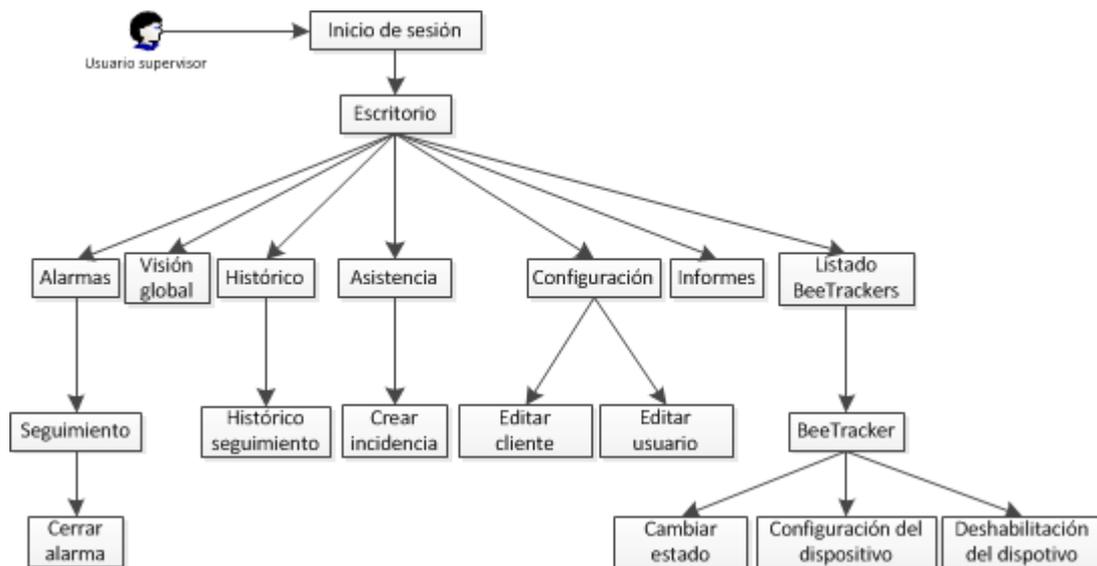


Diagrama navegacional usuario supervisor

Usuario con permisos de escritura: Este tipo de usuario puede realizar tareas que signifiquen una modificación del estado en el que se encuentra un dispositivo.

El usuario con permisos de escritura tiene el siguiente diagrama navegacional:

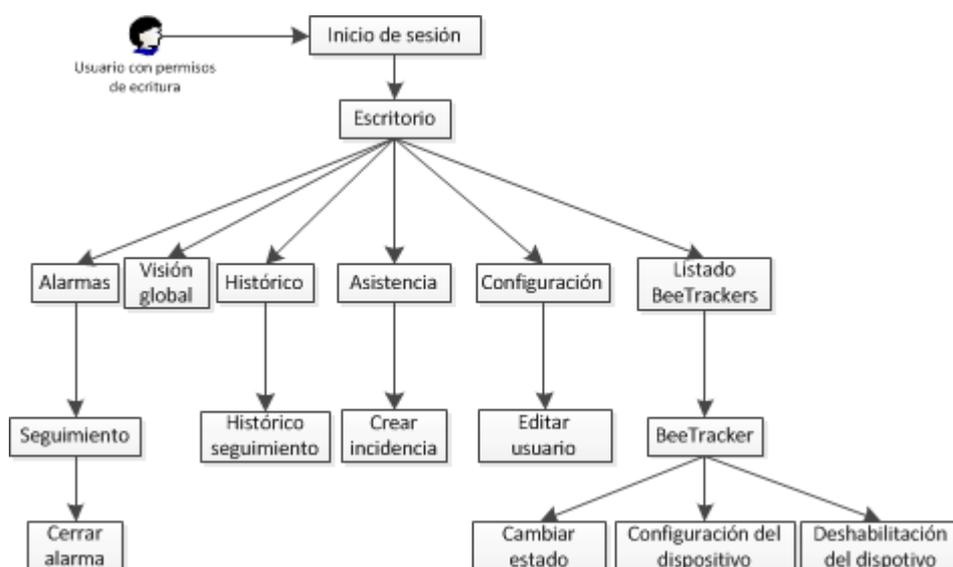


Diagrama navegacional usuario con permisos de escritura

Usuario con permisos de lectura: Este tipo de usuario únicamente podrá visualizar el contenido del dispositivo sin poder realizar ninguna acción sobre ellos.

El usuario con permisos de lectura tiene el siguiente diagrama navegacional:



Diagrama navegacional de usuario con permisos de lectura

5.4.2.- Estructura de directorios

La estructura de los directorios de la aplicación web BeeTracker son:

Directorio raíz

En el directorio raíz nos vamos a encontrar los directorios y archivos que contienen toda nuestra aplicación web, desde la fase de inicio de sesión a la correcta gestión de la aplicación.

En este directorio nos encontramos con 7 directorios y 7 ficheros. Estos siete ficheros contienen la página de inicio de sesión que es login.php y la página menu.php donde el usuario podrá seleccionar el acceso a la Intranet deseado. También nos encontramos con el fichero actualizar_password.php que nos permitirá realizar un cambio de password nada más el usuario acceda al sistema. Tenemos el directorio css que contiene las hojas de estilo e imágenes necesarias para esta primera página, mientras que en el directorio lib encontraremos tanto la cabecera como el fichero de recuperación de contraseña y el directorio de librerías Javascript. En el directorio img contiene las imágenes para el menú de selección de la Intranet y por último tenemos el directorio xajax_lib donde se encuentran las librerías de XAJAX.

- **Directorio AccessControl**
- **Directorio Conf**
- **Directorio css**
 - **Directorio images** (contiene 8 imágenes)
 - ie6_screen_login.css
 - css.css
 - ie7.css
 - otros.css
 - screen_login.css

- **Directorio img** (contiene 19 imágenes)
- **Directorio intrabeetracker**
- **Directorio lib**
 - **Directorio jquery** (contiene 4 ficheros)
 - actualizar_password_ajax.php
 - actualizar_password_js.js
 - header.php
 - head_login.php
 - head.php
 - login_ajax.php
- **Directorio xajax_lib** (contiene librería xajax)
- actualizar_password.php
- brutus.php
- error.php
- index.html
- index.php
- login.php
- menu.php

Los otros tres directorios se explican a continuación con más detalle.

Directorio AccessControl

El directorio AccessControl es el encargado de la seguridad de toda la aplicación web, también contiene el fichero Logger.php que se encarga de registrar todo en un log. En un posterior apartado realizaremos un estudio sobre la seguridad de nuestra aplicación web BeeTracker.

- **Directorio Auth**
 - Inst_val.php
 - Auth.php
- **Directorio Logger**
 - Logger.php
- **Directorio Session**
 - Session.php
- CAC_module_menu.php
- CAC_module.php

Directorio Conf

En el directorio Config.php nos encontramos con las variables necesarias para el correcto funcionamiento de toda la aplicación web.

- Config.php

Directorio intrabeetracker

Por último falta hablar del directorio intrabeetracker que es donde se encuentra alojada la aplicación web BeeTracker. Este directorio se explica de forma más concreta a continuación:

- **Directorio css**
- **Directorio img** (contiene 119 imágenes)
- **Directorio lib**
- **Directorio tmp**
- **Directorio xajax_lib**
- Alarmas.php
- Beetracker.php
- beetrackers.php
- config.php
- crear_incidencia.php
- Dashboard.php
- error.php
- generarCSV.php
- generarGPX.php
- GenerarPDF.php
- generar_mensual_CSV.php
- generar_seg_CSV.php
- help.php
- historico.php
- h_seguimiento.php
- index.html
- index.php
- informes.php
- Localizacion.php
- saveIncidencia.php
- seguimiento.php
- ver_documento.php
- ver_factura.php
- ver_informe.php

El directorio intrabeetracker contiene 5 directorios, 2 de los cuales explicaremos de forma más detallada, y 25 ficheros php.

Los tres directorios menos importantes son img, que contiene todas las imágenes empleadas por la aplicación web, el directorio tmp, que es un directorio en el que se generan las imágenes para los pdf que solicita el usuario y el directorio xajax_lib, en el cual nos encontramos todos los ficheros necesarios para el uso de la librería XAJAX.

Dentro de este directorio nos encontramos todos los ficheros php que se ejecutan para nuestra aplicación web BeeTracker. Más adelante, se realizará un estudio sobre los principales ficheros que se usan en la aplicación así como se relacionan entre ellos.

Directorio intrabeetracker/css

En este directorio nos encontramos todas las hojas de estilo que se utilizan en la aplicación web. No encontramos 3 tipos de css.

Las hojas de estilo para la impresión:

- Beetracker_print.css
- seguimiento_print.css
- h_seguimiento_print.css

Las hojas de estilo del apartado de asistencia on-line:

- css_asistencia.css
- ie7_asistencia.css
- otros_asistencias.css

Y por últimos, las hojas de estilo de la aplicación web:

- css.css
- ie7.css
- asistencia.css

Directorio intrabeetracker/lib

El directorio lib de intrabeetracker contiene todos los archivos secundarios necesarios para la correcta ejecución de la aplicación web BeeTracker.

- **Directorio ajax**
- **Directorio cms_xhtml**

En este directorio nos encontraremos una librería para el uso del calendario. Contiene elementos Javascript, imágenes y hojas de estilo.
- **Directorio flot**

En el directorio flot tenemos todo lo relativo al uso de gráficas en la aplicación. También nos encontramos con la librería jquery que se utiliza en todas las ventanas de la aplicación.
- **Directorio tcpdf**

Con los ficheros que encontramos dentro del directorio tcpdf podemos generar archivos pdf solicitados por el usuario.
- **Directorio trackbar**

En este directorio compuesto por ficheros Javascript, imágenes y hojas de estilo es como se consigue los sliders de los formularios.
- **Directorio js**
- base64.js – Este fichero Javascript convierte una imagen a base 64 para poder insertarlo dentro de un pdf.
- canvas2image.js – El fichero Javascript permite transformar un objeto canvas en una imagen con la ayuda del fichero que hemos comentado previamente.
- footer.php – Fichero php que se incluye en todos los ficheros. Es el pie de página.
- head.php – Es el fichero php que se incluye en todas las páginas y contiene la información de las hojas de estilo que se utilizan, el título de la página y los ficheros Javascript conjuntos.
- header.php – Es el fichero php que contiene la cabecera que se ve en las páginas.

- menu.php – Como su propio nombre indica es el fichero que contiene el menú principal de la aplicación.

A continuación, se comentaran los directorios ajax y js de este directorio.

Directorio intrabeetracker/lib/ajax

El directorio ajax contiene todos los documentos ajax de todas las pantallas de la aplicación. La estructura de nombre que tiene es el del fichero php contenido en la carpeta intrabeetracker añadiéndole "_ajax.php" a su nombre.

- Alarmas_ajax.php
- Beetracker_ajax.php
- beetrackers_ajax.php
- config_ajax.php
- h_seguimiento_ajax.php
- help_ajax.php
- historico_ajax.php
- informes_ajax.php
- Localizacion_ajax.php
- seguimiento_ajax.php

Directorio intrabeetracker/lib/js

Al igual que en el directorio ajax, previamente comentando, en esta carpeta encontramos todos los ficheros Javascript generados para la aplicación web . La estructura de nombre del directorio es el mismo que contiene la carpeta intrabeetracker añadiendo al final del mismo "_js.js"

- Alarmas_js.js
- Beetracker_js.js
- beetrackers_js.js
- config_js.js
- crear_incidencia_js.js
- eshapes.js
- h_seguimiento_js.js
- help_js.js
- historico_js.js
- informes_js.js
- Localizacion_js.js
- seguimiento_js.js

5.4.3.- Uso de principales ficheros intrabeetracker

En este apartado vamos a realizar una explicación de la funcionalidad e implementación de las secciones más importantes de la aplicación, así como ver cuales han sido los principales problemas que han ido surgiendo durante el desarrollo de la aplicación web y como se han ido solucionando hasta conseguir el objetivo deseado.

Como se ha comentado anteriormente, los ficheros comunes a toda la aplicación son:

- lib/header.php
- lib/footer.php
- lib/menu.php
- lib/flot/jquery.js
- css/ie7_css.css
- css/css.css
- css/otro_css.css

5.4.3.1.- Comienzo del proyecto

Uno de los principales problemas con los que me encontré al comenzar este proyecto fue que todas las horas que teníamos en la base de datos almacenadas estaban en uso horario UTC. Esto podía hacer que los usuarios cayeran en confusión en cuanto a las horas se refieren, por lo que se configuró el sistema en un primer lugar, para que el navegador mediante Javascript nos indicara el uso horario del cliente y así a la hora de mostrar al usuario las horas les añadiéramos el desfase correspondiente.

Esto nos llevo, a que si en el servidor del usuario estos datos estaban mal configurados, estaríamos produciendo el mismo error que intentábamos evitar con esta solución, por lo que se decidió cambiar esta forma de recuperar el desfase horario siendo ahora un campo configurado por el propio usuario. De esta forma se consiguió ser coherente con las horas a mostrar ya no depende del usuario sino de la información que nosotros tenemos almacenada sobre él.

5.4.3.2.- Permisos de usuario

Cuando un usuario accede al sistema, se comprueba que el usuario tenga acceso a cualquiera de los dispositivos que tiene el cliente asignado, en el caso, de que un usuario no tuviera acceso a ningún dispositivo, no podrá acceder a la aplicación mostrando así un mensaje acorde a la situación del usuario.

Como hemos visto anteriormente, nos encontramos con 3 tipos de usuarios:

- El usuario supervisor, que es aquel que tiene acceso completo a toda la aplicación web pública. Este usuario puede realizar todas las acciones disponibles y visionar cualquier contenido que pertenezca a su cliente.
- Los usuario restrictivos. Aquí nos encontramos con 2 tipos de permisos como son el de lectura y el de escritura. Estos permisos pueden estar asociado a un grupo, como en este caso es BeeTracker, a un asentamiento o a un dispositivo. Para que un usuario restrictivo tenga acceso a la aplicación debe tener permisos de grupo o de asentamiento, no así de dispositivo ya que no se contempla en las especificaciones del sistema. Los permisos son heredables y si no se especifica nada de un nivel inferior se deben coger los del nivel superior manteniendo este orden, grupo – asentamiento – dispositivo.

Por todo esto, nos podemos encontrar con un mismo usuario con permisos de escritura sobre un dispositivo mientras que en otro dispositivo solo tenga permisos de lectura, o incluso que no tenga acceso a algún otro dispositivo.

- Usuario con permisos de escritura: El usuario que tenga acceso a nuestra aplicación y tenga permisos de escritura sobre un dispositivo podrá realizar acciones que modifiquen los estados del BeeTracker.
- Usuario con permisos de lectura: El usuario que tenga acceso a nuestra aplicación y tenga permisos de lectura sobre un dispositivo no podrá realizar acciones que modifiquen los estados del BeeTracker, pero si visionar su contenido.

Una vez el usuario ha sido validado y se comprueba que tiene permiso para acceder a la aplicación web, se muestra el escritorio principal de nuestra aplicación donde el usuario podrá observar de una manera rápida y sencilla el estado de todos los dispositivos a los cuales tiene acceso.

En cada una de las pantallas de nuestra aplicación se realiza la validación de que el usuario tenga permisos suficientes para poder acceder, así como si a los datos que está accediendo también le son permitidos visionarlos. Un ejemplo de lo que se acaba de exponer, es que si un usuario no es supervisor no tiene que tener acceso a la parte de informes de la aplicación por lo que si el usuario intenta acceder escribiendo la dirección, este se verá redireccionado al escritorio de la aplicación. De este modo, si un usuario no tiene permisos para ver un dispositivos pero intenta acceder a él de forma directamente mediante una URL, se le redireccionará al listado de BeeTrackers. Más adelante, en el apartado de seguridad volveremos sobre estos puntos completando con las otras medidas de seguridad existentes en el sistema.

5.4.3.3.- Sección BeeTrackers

La primera sección importante a abarcar en este proyecto, es el apartado de BeeTrackers. En esta sección los ficheros

- beetrackers.php
- beetrackers_ajax.php
- beetrackers_js.js
- Librería XAJAX

En el se realiza un listado de todos los dispositivos a los que el usuario solicitante tiene acceso, ya sea de lectura o de escritura, mostrando información relativa al dispositivo, como son los datos de número de serie y nombre de dispositivo, el tipo de servicio del dispositivo, el estado actual en el que se encuentra, el asentamiento al que pertenece, la última ubicación y cobertura transmitida por el dispositivo.

LISTADO DE BEE TRACKERS							
Número de Serie▼	Nombre del dispositivo	Tipo	Estado	Asentamiento	Localización	Cobertura	Gráficas
BT0000002	BT_TEST	Básico	⚠ Alarma	Pruebas LEPTON	LAT. --- LONG. ---	Regular	
antesBT7	Pruebas Pastrana	Básico	💡 Apagado	Pruebas LEPTON	LAT. 39.4656 - LONG. -0.387092	Regular	
antesBT2	BT laboratorio	Básico	💡 Apagado	Pruebas LEPTON	LAT. 39.474 - LONG. -0.36312	Buena	
antesBT5	Pruebas Ricardo	🌟 Premium	💡 Apagado	Pruebas LEPTON	LAT. 39.4659 - LONG. -0.385772	Regular	

Listado de BeeTrackers

Este listado está constantemente actualizando la información mostrada para dar la sensación de tiempo real, ya que en si es una pantalla de monitorización para conocer cual es el estado de todos los dispositivos del cliente. Esto hace que sea un sistema vivo de forma automática y sin que el usuario tenga que realizar ninguna acción.

Cuando en este listado nos encontramos con un dispositivo en alarma, los datos de posición que nos va mostrando son los referentes al seguimiento que se está produciendo, esto hace que estemos al conocimiento de que el dispositivo está en movimiento y se está produciendo una alarma en el mismo.

Como es posible que este listado contenga bastantes dispositivos, la pantalla está dotada de un filtro mediante el cual el usuario puede concretar los dispositivos que desea visualizar.

FILTRO

Nombre: Tipo: Estado: Beetracker: Asentamiento:

Filtrado de BeeTrackers

Este filtro, llevo a lugar a un conflicto para la visualización del listado. Como hemos dicho anteriormente el listado está constantemente actualizando, mediante una función XAJAX, por lo que si procediéramos a realizar una nueva llamada para este filtro, podríamos llegar a mostrar los dispositivos de manera incorrecta y llevar a confusión al usuario. Esto se solvento mediante variables globales en el código de Javascript, haciendo que tanto para la actualización automática del listado y el filtrado hagan uso de la misma función. Esto hace que desde que el usuario realiza el filtrado, hasta que actualiza pueden pasar 2 ciclos de actualización, el que actualmente está en curso y el que realiza la actualización.

5.4.3.4- Sección BeeTracker

Esta es una de la secciones que más ha costado su desarrollo por la cantidad de información diferente que muestra y todas las acciones que el usuario puede efectuar en ella.

Los ficheros que utilizan:

- Beetracker.php
- lib/ajax/Beetracker_ajax.php
- lib/js/Beetracker_js.js
- css/Beetracker_print.css
- generarCSV.php
- generarPDF.php
- librería flot (para las gráficas)
- Librería XAJAX
- Librería tcpdf

La estructura del contenido de esta página está estructurado en seis apartados, para diferenciar el tipo de información que se muestra en cada una de ellas.

BT0000001 | LAT. 38.0399 | LONG. -2.02625 | AARM - COAG

Estado



Última conexión: miércoles 7 de septiembre de 2011 a las 02:30:30

Estado de BeeTracker

En la primera de ellas nos muestra los últimos datos enviados por el dispositivos relativos a su posición mediante la latitud y longitud, el nombre del BeeTracker, el asentamiento al que pertenece y al estado en que se encuentra, esto el gracias a que según el estado en el que se encuentre irá cambiando de color a rojo, en caso de alarma de seguimiento, amarillo, en caso de alarma técnica, blanco, si el dispositivo se encuentra apagado y verde si todo se encuentra correcto.

En la siguiente sección denominada estado, nos enseña un pequeño resumen sobre el estado actual del dispositivo ya sea encendido, apagado, en alarma, o cualquier estado intermedio en el que se pueda encontrar. Desde este punto, y siempre que nuestro dispositivo se encuentre en estado encendido podremos apagar el BeeTracker de forma remota. Este apagado, no se produce al instante sino que lo que se solicita es un apagado y hasta que no vuelve a conectar el dispositivo con su servidor de configuración su estado no cambiará. También nos muestra la última conexión que realizó el BeeTracker en hora local.

Los dos campos siguientes, el de deshabilitar dispositivo y el de avisos, son los que nos permiten la configuración del dispositivo BeeTracker.

Deshabilitar Beetracker BT0000002

Deshabilitar Beetracker BT0000002 Hora: : Duracion: en horas

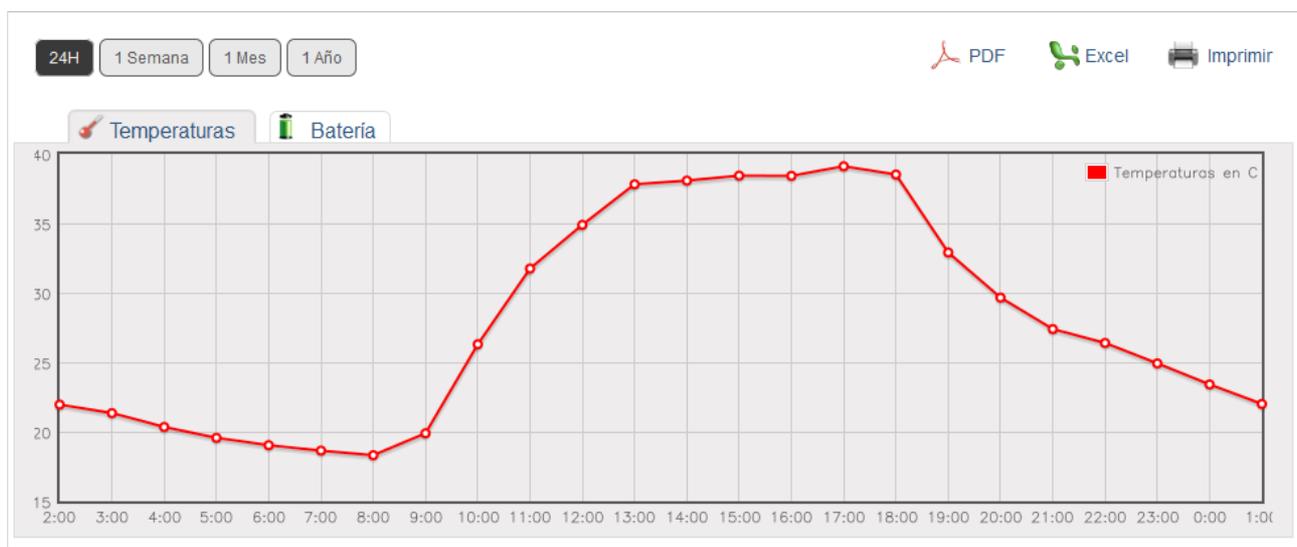
Deshabilitación de BeeTracker

También y como hemos dicho anteriormente el usuario con los permisos suficientes podrá configurar los parámetros para alarmas de un dispositivos de manera muy fácil e intuitiva mediante un sencillo formulario. El único punto comprometido de este formulario, es que los datos que son actualizados por el usuario, no se ven reflejados en el dispositivo BeeTracker hasta que este realice una nueva conexión con el CPD, ya que por requisitos de ahorro de batería es el propio dispositivo el que realiza las conexiones con el CPD.

El punto más comprometedor de está sección es el de las gráficas por la cantidad de información que pueden llegar a mostrar. La librería utilizada para mostrar estas gráficas es Flot. Es una librería realizada en Javascript por lo que es el navegador web el que se encarga de mostrarlas. Uno de los primeros problemas que nos encontramos para poder mostrar las gráficas fue el sistema de horas, como en explicado en el punto anterior, todas las horas que están almacenadas son UTC por lo que al intentar actualizarlas con los desfases las gráficas no se dibujaban correctamente. El problema se encontraba en la forma de calcular cada una de las horas del día que hacíamos al recuperar los valores a mostrar, ya que la librería Flot los trataba siempre como si fueran horas UTC aunque se les actualizara el desfase horario. La solución se encontró modificando las horas que tenia un día a las cuales se le actualizaba el desfase horario por lo que al pintar la gráfica eran horas UTC actualizadas.

Debido a esta librería, es por lo que encontramos una de las restricciones sobre el uso de los navegadores y se recomienda el uso de todos los exploradores excepto las versiones de Internet Explorer anteriores al 8, ya que estas gráficas en estos navegadores no son pintadas de forma correcta y es una parte importante de nuestra aplicación.

Otros de los inconvenientes que surgieron a posterior, cuando se implementó la funcionalidad de mostrar las gráficas en PDF era la imposibilidad de algunos navegadores de transformar el objeto canvas que contenía la gráfica en una imagen para poder insertarla a posterior en un fichero PDF. Esta imposibilidad se debía a que los navegadores que no contenían la especificación del objeto canvas y no sabían como tratarlos, ya la forma de conseguir mostrar estas gráficas generadas por el explorador web del usuario no existe una forma de convertir este objeto a imagen. Este problema ya no ocurre en los más novedosos, por lo que para evitar complicaciones en los navegadores que no son capaces de soportar dicho objeto canvas se muestra un mensaje de información.



Grafica de temperatura de BeeTracker

Además los usuarios pueden almacenar también dicha información en formato CSV o imprimirlas directamente.

Por último, nos encontramos información relativa a los últimos eventos producidos por el dispositivo que estamos viendo indicándonos cual es el estado de cada incidencia.

5.4.3.5.- Sección alarmas

La sección de Alarmas es aquella en la que mostramos la información de relativa a las distintas alarmas que produce un BeeTracker.

Los ficheros utilizados para esta sección son:

- Alarmas.php / historico.php
- Alarmas_ajax.php / historico_ajax.php
- Alarmas_js.js / historico_js.js
- eshapes.js
- Librería flot (para las gráficas)

- Librería XAJAX
- API Google Maps

Dentro de este apartado podemos tratar de forma simular las alarmas activas en el sistema como las alarmas que ya han sido cerradas, ya que tanto los listados que mostramos como la forma de mostrar la información son idénticas, la única diferencia es que mientras en las primeras nos encontramos con información viva en el sistema, está constantemente actualizándose, en los segundos esta información no puede ser modificada.

LISTADO DE ALARMAS

⚠ | 5 Alarmas en total 🛠 | 4 Alarmas técnicas 🚔 | 1 Alarmas por robo

Referencia	Tipo de alarma	Número de Série	Asentamiento	Fecha inicio	Cerrar	Atender
00001571	Robo de dispositivo	BT0000002	Pruebas LEPTON	2011-08-22 14:04:32	✖	➡
00001556	Bateria baja	BT0000002	Pruebas LEPTON	2011-07-23 03:06:10	✖	➡
00001555	Bateria baja	BT0000002	Pruebas LEPTON	2011-07-23 02:35:30	✖	➡
00001554	Bateria baja	BT0000002	Pruebas LEPTON	2011-07-23 02:02:35	✖	➡
00001553	Bateria baja	BT0000002	Pruebas LEPTON	2011-07-23 01:44:24	✖	➡

Listado de alarmas activas

Nos podemos encontrar con seis tipos de alarmas divididas en dos grandes grupos, debido a la forma de implementación y de información que se muestra: alarmas por robo y alarmas técnicas.

En las alarmas por seguimiento, la gran dificultad que surgió durante su implementación fue el hecho de que si el dispositivo no nos enviaba posición en su última trasmisión, debido a la falta de cobertura o debido a algún error, no encontrábamos una posición para mostrar por lo que para un primer vistazo de la información no sabíamos donde se encontraba. La solución se encontraba en toda la información referente a la cobertura que el dispositivo nos enviaba en cada una de sus transmisiones, gracias a esta información podíamos obtener una posición aproximada en la que se encontraba el dispositivo y darle dicha información al usuario de forma preliminar. En la siguiente apartado se profundizará en este tipo de alarmas, dando más detalles de como se obtiene dicha solución.

✖ Alarma: 00001571 | Tipo: Robo de dispositivo | Fecha: 2011-08-22 14:04:32
 BeeTracker: BT0000002 | Asentamiento: Pruebas LEPTON + Información Cerrar Seguimiento



📍 Centrar 🌐 Mapa

🔍 Zoom + 🔍 Zoom -

📍 LAT.: 39.4632012 | LONG.: -0.386447 | Radio: 621m

🚶 -- Km/h

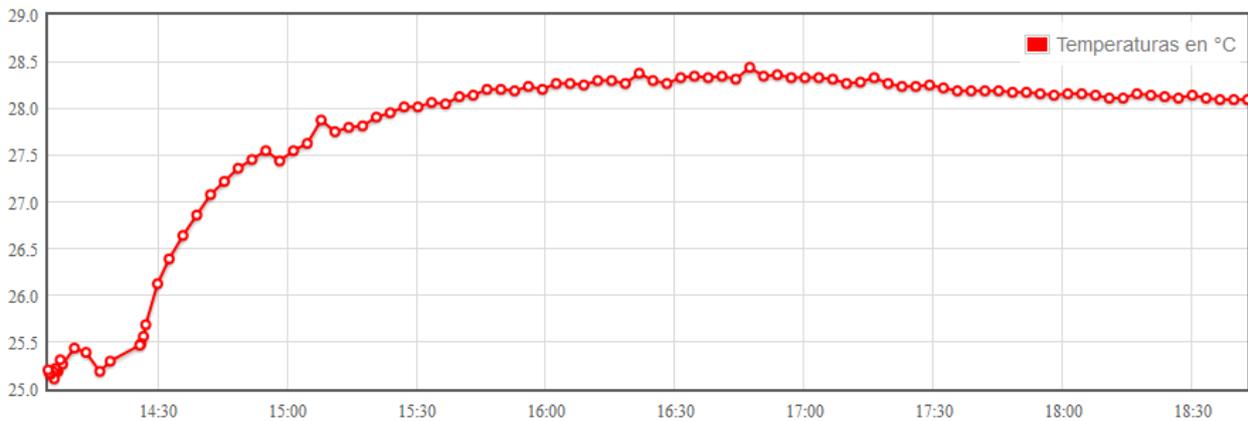
🌡 Regular 28.0938 °C

🔋 94.53733 %

+ Información

Información alarma de seguimiento

Aparte de mostrar información relativa a la última posición que ha sido transmitida por el dispositivo BeeTracker también podemos visionar gracias a las pestañas laterales, todos los datos recogidos por cada uno de los parámetros como temperatura, humedad, batería y velocidad.



Gráfica de temperatura de alarma de seguimiento

Por último dentro de este resumen sobre una alarma de seguimiento tendremos la posibilidad de acceder a la visión de un seguimiento en tiempo real más detallado en el que podremos observar cual ha sido el recorrido que ha realizado el dispositivo desde que saltó la alarma hasta el momento actual.

Los otros tipos de alarmas, las denominadas técnicas, tiene la gran dificultad de procesar la información que hemos recibido. Estas alarmas únicamente nos muestran una información concreta de los datos que han producido la alarma. En estas alarmas, la posición del dispositivo ya no es lo primordial, pero si el dato de posición ha sido transmitido se mostrará el lugar en el que está ubicado el dispositivo en el momento de la alarma, sino se mostrará un mensaje acorde al problema de no localización del dispositivo.

✕ Alarma: 00001489 | Tipo: Temperatura | Fecha: 2011-06-25 02:30:31
 BeeTracker: BT0000001 | Asentamiento: AARM - COAG Excel



Centrar Mapa

Zoom + Zoom -

LAT. : 38.1199 | LONG. : -2.0707

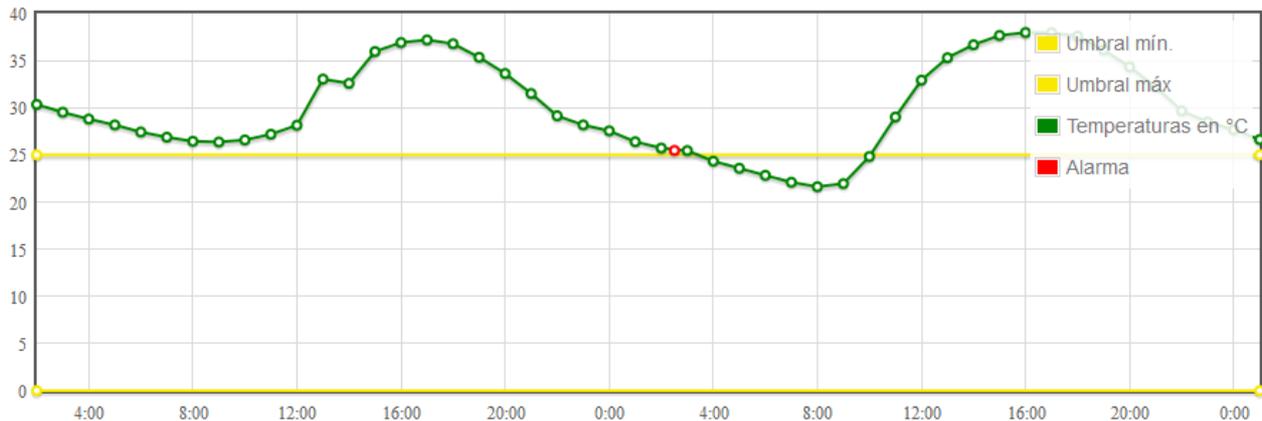
25.4688 °C

59.0564 %

Imágenes ©2011 - Términos de uso

Información alarma técnica de temperatura

Al usuario una información tan concreta puede que no le resulte fácil de procesar, para ello se muestra una gráfica con la información que ha producido la alarma más una pequeña evolución de las últimas 48 horas para que se compruebe de forma más concreta cuales han sido los motivos de dicha alarma.



Gráfica de temperatura de alarma técnica

5.4.3.6.- Sección seguimiento

Los ficheros que son utilizados en esta sección son los siguientes:

- seguimiento.php
- seguimiento_ajax.php
- seguimiento_js.js
- eshapes.js
- Librería flot (para las gráficas)
- Librería XAJAX
- API Google Maps

Esta es la sección más crítica de toda la aplicación, ya que debido a que lo que nos muestran es la información relativa a un alarma por robo. Al tratarse de una incidencia grave debe ser lo más precisa, rápida e intuitiva dándole una gran importancia a la posición en la que se encuentra el dispositivo en todo momento.

Para poder ver el recorrido que ha realizado el dispositivo se ha decidido utilizar la API de Google Maps en su versión 2. Gracias a esta API y a todos los datos que transmite el dispositivo BeeTracker mientras se encuentra en estado de alarma de seguimiento, se puede mostrar de forma muy precisa la posición del dispositivo en todo momento.

Como hemos comentado, la gran importancia de esta parte de la aplicación es el mostrar la posición exacta del dispositivo en todo momento. Esto nos lleva a que el dispositivo por motivos de cobertura GPS puede no transmitir una ubicación en alguno de las transmisiones que realiza.

Esto significa que el dispositivo puede estar durante cierto tiempo de forma continua sin mostrar ninguna posición lo que es un grave problema tanto para el objetivo del dispositivo BeeTracker como para nuestra aplicación web. Para solucionar este inconveniente, en las tramas que transmite el dispositivo nos da información de las coberturas de las antenas de telefonía móvil a las cuales puede acceder, también conocido como posición AGPS, y nosotros con esta información podemos obtener la posición aproximada en la que se encuentra el dispositivo.

Para obtener los datos de posición a partir de los datos de la cobertura móvil, debemos realizar una petición POST sobre la URL <http://www.google.com/loc/json> como la que se muestra a continuación:

```
{
  "version": "1.1.0",
  "host": "maps.google.com",
  "radio_type": "gsm",
  "request_address": true,
  "cell_towers": [{
    "cell_id": "42",
    "location_area_code": 415,
    "mobile_country_code": 310,
    "mobile_network_code": 410,
    "age": 0,
    "signal_strength": -60
  }]
}
```

Como respuesta obtendremos un mensaje como el siguiente:

```
{
  "location": {
    "latitude": 51.0,
    "longitude": -0.1,
    "altitude": 30.1,
    "accuracy": 1200.1,
    "altitude_accuracy": 10.1,
    "address": {
      "street_number": "100",
      "street": "Amphibian Walkway",
      "postal_code": "94043",
      "city": "Mountain View",
      "county": "Mountain View County",
      "region": "California",
      "country": "United States of America",
      "country_code": "US"
    }
  }
}
```

En esta respuesta que recibimos de la petición POST realizada obtenemos cual es la latitud, longitud y el radio en el que se encuentra nuestro dispositivo por lo que podemos seguir mostrando cual es la posición de nuestro dispositivo en todo momento.

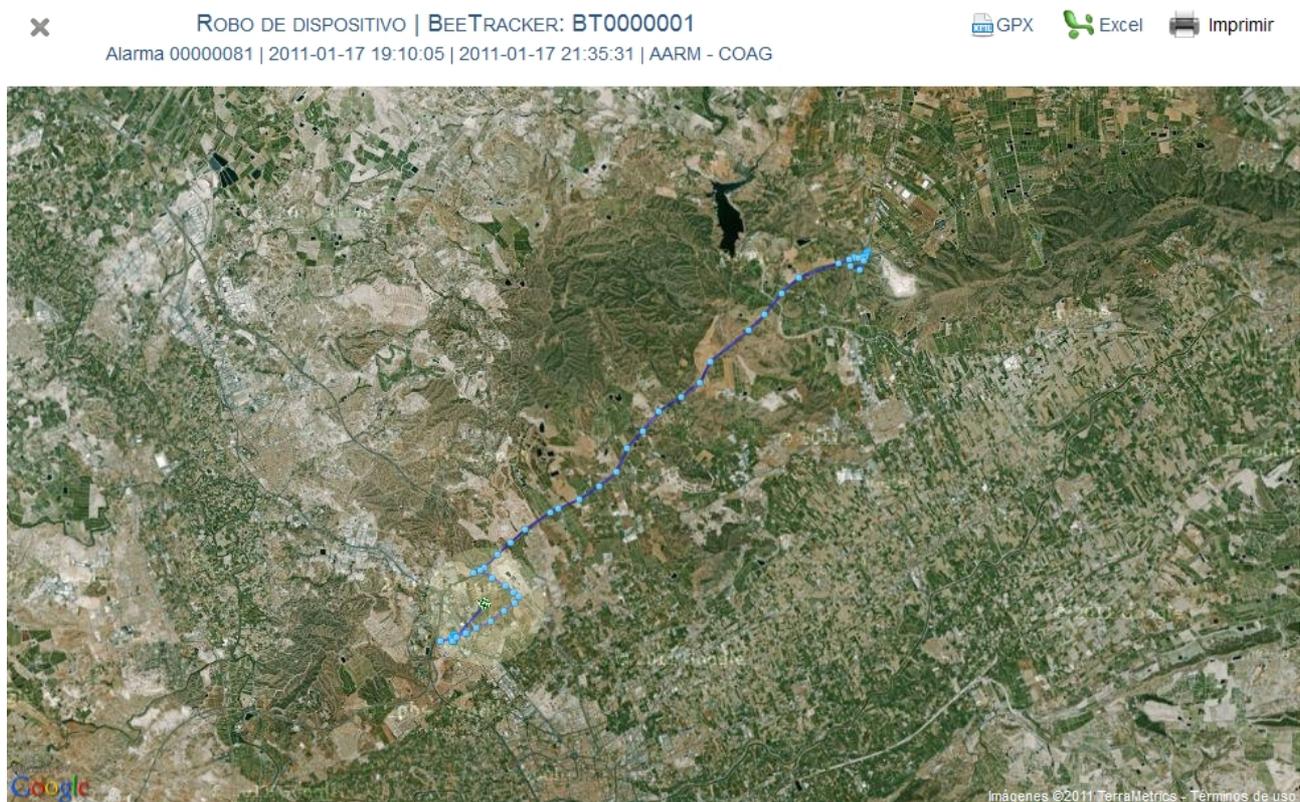
5.4.3.7.- Sección seguimiento histórico

En esta sección los ficheros utilizados son:

- h_seguimiento.php
- h_seguimiento_ajax.php
- h_seguimiento_js.js
- eshapes.js
- Librería flot (para las gráficas)
- Librería XAJAX
- API Google Maps

Aunque con anterioridad, habíamos englobado los listados históricos y los activos en el sistema debemos de hacer una pequeña referencia a los seguimientos históricos ya que la forma de mostrar la información si que tienen diferencia con los activos.

En esta sección de nuestra aplicación web lo más importante es sacar la ruta completa que ha seguido nuestro dispositivo a lo largo de la duración de nuestra alarma de seguimiento, dejando de lado la cantidad de punto a mostrar dando prioridad al recorrido. Esto significa que a la hora de mostrar, un mismo punto únicamente se mostrará la primera vez y se desechará el resto de ocasiones que aparezca.



Mapa de seguimiento histórico

Otra de las importancias ha resaltar en estos seguimientos es la posibilidad de que tanto la primera trama que ha transmitido el dispositivo como la última, no tengan posicionamiento GPS por lo que se recurrirá a la posición AGPS, por cobertura de telefonía móvil, para la obtención de una posición GPS explicado en el punto anterior.

	2011-01-17 20:15:40	38.0171	-1.16173	1.27788	19.2656	47.64954	Buena
	2011-01-17 20:15:23	38.0171	-1.16176	3.66696	19.2656	47.64954	Buena
	2011-01-17 20:15:06	38.0173	-1.16193	0.2778	19.2812	47.2285	Buena
	2011-01-17 20:14:48	38.0172	-1.16193	4.12996	19.2656	47.64954	Buena
	2011-01-17 20:14:30	38.0165	-1.16176	1.92608	19.2812	47.64954	Buena
	2011-01-17 20:14:11	---	---	---	19.2812	47.64954	Buena
	2011-01-17 20:13:54	---	---	---	19.25	46.80469	Buena
	2011-01-17 20:13:35	---	---	---	19.2969	47.64954	Buena
	2011-01-17 20:13:17	---	---	---	19.2656	47.64954	Ninguna
	2011-01-17 20:12:59	---	---	---	19.25	47.64954	Buena

Listado de puntos de seguimiento histórico

También nos encontramos con un listado, en el que tendremos todas las tramas transmitidas por el dispositivo y en la que pondremos situar en nuestra mapa cada uno de los posiciones que nos está indicando en trama, tanto con GPS como con AGPS.

5.4.3.8.- Sección visión global

Los ficheros que implementan dicha sección son:

- Localizacion.php
- Localizacion_ajax.php
- Localizacion_js.php
- eshapes.js
- Librería XAJAX
- API Google Maps

La visión global nos permite de ver de forma muy rápida cual es la localización de todos nuestros dispositivos. Aquí nos vamos a encontrar con la clasificación de los dispositivos en 3 posibles grupos: localizados, no localizados y apagados.

Localizados, son aquellos dispositivos que están activos y de los cuales tenemos una posición GPS concreta. Esta posición puede venir dada por la información transmitida diariamente por el dispositivos o mediante la posición que se transmite durante un seguimiento. En este caso se mostrará su posición mediante un símbolo de peligro, para advertirnos de que este dispositivo se encuentra en una alarma de seguimiento.

Los no localizados, son aquellos dispositivos que se encuentran encendidos pero no tenemos una posición GPS concreta.

Por último tenemos los dispositivos apagados, de los cuales se muestra la última posición conocida del dispositivo.

5.4.3.9.- Sección informes

Los ficheros necesarios para la implementación de esta sección son:

- Informes.php
- Informes_ajax.php
- Informes_js.php
- ver_facturas.php / ver_informes.php
- generarGPX.php
- generar_mensual_CSV.php
- generar_seg_CSV.php
- Librería XAJAX

Esta sección, la cual no conlleva una gran dificultad en cuanto a implementación, sino por lo que cabe ser comentada es como se obtienen los contenidos que se muestran.

Un dato importante a remarcar en esta sección, es que a ella únicamente tiene acceso aquellos usuarios que tiene permisos de supervisor en el sistema, toda aquel que intente acceder al mismo y no tenga dichos permisos serán redirigidos al escritorio de nuestra aplicación.

INFORMES

🚨 | 10 (9) Informes de Alarmas
📊 | 15 (11) Informes de Telemedidas
📄 | 0 (0) Facturas
📅 | 15 (14) Informes de registro

🚨 10 Informes de Alarmas (9 sin leer) ▾

Referencia:
 Beetracker: Todos ▾
 Asentamiento: Todos ▾
 Estado: Todos ▾

Fecha inicio: Posterior ▾ a / /
🗑️
 Fecha fin: Posterior ▾ a / /
🗑️
✕ ✓

Ref.	Descripcion	Fecha alta	Fecha fin	Dispositivo	Asentamiento	
1399	Robo de dispositivo	2011-06-07 16:19:23	2011-06-07 16:30:06	BT0000001	AARM - COAG	
1423	Robo de dispositivo	2011-06-24 12:13:27	2011-06-24 12:45:10	BT0000001	AARM - COAG	
1560	Robo de dispositivo	2011-08-03 13:04:11	2011-08-03 13:50:26	BT0000001	AARM - COAG	
1573	Robo de dispositivo	2011-08-23 10:28:14	2011-08-23 11:46:41	BT0000001	AARM - COAG	

Listado de informes

Los informes que se muestran aquí han sido obtenidas a partir de aplicaciones desarrolladas bajo Java, que aunque no son parte visible de la aplicación web, si que entran dentro de nuestro proyecto de BeeTracker e dependen de una forma muy directa con nuestra aplicación web.

Estos informes se generan automáticamente durante los periodos establecidos mediante el cron de nuestro CPD. Así nos encontramos que todos los días a las 00.00 se generarán los informes de alarmas de seguimiento que el cierre de los mismos se hayan generado el día anterior. Los días 2 de cada mes se generarán los informes de registro de usuario y acciones realizadas por cada uno de ellos. El día 3 de cada mes se generaran los informes de todas las telemedidas recogidas por todos los dispositivos activos en el sistema durante el mes anterior. Y por último, el día 4 de cada mes se procederá a la generación de las facturas de los clientes del periodo perteneciente al mes anterior.

A continuación, detallamos cada uno de los posibles informes generados dentro de nuestra aplicación BeeTracker dando una pequeña explicación de como se generan, y en caso de fallo que posibles soluciones nos encontramos:

- **Informes de alarmas de seguimiento:** Son aquellos informes que nos muestra la información resumida de todas las alarmas de seguimiento que han sido cerradas en el sistema. Como hemos dicho anteriormente, para generar estos informes se ejecuta un programa en Java todos los días mediante el cual se generan informes de todas aquellas alarmas de seguimiento que su cierre se haya efectuado el día anterior. En estos informes, se muestra un pequeño mapa mostrando el punto inicial del recorrido y el punto final, y un listado con la información de todos los puntos con posición GPS obtenidos por el dispositivo en el seguimiento. También se muestra los datos de todas las medidas recogidos durante el mismo recorrido.

Estos informes se han decidido generar mediante una aplicación Java debido a que evitamos mucha carga al sistema, ya que no genera cada vez que se desee consultar el informe, sino que se genera en las horas en las que el CPD tiene menos carga de procesamiento y su consulta resulta más rápida, ya que minimiza las consultas a la Base de Datos.

En el caso de que se produzca un fallo en este proceso, se daría aviso al administrador del sistema informándole del error mediante un correo electrónico, y el mismo administrador podría volver a lanzar el proceso para todos los dispositivos o generar únicamente un informe de una alarma en concreto.

- **Informes de registro:** Estos informes recogen todas las conexiones que ha realizado todos los usuarios de un cliente. En el se detalla todos los accesos realizados por los usuarios y así como las acciones que han realizado en nuestra aplicación web. En el se detalla los días, horas de acceso y desde que dirección IP se ha realizado el acceso.

En cuanto las acciones de los usuarios, nos informan sobre que dispositivos BeeTracker se han producido y dando datos muy concretos de las acciones que se han realizado sobre cada uno de ellos, para así tener una completa información de los que está ocurriendo sobre ellos.

Estos informes se generan de forma automática el día 2 de cada mes, durante los momentos de menos carga del sistema. Al igual que en los informes de alarma de seguimiento, en caso, de que durante la ejecución de esta aplicación Java se produjera un error, se informaría del mismo mediante un correo electrónico al administrador del sistema. El administrador siempre que lo desee, podrá volver a ejecutar el proceso de generación de informes de registro tanto mensual como únicamente de un cliente en concreto.

- **Informes de telemedidas:** Este es uno de los informes más útiles que se pueden encontrar en nuestra aplicación, ya que lo que obtenemos es un resumen mensual de todos los datos de telemedidas recogidos por cada uno de los dispositivos BeeTrackers.

En estos informes se pueden ver de forma muy sencilla las variaciones diarias de las temperaturas recogidas, desde la temperatura media, la mínima y la máxima tanto de forma gráfica como mediante un listado diario con los datos. En el mismo informe siempre nos podemos encontrar información de la variación de la batería tanto en forma de gráfica como en un listado del mes anterior.

Adicionalmente, y si el dispositivo está capacitado para ello también nos podremos encontrar con gráficas y listados con información relativa a la humedad que detecta el sistema, el sonido que se produce dentro de la colmena y el peso que está adquiriendo a lo largo del tiempo.

Como se ha dicho anteriormente, estos informes se generar de forma automática todos los días 3 de cada mes en las horas de menos carga del sistema. En el caso de que se produzca un error durante la ejecución de la aplicación, se procederá a informar del mismo al administrador del sistema mediante un correo electrónico. El administrador siempre que lo desee, podrá volver a ejecutar el proceso de generación de informes de telemidas tanto mensual como únicamente de un cliente en concreto.

- **Facturación:** Las facturas son los informes en los que cada usuario se les detalla el consumo desglosado por dispositivos de cada mes en los que ha estado activo en el sistema. Para poder realizar estas facturas, se recurren a apuntes contables de cada uno los clientes. En ellos encontramos a que dispositivo hacen referencia y que tarifa tiene asignada el mismo, por lo que la facturación es la recogida de forma organizada de cada uno de estos datos.

Todos los importes que nos encontramos en los apuntes, nos los encontramos sin IVA por lo que en este proceso deberemos realizar el sumatorio de todo lo que se le debe facturar al usuario y posteriormente se realizará el cálculo del importe total a pagar a partir del IVA actual.

Al igual que en los casos anteriores, este proceso se ejecuta una vez al mes, en concreto el día 4 de cada mes en el momento de menos carga computacional del sistema. Si durante el proceso de ejecución de esta aplicación Java se produjera un error, se procedería a informar al administrador del mismo mediante un correo electrónico.

5.4.3.10.- Sección configuración

En esta sección de la aplicación los ficheros que la implementan son:

- config.php
- config_ajax.php
- config_js.js
- Librería XAJAX

Al igual que en la sección anterior, no es una de las secciones más complicadas de realizar su implementación.

Dependiendo del tipo de permisos de usuarios que nos encontremos, la información que se muestra en esta sección será distinto.

Si el usuario no es supervisor, la información que visionará es sus datos personales con la posibilidad de poder editarlos, poder configurar la franja horaria que el usuario desee todas las horas del sistema, ya que como hemos dicho anteriormente en nuestro sistemas todas las horas se encuentra almacenadas en UTC, y por último se le da la posibilidad al usuario de poder realizar un cambio de contraseña.

Lo más complejo que nos encontramos en esta sección, es la de mantener datos coherentes en todo momento, es decir, que nadie pueda tener acceso simultaneo a la misma información en ningún momento. Para evitar esto, se realiza un bloqueo en información crítica, como es la personal. Más adelante, en el apartado de seguridad volveremos a mencionar dando una descripción concreta de como se realiza este proceso de bloqueo.

En caso contrario, de que el usuario sea supervisor, también tendrá acceso a poder visionar los datos de todos los usuarios de su cliente, así como ver que tipo de permisos tiene cada uno de los usuarios. También podrá editar los datos del cliente, y al igual que acabamos de comentar, esta información se puede considerar crítica por lo que es necesario que se realice un bloqueo sobre la misma para que 2 usuarios no puedan estar accediendo a la vez.

5.4.3.11.- Sección de ayuda en línea

En nuestra última sección necesitaremos los siguientes ficheros:

- crear_incidencia.php
- help.php
- saveIncidencia.php
- ver_documento.php
- Librería XAJAX

Como ya se ha comentado en anteriores apartados, el objetivo de la ayuda en línea es la de crear un sistema sencillo con la que cualquier usuario tenga acceso para poder realizar consultas a los administradores. Es una herramienta que guarda similitud con un foro, siendo este privado entre los usuarios y los administradores del sistema, siendo el usuario supervisor de cada cliente, el único capaz de poder visionar todos los mensajes publicados por los usuarios no supervisor del mismo cliente.



Incidencia	Fecha	Asunto	Estado	Ver
75	2011-05-25 11:02:50	Prueba	Cerrada	→
78	2011-05-25 11:08:57	Prueba 2	Cerrada	→
79	2011-05-25 11:11:21	Prueba 3	Cerrada	→
80	2011-05-30 12:52:51	Prueba 4	Cerrada	→

Listado de incidencias de ayuda en línea

Todos los usuario pueden crear de una forma fácil una incidencia, ya que únicamente es necesario poner un titulo y descripción de la incidencia que se desea consultar. Además se da la posibilidad de que el usuario pueda adjuntar diversos tipos de documentos para completar la consulta a realizar. Es aquí donde surgió uno de los problemas que se tuvo que solucionar con las configuraciones tanto de MySQL como de PHP como del servidor Apache, ya que estos tenían un limite de ficheros que pueden aceptar. Este limite tuvo que ser modificado para que no ocasionara problemas a los usuarios que desearan utilizar esta herramienta.

CREAR INCIDENCIA

Cliente
LEPTON Ingenieros S.L.

Usuario
Alberto

Resumen incidencia

Descripción

Subir fichero

Prioridad

Crear incidencia de ayuda en línea

Uno de los detalles que ha destacar de esta sección de nuestra aplicación Web BeeTracker es que automáticamente se abre en una ventana auxiliar, facilitando así la posibilidad de que el usuario este trabajando y consultar o realizar al mismo tiempo las incidencias. Este proceso también ocasionó problemas ya que al poder abrir esta sección desde todas las otras secciones de nuestra aplicación, podría producirse el problema de abrir en multitud de ocasiones la misma ventana. Para ello, se ha procedido a generar una Cookie en la que se mantiene la información de si la ventana ha sido abierta o no. En caso de que la ventana ya haya sido abierta, se informa al usuario de que existe una ventana en uso en la actualidad.

Por último, el usuario al consultar las incidencias podrá realizar nuevos comentarios sobre las mismas y mantener una comunicación abierta con los administradores para así dar a conocer si el problema persiste o si ya se le ha dado solución.

Listado Incidencia 75 ✖

Incidencia: 75

Asunto: Prueba

Estado: Cerrada

Fecha: 2011-05-25 11:02:50

Abierta por: Alberto Sánchez Gómez

Adjunto  Informe de incidencia 1274.pdf

 De Alberto Sánchez Gómez a LEPTON-Admin - 2011-05-25 11:03:47
Esto es una prueba

 De LEPTON-Admin a Alberto Sánchez Gómez - 2011-05-25 11:04:46
En que puedo ayudarte

Detalle incidencia de ayuda en línea

5.5.- Seguridad de la aplicación

Una de los apartados más importantes de nuestra aplicación es la de la seguridad, ya que al tratarse de una aplicación mediante registro es importante evitar intrusiones en nuestro sistema, así como evitar que se muestre información a la que los usuarios no tienen acceso. De todo esto se encargan los siguientes ficheros situados en el directorio raíz:

- AccessControl/Auth/Inst_val.php
- AccessControl/Auth/Auth.php
- AccessControl/Session/Session.php
- AccessControl/CAC_module_menu.php
- AccessControl/CAC_module.php

5.5.1.- Acceso a la aplicación

Al tratarse de una aplicación privada para acceder a nuestra aplicación web BeeTracker es necesario que se registren los usuarios. Para ello, si no se ha iniciado ninguna sesión, aunque intentemos acceder a la aplicación mediante una URL en concreto, de inmediato seremos redireccionados a la página de login.

El registro se realiza mediante usuario y contraseña que son cifradas en la Base de Datos para evitar fallos de seguridad. Una vez, el usuario y contraseña han sido validados se generan variables de sesión, a partir de las cuales, en las siguientes páginas a las que acceda el usuario se comprobará que el usuario sigue siendo válido para navegar por nuestra aplicación. Entre las variables de sesión se almacena información como el nombre de usuario, el desfase horario, si es supervisor y el usuario y contraseña codificados mediante md5.

5.5.2.- Validación de usuario

Cada vez que se accede a una nueva página en nuestra aplicación, se comprueba que el usuario que está intentando acceder continúe con una sesión activa en el sistema. En caso de que no sea así, será redirigido nuevamente a la pantalla de login del sistema para que vuelva a introducir los datos de registro.

Una sesión se considera activa siempre y cuando la codificación del usuario y contraseña sean correctas y no haya sobrepasado el tiempo máximo de estancia en la aplicación.

Al principio, la aplicación únicamente comprobaba que la sesión estuviera activo al acceder a cualquier página, pero al utilizar en nuestra aplicación AJAX hacia que aún habiendo caducado la sesión hasta que no se cambiará de pantalla no se nos informara de que ya no eramos un usuario correcto en nuestra aplicación. Para ello, en cualquiera de las funciones AJAX implementadas en nuestra aplicación se está validando que el usuario siga activo evitando así situaciones que puedan hacer peligrar la integridad de la aplicación.

5.5.3.- Permisos de usuario

El siguiente paso en cualquiera de las páginas de nuestra aplicación es la validación de que el usuario que está solicitando que se le muestre una información, realmente tenga acceso a dicha información.

Al acceder a cualquier página se comprueba que tipo de usuario es el que está solicitando la información, es decir, si es supervisor, que como hemos dicho anteriormente dicha información está almacenada en variables de sesión para evitar continuas consultas a la Base de Datos. Si se comprueba que el usuario es supervisor, se mostrara siempre toda aquella información que se corresponda con el cliente del usuario que está mostrando la información. En caso contrario, se procederá a ejecutar las funciones del ficheros `Inst_val.php` para validar cuales son los dispositivos o instalaciones a las que tiene acceso el usuario para así poder mostrar dicha información.

Poniendo un ejemplo para este caso, supongamos que el usuario Devel tiene acceso al dispositivo con ID 20. Si accediera a la URL <https://leptoning.dnsalias.com/Beetracker?ID=20&Offset=0> se le debería de mostrar toda la información relativa a dicho dispositivo. En caso contrario, si intentamos acceder a un dispositivo que no sea de nuestro cliente o que no tengamos permisos para poder visualizarlo, seriamos redireccionados inmediatamente al listado de los dispositivos a los cuales si que tenemos acceso. Este ejemplo podríamos extenderlo a alarmas, visualización de documentos, visualización de informes y visualización de facturas.

5.5.4.- Bloqueo de información

Como hemos en el apartado de Sección configuración, hay cierta información que es necesaria que se mantenga consistente en la Base de Datos, es decir, que todos los usuarios deben de visualizar la información de forma coherente en todo momento. Para evitar que varios usuarios estén editando la misma información al mismo tiempo, se ha procedido a crear un bloqueo mediante la Base de Datos. Cuando un usuario accede a una información crítica, adquiere el recurso y este es desbloqueado cuando el usuario a finalizado con la edición del mismo o cuando hayan transcurrido 10 minutos desde el acceso al mismo.

Con este simple bloqueo, evitamos que si un usuario de nuestra aplicación está editando información crítica y un usuario de la Intranet intenta editar la misma información, no podrá hacerlo si no han transcurrido esos de 10 minutos establecidos o el recurso ha sido liberado por el usuario que lo había adquirido.

6.- PRUEBAS

En este apartado vamos a exponer cuales han sido las pruebas realizadas sobre nuestra aplicación web, los resultados obtenidos con cada una de ellas y las conclusiones finales que se han obtenido

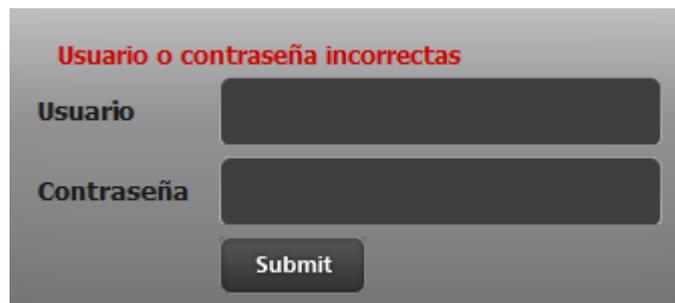
-
- 6.1.- Pruebas funcionales
 - 6.1.1.- Seguridad de acceso
 - 6.1.1.1.- Pruebas realizadas
 - 6.1.2.- Error de conexión a la Base de Datos
 - 6.1.2.1.- Pruebas realizadas
 - 6.1.3.- Bloqueo de información crítica
 - 6.1.3.1.- Pruebas realizadas
 - 6.1.4.- Acceso a múltiples servidores
 - 6.1.4.1.- Pruebas realizadas
 - 6.1.5.- Alarmas técnicas
 - 6.1.5.1.- Pruebas realizadas
 - 6.1.6.- Alarmas de seguimiento
 - 6.1.6.1.- Pruebas realizadas
 - 6.1.7.- Logs servidores
 - 6.1.7.1.- Pruebas realizadas
 - 6.1.8.- Ayuda en línea
 - 6.1.8.1.- Pruebas realizadas
 - 6.1.9.- Tareas automáticas de generación de informes
 - 6.1.9.1.- Informes de alarmas de seguimiento
 - 6.1.9.2.- Informes de registro
 - 6.1.9.3.- Informes de telemidas
 - 6.1.9.4.- Facturas
 - 6.1.10.- Proceso completo
 - 6.1.10.1.- Pruebas realizadas
 - 6.2.- Pruebas de validación
 - 6.2.1.- Test W3C
-

6.1.- Pruebas funcionales

6.1.1.- Seguridad de acceso

Para poder acceder a nuestra aplicación y como hemos comentado en los apartados anteriores, se necesita que el usuario se encuentre dado de alta en el sistema y para su acceso se le debe de haber facilitado por medio de los administradores de un usuario y una contraseña.

Si un usuario intenta acceder a nuestra aplicación y no ha introducido usuario y contraseña se le redirigirá automáticamente a la página de registro. Esto ocurrirá al igual que si intenta acceder con un usuario o contraseña no integradas en el sistema. Se informará al usuario de que se está produciendo un registro incorrecto.

Una captura de pantalla de un formulario de inicio de sesión con un fondo gris. En la parte superior, un mensaje de error en rojo dice "Usuario o contraseña incorrectas". Debajo, hay dos campos de entrada: "Usuario" y "Contraseña", ambos con barras de texto grises. Abajo de los campos hay un botón "Submit" con un fondo gris y texto blanco.

Acceso de usuario incorrecto

Si el usuario que accede a la aplicación está introduciendo un usuario correcto pero su contraseña es incorrecta, tendrá un máximo de 4 intentos consecutivos de poder acceder a la aplicación y en caso de que este último tampoco sea correcto se procederá a bloquear la cuenta que solo podrá ser desbloqueado por los administradores del sistema. Al igual que en el caso anterior, se le dará aviso al usuario de que la cuenta ha sido bloqueada.

Una captura de pantalla de un formulario de inicio de sesión con un fondo gris. En la parte superior, un mensaje de error en rojo dice "Usuario bloqueado. Contacte con el administrador". Debajo, hay dos campos de entrada: "Usuario" y "Contraseña", ambos con barras de texto grises. Abajo de los campos hay un botón "Submit" con un fondo gris y texto blanco.

Acceso de usuario bloqueado

Para evitar que el usuario bloquee la cuenta, se ha dado añadido la funcionalidad de poder recuperar una nueva contraseña mediante la introducción del correo electrónico que el usuario tiene en nuestro sistema. Una vez generada la nueva clave, en el primer acceso que se produzca a la aplicación se le solicitará que introduzca una nueva contraseña para así evitar posibles problemas de suplantación de identidad.

Una vez un usuario ha accedido a nuestra aplicación, tiene un tiempo máximo de estancia en la aplicación, si supera el tiempo de estancia, será redirigido automáticamente a la página de registro para que vuelva a autenticarse en el sistema y pueda seguir con las acciones que desee. Al igual que en los casos anteriores, se le informará al usuario de que ha sobrepasado el tiempo máximo de estancia en la aplicación.

La sesión ha caducado. Vuelva a introducir usuario y contraseña

Usuario

Contraseña

Submit

Sesión caducada por tiempo

Por último, cabe recordar que la aplicación está validando constantemente que la información a la que un usuario intenta acceder es acorde al cliente que pertenece y a los permisos de usuario que tiene asignados dentro de nuestro sistema. En caso contrario, que un usuario intentará acceder a una parte de la aplicación que no tuviera permisos de acceso o que no correspondiera con los datos del cliente al que pertenece, sería redireccionando de forma inmediata al escritorio de la aplicación sin que se pudiera visualizar ninguna información ni datos.

6.1.1.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario introduce un usuario y contraseña correcto	El sistema permite acceso
El usuario introduce un usuario y contraseña incorrectos	El sistema deniega el acceso
El usuario introduce un usuario correcto y contraseña incorrecto	El sistema deniega el acceso
El usuario introduce 4 veces mal la contraseña con un usuario correcto	El sistema bloquea el acceso
El usuario solicita una nueva contraseña por correo	El sistema envía nueva contraseña si existe el correo introducido en el sistema
El usuario sobrepasa el tiempo máximo de estancia en la aplicación web	El sistema cierra la sesión y lo reenvía a la página de registro
El usuario solicita acceso a información inalcanzable por los permisos que tiene asignados	El sistema lo reenvía al escritorio de la aplicación
El usuario solicita acceso a información que no se corresponde su cliente	El sistema lo reenvía al escritorio de la aplicación

De esta manera se certifican que todas las pruebas realizadas son correctas, ya que la aplicación reacciona tal y como ha sido configurada.

6.1.2.- Error conexión a la Base de Datos

Cuando un usuario intenta acceder a uno de los servidores de Base de Datos, ya sea intraLEPTON o DATA_SLOT, es sistema comprobará, cada vez que se realice una conexión sobre ellos que están disponibles y que el usuario que intenta acceder a las mismas tiene los permisos suficientes para poder acceder. En caso contrario, la aplicación redireccionará al usuario a una página de información mostrando un mensaje mediante el cual se solicita el contacto con el administrador del sistema para que sea informado del error.



Ha sido imposible conectar con el servidor.
Vuelvalo a intentar o contacte con el administrador.

Error de conexión con servidor

Para la realización de esta prueba se decide cambiar el puerto sobre el que accederemos a la Base de Datos en los ficheros de nuestra aplicación, por lo que al intentar acceder a dicha Base de Datos a un puerto el cual no está preparado para recibir las consultas nos notifica que nos encontramos ante un problema de conexión con el servidor y se nos recomienda ponernos en contacto con el administrador de la aplicación.

6.1.2.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
Los servidores funcionan correctamente	El sistema muestra la información
Un servidor está caído	El sistema muestra un mensaje de error
Un servidor al que se intenta acceder nos deniega el acceso	El sistema muestra un mensaje de error

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.3.- Bloqueo de información crítica

Como se ha comentado en el punto de seguridad de la aplicación expuesto en el apartado de implementación, hay cierta información que debe ser tratada con más cuidado. Para ello se procedió a implementar un bloqueo para evitar que varios usuarios pudieran entrar a la edición de la misma información al mismo tiempo.

En el caso de que una información no se encuentre bloqueada por ningún usuario, se procederá a realizar un bloqueo sobre la misma impidiendo que ningún otro usuario pueda utilizarla hasta que hayan transcurridos 10 minutos de la asignación del recurso o hasta que el usuario no haya liberado la información.

Usted no puede editar en estos momentos los datos del usuario, intente más tarde

Aceptar

Información bloqueada por usuario

En caso contrario, que dicha información se encuentre bloqueada se nos mostrará un cuadro de diálogo informándonos de que actualmente no podemos acceder a su edición y que lo intentemos más tarde.

6.1.3.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario solicita editar un recurso libre	El sistema le permite editar el recurso
El usuario sobrepasa el tiempo máximo de mantener un recurso	El sistema si detecta que otro usuario ha solicitado el recurso se lo asignara al nuevo usuario
El usuario sobrepasa el tiempo máximo de mantener un recurso	El sistema si detecta que nadie ha solicitado la edición del mismo recurso permitirá al usuario editarlo
El usuario solicita la edición de un recurso ocupado	El sistema deniega editar el recurso

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.4.- Acceso a múltiples servidores

Una de las característica de nuestro sistema es que la información de los dispositivos nos la podemos encontrar en distintos servidores, sin que estos influya en el modo de visualización que tienen los usuarios. Esto quiere decir que en un mismo listado, hemos podido que tener que conectarnos a distintos servidores para recuperar la información.

LISTADO DE BEE TRACKERS

Número de Serie	Nombre del dispositivo	Tipo	Estado	Asentamiento	Localización	Cobertura	Gráficas
BT0000002	BT_TEST	Básico	⚠️ Alarma	Pruebas LEPTON	LAT. --- LONG. ---	Regular	
antesBT7	Pruebas Pastrana	Básico	💡 Apagado	Pruebas LEPTON	LAT. 39.4656 - LONG. -0.387092	Regular	
antesBT2	BT laboratorio	Básico	💡 Apagado	Pruebas LEPTON	LAT. 39.474 - LONG. -0.36312	Buena	
antesBT5	Pruebas Ricardo	🌟 Premium	💡 Apagado	Pruebas LEPTON	LAT. 39.4659 - LONG. -0.385772	Regular	

Listado de BeeTrackers alojados en distintos servidores

Un caso práctico que nos podríamos encontrar, es el listado de BeeTracker. En el listado de la imagen anterior nos encontramos con 4 dispositivos cuya información se encuentra almacenada en 2 servidores distintos.

Para el usuario, esa información es totalmente transparente pero para el sistema supone tener todo organizado y los sistemas siempre funcionando para que no se produzca ningún fallo. En caso de no poder conectarse con alguno de los servidores, se procedería a mostrar un mensaje de error como el del punto 6.1.2.

6.1.4.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario solicita información sobre un múltiples servidores operativos	El sistema muestra la información
El usuario solicita información alojada en otro servidor distinto del actual	El sistema muestra la información
El usuario solicita información sobre un múltiples servidores con problemas	El sistema informa de que está teniendo problemas

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.5.- Alarmas técnicas

El procedimiento de una alarma técnica conlleva que el BeeTracker generará una alarma y nuestro sistema debe de procesar la información para mostrar los datos al cliente. Como hemos dicho en apartados anteriores, en una alarma técnica recibimos información concreta de un parámetro en un momento dado.

Al usuario se le informa de que se ha producido dicha alarma, pero no puede ver de una forma gráfica toda la información completa, ya que no es hasta el día siguiente cuando recibe toda la información relativa a los parámetros capturas durante cierto día. Dicho de otra manera, únicamente obtendríamos un punto sin ninguna referencia de lo que ha ocurrido el resto del día. Pero esto se soluciona al recibir la información de los datos del día anterior, ya que se muestra una gráfica con la temperatura de 48 horas por lo que podemos comprobar la evolución que ha sufrido durante ese periodo de tiempo.

Una vez el usuario desee cerrar la incidencia, está pasara a un histórico donde podrá ser consulta cuantas veces lo desee el usuario. Puede ocurrir que debido al tipo de acceso que tenga el usuario no pueda cerrar la incidencia pero si visualizarla.

6.1.5.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario solicita información sobre alarma técnica	El sistema muestra la información
El usuario solicita información sobre alarma técnica en forma de gráfica	El sistema muestra la información con los datos históricos que tenga
El usuario cierra una alarma técnica	El sistema realiza la transición para ser alarma técnica histórica
El usuario cierra una alarma técnica que no tiene acceso o permiso	El sistema deniega la transición para ser alarma técnica histórica

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.6.- Alarmas de seguimiento

Al igual que ocurre con las alarmas técnicas para que se genere una alarma por seguimiento es necesario que el BeeTracker detecte mediante sus sensor un movimiento suficiente para ser activada. Una vez generada el sistema registra que se ha producido una alarma y la introduce en alarmas activas.

Como se ha comentado en los apartados anteriores la gran importancia de estos seguimientos es que te permite seguirlo en tiempo real. Por lo que una de las pruebas principales que se deben comprobar en el sistema es que el seguimiento vaya actualizando constantemente los datos sin que se produzca ningún error.

También en estos seguimientos es muy importante que si el sistema lleva un tiempo previamente establecido por el sistema sin recibir una posición GPS del dispositivo, es que facilite al usuario una posición aproximada en la que se encuentra el BeeTracker de forma automática a partir de los datos transmitidos por el dispositivos sobre la cobertura y posición de las antenas de telefonía móvil, mediante la posición AGPS.

Una vez el usuario desee cerrar la incidencia, está pasara a un histórico donde podrá ser consulta cuantas veces lo desee el usuario. Puede ocurrir que debido al tipo de acceso que tenga el usuario no pueda cerrar la incidencia pero si visualizarla. En este tipo de alarmas también se va a producir a generar un informe automático una vez el usuario ha realizado la transición. Esto se explicará en el punto 6.1.10 sobre procesos de tareas automáticas de generación de informes.

6.1.6.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario solicita información sobre alarma de seguimiento	El sistema muestra la información
El usuario solicita información sobre alarma de seguimiento en forma de gráfica	El sistema muestra la información con los datos recibidos hasta el momento de la petición
El usuario solicita visualizar el seguimiento completo de la alarma en tiempo real	El sistema muestra el seguimiento de la alarma actualizando la información cada segundo
El usuario solicita información sobre alarma de seguimiento que no tiene acceso o permiso	El sistema deniega la información
El usuario cierra una alarma de seguimiento	El sistema realiza la transición para ser alarma de seguimiento histórica
El usuario cierra una alarma de seguimiento que no tiene acceso o permiso	El sistema deniega la transición para ser alarma de seguimiento histórica
El usuario está visualizando un seguimiento sin nuevas posiciones GPS	El sistema muestra última posición del dispositivo mediante AGPS

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.7.- Logs servidores

Toda acción que se realiza en el sistema debe ser registrada para así poder gestionar y controlarla de forma más correcta y eficiente. De este modo, se procede a realizar el registro de logs por cada acceso que se intenta en el sistema, por cada vez que se accede al sistema y por cada vez que se abandona el mismo.

También se registra todas las acciones que se realizan sobre los dispositivos. De este modo se sabe que usuario ha apagado un BeeTracker, ha realizado cambio en su configuración, ha deshabilitado temporalmente el dispositivo o a cerrado una alarma.

Toda esta información se almacena, para que así posteriormente se puedan generar informes de registro de usuario y acciones realizadas por cada usuario.

6.1.7.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario se registra correctamente	El sistema registra que el usuario ha accedido a la aplicación
El usuario se registra con un usuario y contraseña incorrectas	El sistema registra la información incorrecta de acceso
El usuario se registra con un usuario correcto y una contraseña incorrecta	El sistema registra la información incorrecta de acceso
El usuario intenta acceder con un usuario bloqueado	El sistema registra la información del usuario bloqueado
El usuario cierra sesión en la aplicación	El sistema registra que el usuario ha cerrado sesión en la aplicación
El usuario apaga un BeeTracker	El sistema registra que usuario ha apagado el BeeTracker
El usuario deshabilita temporalmente un BeeTracker	El sistema registra que usuario ha deshabilitado temporalmente el BeeTracker
El usuario cambia los parámetros de configuración de alarmas de un BeeTracker	El sistema registra que usuario ha cambiado los parámetros de configuración de alarmas del BeeTracker y cual es su nueva configuración
El usuario cierra una alarma	El sistema registra que usuario ha cerrada una alarma del BeeTracker

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.8.- Ayuda en línea

En la ayuda en línea es el apartado donde el usuario podrá exponer todas sus dudas y opiniones sobre la aplicación de una forma sencilla a los administradores de la aplicación. Dentro de este apartado al usuario se les debe notificar la existencia de nuevas respuestas por parte de los administradores, crear incidencias de una forma muy sencilla y poder visualizar la información de las incidencias previamente existentes.

Al crear una incidencia el usuario debe poder visualizar que su estado es el de pendiente de contestación por parte del administrador y una vez ha sido contestada por ellos pasará a estado contestada. Una de las restricciones de esta parte es que únicamente los administradores pueden considerar que una incidencia ha sido cerrada.

6.1.8.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
El usuario crea una incidencia	El sistema registra la información correctamente y la marca como pendiente
El usuario crea una incidencia de forma incorrecta	El sistema muestra un mensaje advirtiendo del error que se ha producido
El usuario accede a la aplicación y tiene mensajes contestados	El sistema informa de que hay nuevos mensajes por leer
El usuario solicita visualizar información de una incidencia	El sistema muestra la información

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.9.- Tareas automáticas de generación de informes

Para concluir el apartado de pruebas, hablaremos de los procesos de generación de informes automáticos que como ya hablamos en el apartado anterior, no son parte directa de la aplicación web pero si está muy ligada a ella.

6.1.9.1.- Informes de alarma de seguimiento

La aplicación de informes de alarmas de seguimiento se ejecutan todos los días a las 00.00, creando un informe de todas aquellas incidencias de seguimiento que hayan sido cerradas el día posterior a la ejecución del mismo.

En caso de producirse un fallo durante el mismo se procederá a dar un aviso a los administradores del sistema que ha habido un fallo en la creación de informes.

Los administradores tiene las herramientas suficientes para volver a lanzar la aplicación o de lanzar únicamente la generación de informes de una incidencia en concreto.

Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
Generación automática de informes	El sistema genera y almacena el informe de forma correcta
Generación automática de informes con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe
Generación de un informe en concreto de forma manual	El sistema genera y almacena el informe de forma correcta

Generación de un informe en concreto de forma manual con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe
--	---

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.9.2.- Informes de registro

La aplicación de informes de registro de acceso se ejecutan el día 2 de cada mes, creando un informe de todos aquellos accesos a la aplicación por usuario y todas las acciones realizadas por los mismos durante al mes anterior al de ejecución del programa.

En caso de producirse un fallo durante el mismo se procederá a dar un aviso a los administradores del sistema que ha habido un fallo en la creación de informes.

Los administradores tiene las herramientas suficientes para volver a lanzar la aplicación o de lanzar únicamente la generación de informes de un cliente en concreto.

Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
Generación automática de informes	El sistema genera y almacena el informe de forma correcta
Generación automática de informes con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe
Generación de un informe en concreto de forma manual	El sistema genera y almacena el informe de forma correcta
Generación de un informe en concreto de forma manual con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.9.3.- Informes de telemidas

La aplicación de informes de registro de acceso se ejecutan el día 3 de cada mes, creando un informe de todos aquellos dispositivos activos en el sistema, mostrando la información de todos los datos de temperatura, humedad, sonido y batería recogidos por los BeeTracker en el mes anterior a la ejecución del programa.

En caso de producirse un fallo durante el mismo se procederá a dar un aviso a los administradores del sistema que ha habido un fallo en la creación de informes.

Los administradores tiene las herramientas suficientes para volver a lanzar la aplicación o de lanzar únicamente la generación de informes de un dispositivo en concreto.

Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
Generación automática de informes	El sistema genera y almacena el informe de forma correcta
Generación automática de informes con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe
Generación de un informe en concreto de forma manual	El sistema genera y almacena el informe de forma correcta
Generación de un informe en concreto de forma manual con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.9.4.- Facturas

La aplicación de informes de registro de acceso se ejecutan el día 4 de cada mes, creando una factura por cada uno de los clientes activos en el sistema en la que se incluyen todos los cobros que se le hayan realizado en el mes anterior a la ejecución de la aplicación.

En caso de producirse un fallo durante el mismo se procederá a dar un aviso a los administradores del sistema que ha habido un fallo en la creación de informes.

Los administradores tiene las herramientas suficientes para volver a lanzar la aplicación o de lanzar únicamente la generación de facturas de un cliente en concreto.

Pruebas realizadas

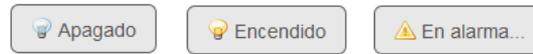
PRUEBA	RESPUESTA DEL SISTEMA
Generación automática de informes	El sistema genera y almacena el informe de forma correcta
Generación automática de informes con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe
Generación de un informe en concreto de forma manual	El sistema genera y almacena el informe de forma correcta
Generación de un informe en concreto de forma manual con error en el proceso	El sistema avisa al administrador de que se ha producido un error al generar el informe

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.1.10.- Proceso completo

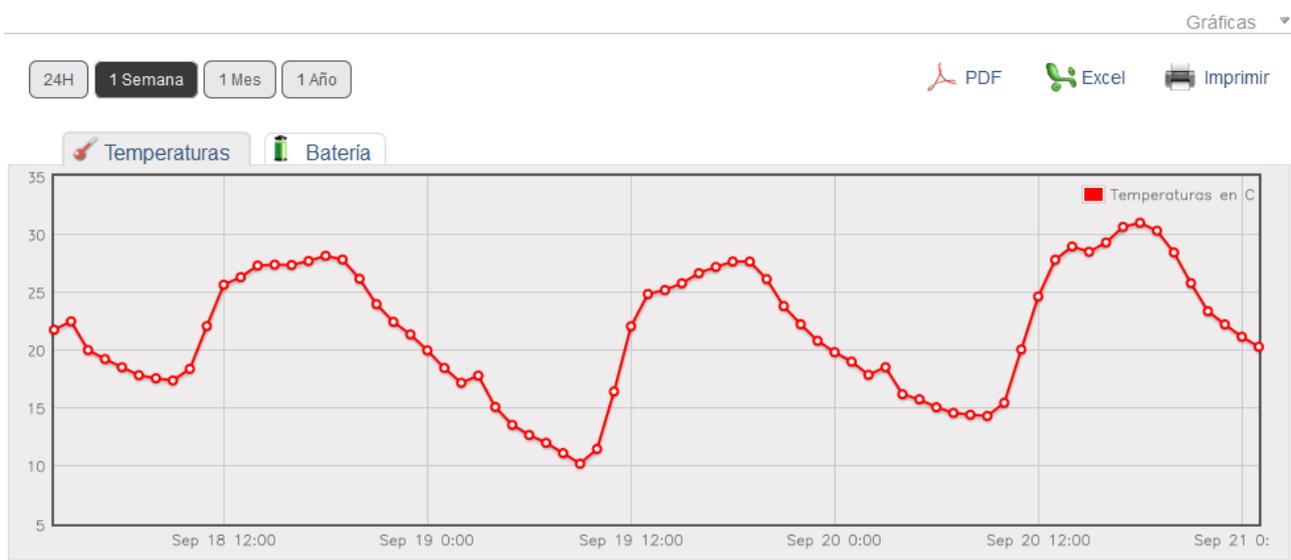
Para concluir el apartado de prueba funcionales, nos encontramos ante la prueba completa del sistema desde que un usuario conecta por primera vez el dispositivo BeeTracker hasta la generación de los primeras alarmas técnicas y de seguimiento.

Cuando un usuario, conecta por primera vez el dispositivo este debe de realizar un cambio de estado dentro de nuestro sistema dando constancia al usuario de que dicho dispositivo se encuentra conectado y trabajando en condiciones.



Botones de estado

El usuario también podrá consultar todos los datos de los diferentes parámetros recogidos por los dispositivos mediante una simple gráfica que recoge los datos de las últimas 24 horas, la última semana, el último mes o el último año, así como permitiendo imprimir o exportarlos a ficheros en formato pdf o csv.

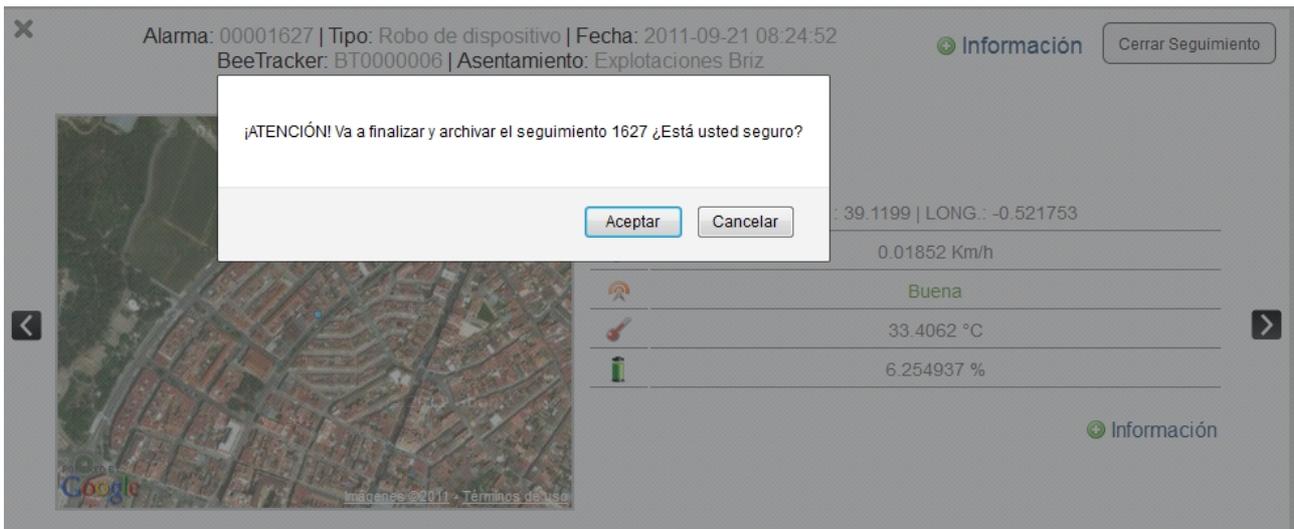


Una vez esté el dispositivo conectado, se nos permitirá siempre y cuando tengamos los permisos necesarios para poder hacerlo, se nos permitirá desconectarlo, deshabilitarlo o cambiarle la configuración de los parámetros de las alarmas.

Cuando un dispositivo genera una alarma de técnica el usuario podrá consultar la información y en cualquier momento podrá proceder a realizar el cierre de la misma siempre que tenga permisos suficientes para poder realizar esta tarea sobre el dispositivo. La información mostrada en este tipo de alarmas debe ser lo más concreta posible y fácil de procesar para el usuario.

Si el dispositivo en cambio ha generado una alarma de seguimiento, el sistema nos debe notificar en cualquier ventana del dispositivo que nos encontremos que se encuentra en modo seguimiento debido a la detección de movimiento.

Al igual que en el caso anterior de las alarmas técnicas, aquí el usuario con los suficientes privilegios sobre el dispositivo podrá proceder al cierre de la incidencia.



Cierre seguimiento

Tanto los usuarios que tiene permisos de lectura sobre el dispositivo como los que tiene poder de escritura pueden entrar a consultar el seguimiento en tiempo real para poder ver cuales han sido todos las posiciones obtenidas por el dispositivo desde que entro en seguimiento hasta el momento actual y continuando con el seguimiento desde ese mismo momento.



Seguimiento en tiempo real

Una vez el usuario ha realizado la transición de los datos al histórico mediante el cierre de las alarmas, estas podrán ser consultadas en cualquier momento por los usuarios y recuperadas en diferentes formatos como csv, pdf, gpx o imprimiendo directamente desde la aplicación web.

6.1.10.1.- Pruebas realizadas

PRUEBA	RESPUESTA DEL SISTEMA
Cambio de estado de dispositivo	El dispositivo cambiaba de estado según se encontrara apagado, encendido sin alarma de seguimiento o en estado de alarma de seguimiento.
Configurar dispositivo, el usuario actualiza los datos de configuración de los parámetros de alarma por usuario con permisos.	El sistema actualiza los cambios en la Base de Datos, y el dispositivo en la próxima conexión que realiza recoge los nuevos parámetros.
Configurar dispositivo, el usuario actualiza los datos de configuración de los parámetros de alarma por usuario con permisos.	Cancela la operación y es devuelto al escritorio de la aplicación web.
Deshabilitar dispositivo, el usuario decide deshabilitar el dispositivo durante un periodo de 24 horas por usuario sin permisos.	El sistema actualiza los cambios en la Base de Datos, y el dispositivo en la próxima conexión que realiza recoge los nuevos datos de deshabilitación permitiendo así durante ese periodo que el dispositivo no notifique alarmas.
Deshabilitar dispositivo, el usuario decide deshabilitar el dispositivo durante un periodo de 24 horas por usuario sin permisos.	Cancela la operación y es devuelto al escritorio de la aplicación web.
Exportar gráficas a CSV	El sistema devuelve un fichero en formato csv con todos los datos de la gráfica que el usuario tiene actualmente en pantalla.
Exportar gráficas a PDF	El sistema devuelve un fichero en formato pdf con todos los datos de la gráfica que el usuario tiene actualmente en pantalla.
Imprimir gráficas mediante aplicación web	El sistema imprime la gráfica que el usuario tiene actualmente en pantalla.
Cierre de alarma técnica por usuario con permisos	Solicita confirmación de la operación y si acepta realiza la transición sobre la alarma.
Cierre de alarma técnica por usuario sin permisos	Cancela la operación y es devuelto al escritorio de la aplicación web.
Cierre de alarma de seguimiento por usuario con permisos	Solicita confirmación de la operación y si acepta realiza la transición sobre la alarma.
Cierre de alarma de seguimiento por usuario sin permisos	Cancela la operación y es devuelto al escritorio de la aplicación web.
Seguimiento en tiempo real	El sistema actualiza correctamente y automáticamente los nuevos datos que han sido recibidos desde el momento posterior al acceso a la aplicación.

Exportar seguimiento a CSV	El sistema devuelve un fichero en formato csv con todos los datos de la alarma de seguimiento que está visionando actualmente el usuario.
Exportar seguimiento a GPX	El sistema devuelve un fichero en formato gpx con todos los datos de la alarma de seguimiento que está visionando actualmente el usuario.
Exportar seguimiento a PDF	El sistema devuelve el informe que se ha generado con la alarma de seguimiento que está actualmente visionando el usuario (informe generado automáticamente)
Imprimir seguimiento desde la aplicación web	El sistema imprime la alarma de seguimiento que está visionando actualmente el usuario.

Con estas pruebas realizadas certificamos que el funcionamiento de la aplicación es el deseado.

6.2.- Pruebas de validación

6.2.1.- Validación W3C

Para poder validar los elementos de la web hemos usado el validador de la W3C.

En las Hojas de Estilo en Cascada se ha usado el perfil CCS3 en cada una de las plantillas utilizadas en nuestra aplicación. Todas las pruebas realizadas han superado la validación correctamente.

Una vez finalizadas las pruebas se ha decidido implementar funciones fuera de los estándares para que el visionado de las páginas más estético.

Para validar el código HTML, al tratarse de una aplicación con registro de usuario, la aplicación de validación no puede acceder a la información, por lo que hemos tenido que validar el código HTML generado por cada una de nuestras páginas en forma de fichero gracias a la extensión del navegador Firefox denominada Web Developer, superando todas las páginas la validación del estándar XHTML 1.0 Transitional.

7.- CONCLUSIÓN

Durante el transcurso del proyecto he ido sacando conclusiones tanto a nivel personal como a nivel técnico que me han ido permitiendo adaptarme a todas las necesidades y dificultades que han ido surgiendo a lo largo de su desarrollo.

A nivel técnico, aunque ya tenía conocimientos sobre los lenguajes de programación usados en el desarrollo de mi aplicación web, nunca me había visto en la necesidad de tener que investigar y enfrentarme con todas las posibilidades de funcionalidad que llegan a darnos cada uno de ellos. Me ha servido para descubrir la librería XAJAX, que ha sido una herramienta bastante útil para mi aplicación. He aprendido a utilizar el JavaScript y los CSS (páginas de estilo en cascada), anteriormente conocía su existencia pero nunca las había usado. He mejorado los conocimientos que tenía con el SGBD MySQL y uno de los aspectos destacables en este punto ha sido evitar el uso de subconsultas para mejorar el rendimiento global. Limitar la maquetación a XHTML 1.0 ha supuesto tener mucha cura en el código para poder pasar los estándares, así como en la maquetación de las páginas de estilos en cascada (CSS) también se tuvo cura para pasar los estándares. También he aprendido a usar y el servidor web Apache así como a usar la compresión en las páginas generadas para optimizar los recursos.

Enfrentarme a crear una aplicación dentro de una empresa me ha llevado a aprender que la labor de desarrollo, aunque yo fuera el único desarrollador de la misma, no es una tarea unilateral sino que ha dependido de los cuatro compañeros que he tenido en la empresa, los 3 miembros de LEPTON y mi compañero que se encargó de la realización de la Intranet.

Cuando me plantearon la realización del proyecto, no me imagine los problemas que podrían llegar a surgir a lo largo de su desarrollo ni como mediante la perseverancia y las ganas de sacarlo todo adelante se puede llegar a cumplir todos los objetivos fijados. De ver como posibles soluciones planteadas al principio se convertían en problemas durante la implementación de cualquier otra parte de la aplicación.

Personalmente pienso, que día a día he ido superando todos los obstáculos que se me iban cruzando en el camino, y aunque alguno de ellos conllevará más dificultad he sabido aprender que en la programación no existe lo imposible y siempre hay una solución a cualquier problema y muchas veces depende de la inspiración y que enfrascarse en un problema sólo conlleva más problemas.

Aunque el ciclo de desarrollo ha sido bastante largo, me ha servido para ver que con planificación y constancia cualquier proyecto por muy complejo que parezca se puede sacar a adelante.

Como conclusión final, quisiera remarcar el hecho de que el desarrollo de la aplicación web me ha servido para interactuar con todo el grupo de desarrollo del sistema BeeTracker y ver que estando dentro de él puedes aportar ideas que en un principio no estaban contempladas pero que pueden llegar a ser de gran utilidad.

La experiencia que me llevo del desarrollo de esta aplicación es la mejor de todas ya que me ha enseñado que puedo ser capaz de enfrentarme a retos que nunca me hubiera planteado.

8.- BIBLIOGRAFÍA

8.1.- Páginas Web

- [1] <http://es.wikipedia.org/wiki/Wikipedia:Portada>
- [2] <http://www.programacion.com/>
- [3] <http://www.desarrolloweb.com/>
- [4] <http://dev.mysql.com/doc/refman/5.0/en/>
- [5] <http://www.webestilo.com/>
- [6] <http://php.net/manual/es/index.php>
- [7] <http://code.google.com/p/flot/>
- [8] <http://www.xajax-project.org/>
- [9] <http://www.tcpdf.org/>
- [10] <http://jquery.com/>
- [11] <http://validator.w3.org/>
- [12] <http://www.w3schools.com/php>
- [13] <http://stackoverflow.com/>

8.2.- Libros

- [1] *Desarrollo web con PHP, Apache y MySQL*
- [2] *LAMP. : desarrollo web, con Linux, Apache, MySQL y PHP 5*
- [3] *Ajax. : web 2.0 para profesionales*
- [4] *Domine JavaScript*