

Anexo A. Manual de usuario de la agenda de vialidad (versión móvil)

En este primer anexo se presenta el manual de usuario elaborado con el fin de detallar el funcionamiento y las posibilidades de la aplicación pensando en el usuario final.

A.1. Introducción

La aplicación "Agenda de vialidad" ha sido desarrollada para permitir su correcta visualización en cualquier tipo de dispositivo (ordenador, tablet o móvil), ya que cuenta con una versión móvil con funcionalidades específicas para permitir la introducción de las actividades realizadas por los operarios en el mismo lugar en el que realizan una acción. Por ejemplo, pueden dar de alta nuevas tareas en el sistema, incorporando automáticamente toda la información asociada como fotografías, localización u otros datos.

La versión optimizada para dispositivos móviles (tablets y smartphones) será el objeto de este manual de usuario, y cuenta con las siguientes características:

- Compatible para diferentes modelos de dispositivo.
- Optimizada para minimizar los tiempos de carga, y agilizar el el uso de la aplicación por parte del usuario en entornos remotos.
- Se tienen en cuenta los problemas de límites de ancho de banda y cobertura de las redes móviles, disponiendo de los mecanismos oportunos para agilizar la velocidad de la aplicación y evitar la pérdida de información por falta de cobertura.

Respecto a la cobertura, la aplicación dispone de todos los elementos necesarios para permitir trabajar con ella aún cuando ésta no exista (de forma autónoma), volviendo a sincronizar la información con los servidores cuando la cobertura vuelva a estar disponible, de forma que se garantice la continuidad del funcionamiento sin perjuicio alguno para el usuario.

La aplicación permite a los operarios la actualización instantánea de las tareas en el mismo momento en que proceden a su actuación, al mismo tiempo que se controla la posición exacta en la que se encuentra, posibilitando el seguimiento de la resolución de las tareas por parte de la empresa para mantener el control sobre las actuaciones realizadas.

El módulo de agenda móvil es similar al que se puede observar en su versión extendida (para plataforma web), aunque está orientado a la introducción de información (y no a la consulta de informes).

A.2. Opciones del usuario

A.2.1. Acceso a la aplicación (login)

Al ejecutar la aplicación en el dispositivo móvil se muestra la pantalla de entrada en la que se deben ingresar los datos del usuario registrado (login/paswword).



Figura 1. Pantalla inicial



Figura 2. Formulario de registro de usuario

Si los datos son correctos, desde ese momento el usuario (operario) podrá acceder a las opciones que, según los permisos asignados a ese usuario, tenga disponibles:

- Ver los registros de tareas asignadas
- Registrar una actuación
- Iniciar / finalizar una actuación
- Validar tareas

Además, se guardan los datos del usuario en la base de datos del móvil, de manera que cuando cierre o minimice la aplicación no se le vuelva a solicitar los datos de usuario (sólo se le vuelve a solicitar cuando el usuario cierra la sesión desde el menú de configuración).

A.2.2. Inicialización y configuración de la aplicación

La primera vez que se ejecuta la aplicación el usuario debe hacerlo asegurándose de que dispone de conexión a Internet. Al registrarnos por primera vez aparecen los siguientes mensajes:

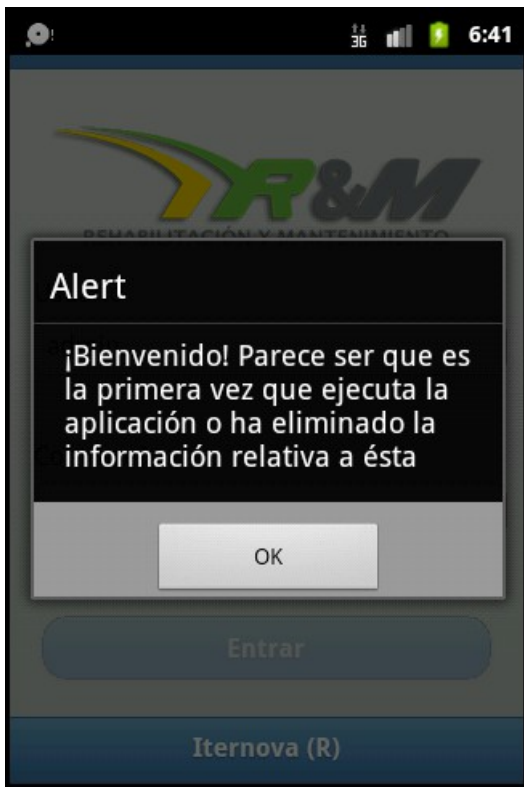


Figura 3. Configuración automática inicial (1)

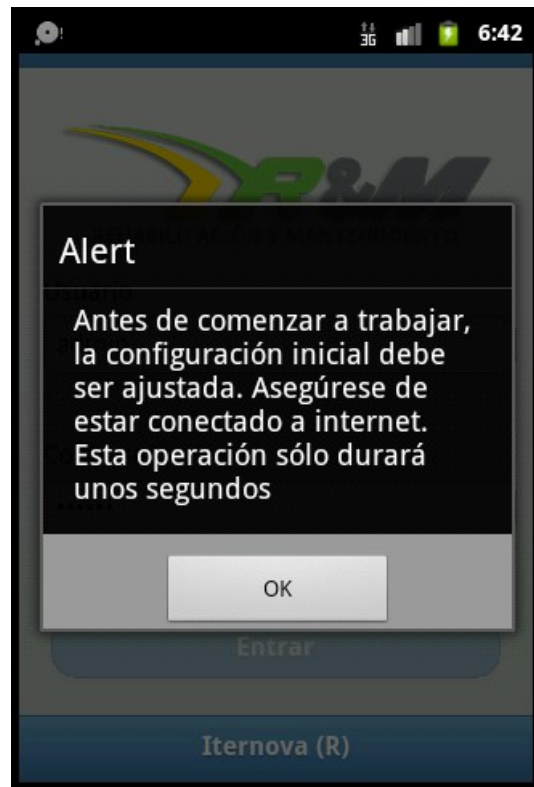


Figura 4. Configuración automática inicial (2)

Simplemente dejamos que la aplicación haga su trabajo, y al finalizar con éxito observaremos el mensaje:



Figura 5. Configuración automática inicial (3)

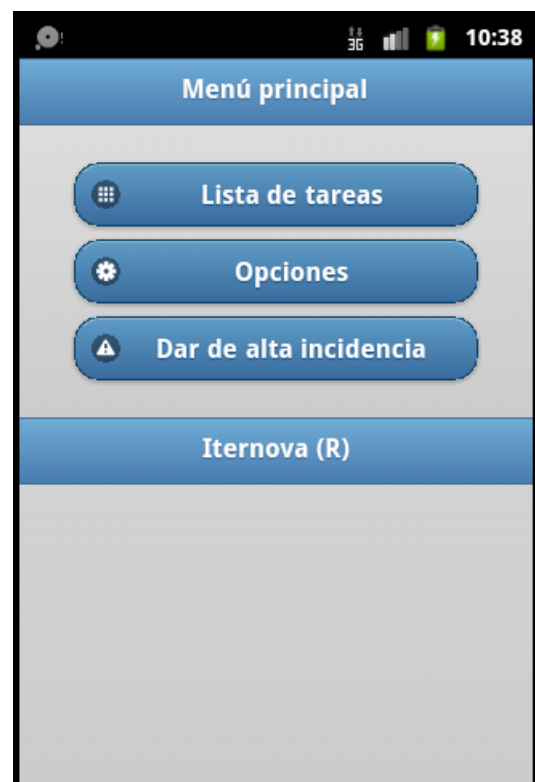


Figura 6. Menú principal

Para poder comenzar a trabajar, lo primero que debemos hacer es establecer un sector y tramo. Para ello deberá acceder al apartado de "Opciones" que se muestra en el menú principal de la aplicación y seleccionar un sector del desplegable correspondiente, ya que solo recibirá tareas del servidor del sector que le corresponda. También puede seleccionar un tramo/segmento de carretera, para recibir incidencias sólo de este tramo.

Si por alguna razón la configuración automática inicial fallara, deberemos hacer click sobre "Opciones" del menú principal y seguir los siguientes pasos por orden:

Desde esta pantalla deberá llevar a cabo las siguientes acciones por orden:

1. Borrado maestro: necesario para realizar la primera instalación.
2. Cargar base de datos - alta: en este momento, la aplicación se conecta con el servidor y se descargará todas las configuraciones "dinámicas" desde el servidor, por ejemplo: clases, tipos, subtipos de tareas de la agenda de vialidad, las carreteras configuradas en el sistema, los sectores, etcétera...
3. Seleccionar sector: el usuario deberá seleccionar un sector del desplegable correspondiente, ya que solo recibirá tareas del servidor del sector que le corresponda. También puede seleccionar un tramo / segmento de carretera, para recibir incidencias solo de este tramo.

Con estos pasos ya está preparada la aplicación para ser utilizada. Sin embargo hay otras opciones de configuración que explicaremos en el siguiente apartado. Las siguientes imágenes corresponden al submenú "Opciones":



Figura 8. Opciones de configuración (1)



Figura 7. Opciones de configuración (2)

Otras opciones de configuración

Desde la pantalla de "Opciones" vamos a poder configurar algunas características de funcionamiento de la aplicación (que podremos variar siempre que deseemos).

Modo de transmisión de imágenes y tamaño

Podemos elegir cómo se va a comportar la aplicación en caso de que tenga que enviar imágenes al servidor:

- A) Usar comunicación 3G para el envío de imágenes: cuando una imagen ha de ser enviada, basta con que exista conexión 3G para ser enviada en ese momento. No es muy recomendable ya que las conexiones 3G suelen tener restricciones de ancho de banda y datos totales enviados.
- B) No usar comunicación 3G: la transmisión de imágenes entre el dispositivo y el servidor sólo se realizará cuando exista conexión Wi-Fi, (opción por defecto).

También podemos configurar la resolución con la que deseamos enviar las imágenes al servidor. Con la ayuda del desplegable seleccionamos la resolución.

Actualización de la lista de tareas

Se puede configurar si la lista de tareas se actualiza automáticamente o no y, en caso de que lo haga, la frecuencia con la que lo hará.

Borrar reportes y tareas

En la base de datos del dispositivo móvil se queda almacenada la información que por falta de conexión con el servidor no ha podido ser todavía enviada, pero también aquella información que ya ha sido enviada y confirmada.

Como esta información ya enviada correctamente ocupa espacio en la base de datos del dispositivo móvil, tenemos la opción de borrarla de la base de datos del dispositivo.

Para ello, en la sección de "Opciones" deberemos pulsar el botón "Borrar reportes y tareas". Debe quedar claro que esta operación sólo afecta a la información que ha llegado con éxito al servidor, por lo que no hay peligro de perder información.

Borrado maestro

Esta opción provocará una restauración completa de la configuración de la aplicación, volviendo a los valores de fábrica. No debe utilizarse a no ser que la aplicación muestre comportamientos extraños que no puedan solucionarse de otro modo.

Carga de base de datos – alta

Si en el sistema se introducen cambios en las bases de datos que describen las tareas (inclusión de nuevos tipo de incidencia, nuevos sectores o cambios similares), será necesario hacer click en este botón para poder dar de alta incidencias que contengan los nuevos cambios.

Sector y tramo

Éstos son dos menús desplegables que permiten seleccionar el sector y el/los tramo(s) en los que va a trabajar el usuario. Estos dos parámetros condicionan la lista de tareas que recibirá desde el servidor, que es específica para cada sector y tramo(s). Es obligatorio seleccionar un sector para poder recibir la lista de tareas. Sin embargo, si no se selecciona ningún tramo se recibirá la lista de tareas asociada a todos los tramos de dicho sector.

Envío automático de información pendiente

De la misma manera que podemos configurar la actualización automática de la lista de tareas, podemos configurar el envío automático de la información pendiente (aquella que permanece almacenada sin que anteriormente haya podido ser transferida con éxito). Tenemos una opción para activar el envío automático y otra para configurar la frecuencia de envío.

Envío de formularios pendientes

Además del envío automático que podemos configurar según el punto anterior, existe la posibilidad de enviar toda la información pendiente al mismo tiempo de forma manual. Para ello, solamente tendremos que hacer click en el botón "Enviar formularios pendientes".

Si hay conexión a Internet en ese instante, toda la información pendiente será transferida al servidor. De lo contrario, seguirá almacenada esperando el momento oportuno.

A.2.3. Acciones con las tareas

Desde el menú inicial, accedemos al botón "Lista de tareas".

Al pulsar sobre el botón, se muestra la pantalla en la que se pueden observar, en caso de existir, las tareas asignadas al operario que todavía no han sido validadas:

Ver detalle de tarea

En el listado de la pantalla anterior, podemos pulsar sobre la tarea deseada y accedemos a la pantalla con los detalles de la tarea a realizar y desde la cual podremos llevar a cabo otras acciones.



Figura 9. Lista de tareas

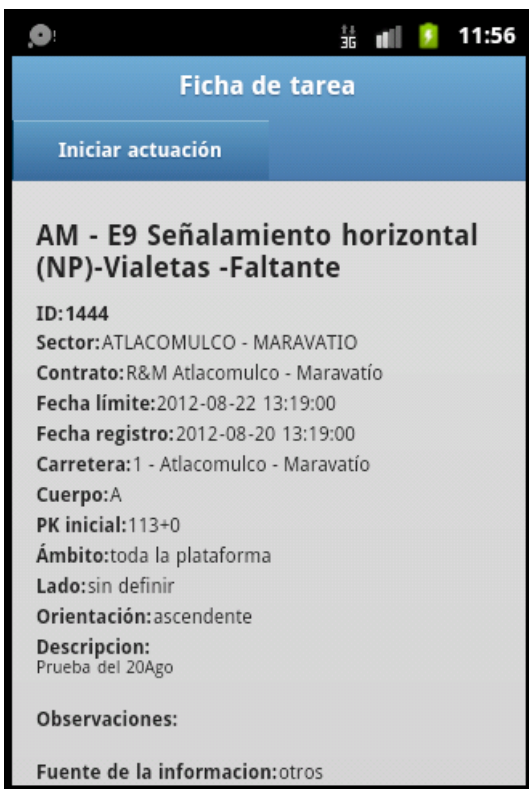


Figura 10. Ficha de tarea (1)



Figura 11. Ficha de tarea (2)

Iniciar actuación

Pulsando en el botón "Iniciar actuación" de la pantalla anterior, pasamos a la pantalla en la que el operario va a rellenar la información de la tarea (redactar el informe). Es importante recordar que tanto para iniciar una actuación como para finalizarla (proceso descrito a continuación) es necesario que el usuario disponga de los permisos correspondientes.



Figura 12. Informe de actuación (1)

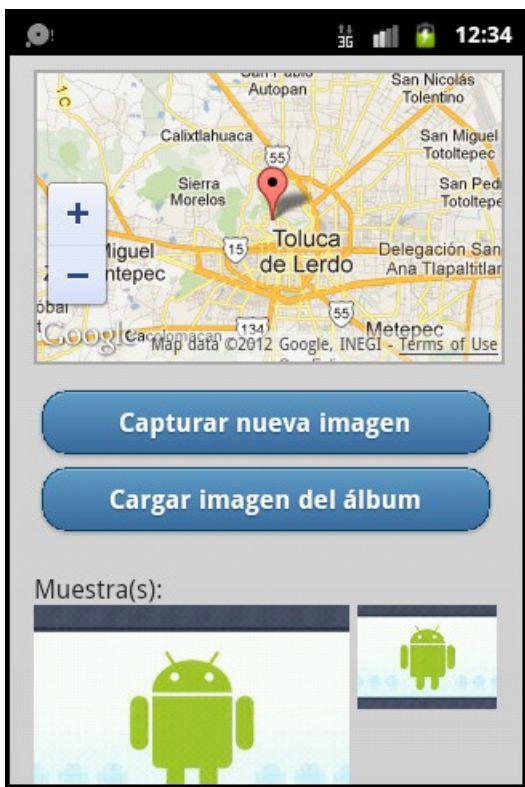


Figura 13. Informe de actuación (2)



Figura 14. Informe de actuación (3)

El operario podrá capturar una imagen en ese momento de la tarea que está realizando o añadir una imagen tomada previamente. A cada tarea se le pueden asociar hasta un máximo de 3 imágenes.

Los datos de la fecha, hora y geolocalización serán tomados automáticamente por el dispositivo. Pulsando el botón "Enviar", todos los informes de actuación (iniciación y finalización) pendientes, incluyendo el que acabamos de redactar, serán enviados al servidor si se dispone de conexión. Si no, el que acabamos de rellenar quedará almacenado a la espera.

Una vez realizada esta acción, haya o no conexión con el servidor, si volvemos al detalle de la tarea vemos que el marcador de reporte inicial aparece como completado.

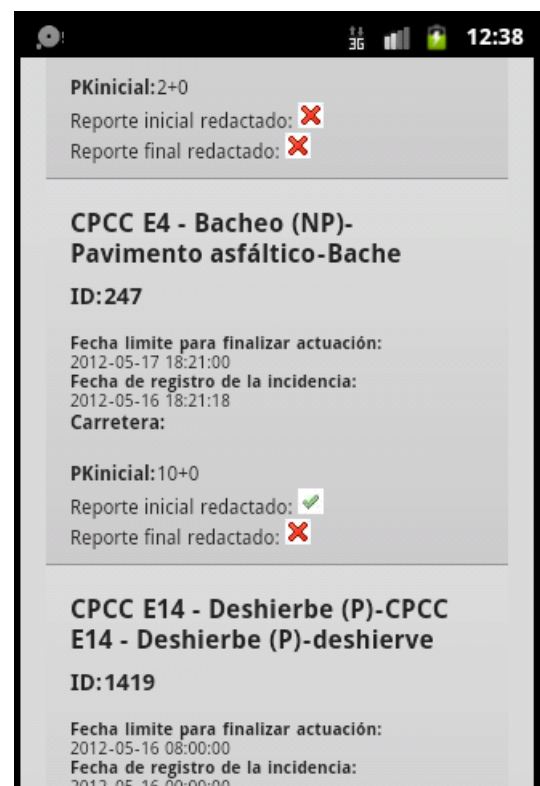


Figura 15. Lista de tareas. Informe inicial redactado

Finalizar actuación

Al igual que para iniciar la actuación, existe otro botón "Finalizar actuación" para realizar la tarea correspondiente. Este botón sólo se mostrará si el reporte inicial ya ha sido redactado.

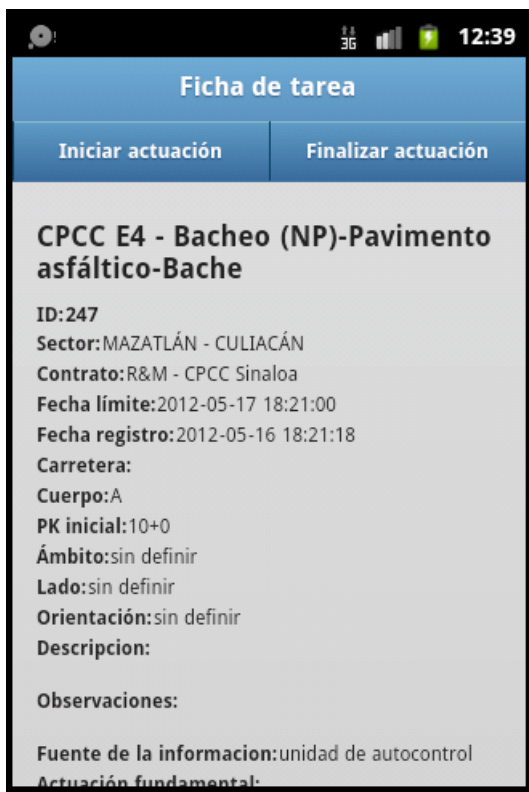


Figura 16. Ficha de tarea. Aparece la opción de finalizar



Figura 17. Informe de actuación final

Pulsándolo se muestran las mismas pantallas que se muestran en "Iniciar la actuación" para redactar (imágenes incluidas) el reporte final, aunque en este caso tendremos un campo "observaciones" para describir el trabajo realizado.

Una vez redactado y pulsado el botón "Enviar", se sucederá el mismo procedimiento que al iniciar la tarea. Volviendo al detalle de la tarea, se mostrará que los reportes de inicio y fin están redactados.

Validar una tarea

Si el usuario tiene permisos para ello, también podrá realizar la validación de una tarea. Esto supone dar el visto bueno a los trabajos realizados y hacerlo así constar en el sistema.

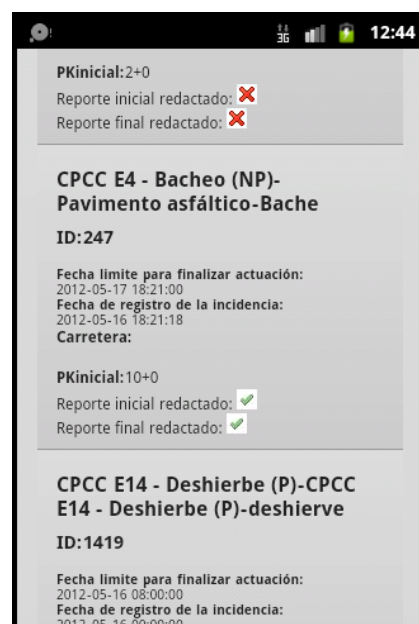


Figura 18. Lista de tareas. Ambos informes redactados

Para validar una tarea, deberemos acceder a la pantalla de "Ver detalle de tarea". Bajo la información de la tarea se muestra, en caso de haber rellenado los informes de iniciación y finalización, el botón "Validar". Pulsamos el botón y si disponemos de conexión, el servidor registrará esta validación. Si no, la aplicación tratará de reenviar la petición en otro momento como en el caso de los otros informes.

A.2.4. Dar de alta incidencias

Los operarios que dispongan del permiso necesario pueden dar de alta nuevas incidencias en el sistema. Para ello, desde el menú de la pantalla inicial se deberá pulsar el botón "Dar de alta incidencia". Al pulsarlo se accede al formulario con el que ingresar los datos de la incidencia (todas las imágenes que se muestran a continuación pertenecen a una misma pantalla con el formulario de ingreso de datos de la incidencia):

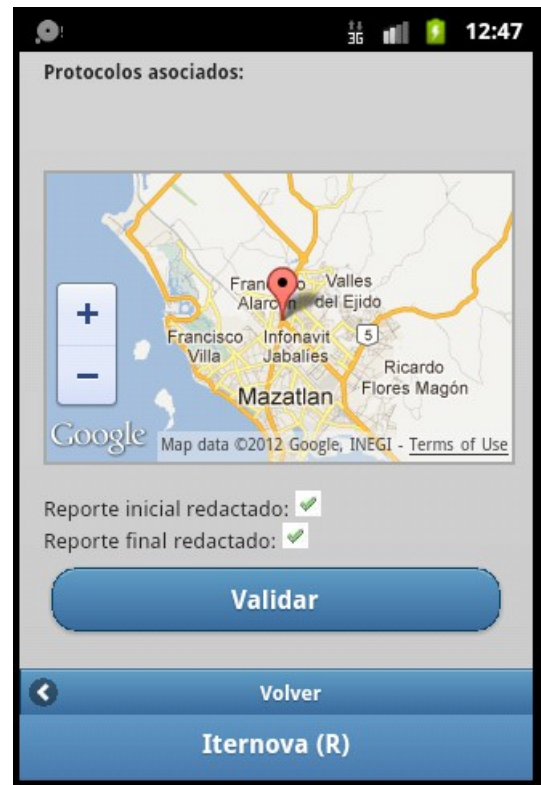


Figura 19. Ficha de tarea. Botón de validar



Figura 20. Alta de nueva tarea (1)

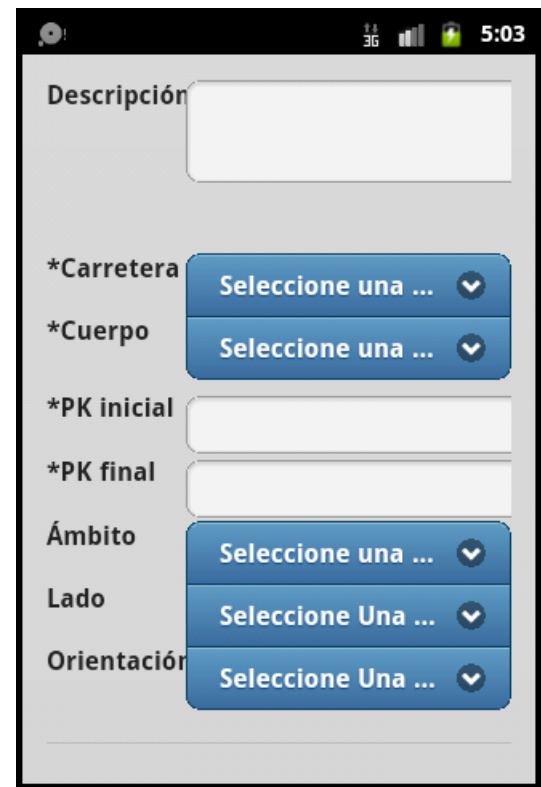


Figura 21. Alta de nueva tarea (2)

Con la ayuda del formulario irá ingresando la información:

- Primero deberá ingresar los datos completos del segmento y carretera de la que se trata con la ayuda de los desplegados y el resto de campos destinados a tal efecto.
- Podrá añadir imágenes, bien tomadas en ese momento o seleccionándolas de un repositorio (máximo 3 en cada formulario).
- Deberá seleccionar el tipo de agente que registra la incidencia
- Dependiendo del tipo de incidencia se muestra la actuación a llevar a cabo (acción fundamental y protocolo específico).
- Podrá añadir las observaciones que considere oportunas.
- Finalmente, dependiendo del tipo de incidencia y de la carta de servicios se especificarán las fechas en las que se debe finalizar la actuación.

Hay que remarcar que los campos marcados con el símbolo '*' deben ser rellenados obligatoriamente (de otro modo no podremos enviar la información).

Al pulsar el botón "Enviar" se tratará de enviar la información al servidor siguiendo el mismo procedimiento visto en los anteriores casos, es decir, gestionando adecuadamente los casos de ausencia de conexión.

Anexo B. Elementos clave del código de la aplicación

En este anexo comentaremos los puntos de la aplicación fundamentales para comprender su funcionamiento sin entrar en detalles de desarrollo demasiado técnicos. Estos elementos clave serán las bases de datos, la caché local del navegador, y algunas funciones empaquetadas para la aplicación y ampliamente utilizadas.

B.1. Base de datos

SQLite es el motor de bases de datos elegido por los desarrolladores de Android ya que ofrece características tan interesantes como su pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y por supuesto ser de código libre.

Android incorpora de serie todas las herramientas necesarias para la creación y gestión de bases de datos SQLite, y entre ellas una completa API para llevar a cabo de manera sencilla todas las tareas necesarias. Sin embargo no entraremos en los detalles sino que en la presente sección se expondrá la estructura de la base de datos que se ha implementado para el desarrollo del sistema.

B.1.1. Tabla de usuarios

Primeramente se presenta la más sencilla de las tablas, aquella que recoge la información del usuario registrado. Hemos de recordar (si se necesita más información, consultar el apartado [3.2.1. Registro y salida en el sistema](#)) que esta tabla nunca almacenará la información relativa a más de un usuario por motivos de seguridad. Por tanto, en esta tabla siempre habrá registrado uno o ningún usuario.

USUARIOS		
login	password	remote

Tabla 1. Usuarios

Como podemos observar, la información almacenada consta de tres campos:

- login: es el nombre de usuario.
- password: la contraseña del mismo.
- remote: éste es un parámetro que en el caso de la aplicación móvil siempre vale el string "tareas".

La información de esta tabla es consultada cada vez que se inicia la aplicación. Si existe algún usuario guardado, se envía al servidor su información automáticamente sin tener que pasar por el formulario de registro. Esto se conoce como "guardar la sesión" o "login automático" y contribuye a la agilidad de la aplicación.

B.1.2. Tabla de reportes de actuación

TAREAS												
rep_id	ini_fin	datos_ok	comments	date	time	lat	lng	img_image1	img_image2	img_image3	hay_img	img_ok

Tabla 2. Informes de actuación

A continuación se verá la tabla donde se almacena toda la información relativa a los informes relativos a las tareas (iniciación, finalización y validación).

- `rep_id`: identificador de tarea. Se trata de un número asignado a cada tarea de manera exclusiva (cada tarea tiene un número distinto) para identificarla, por lo que si tenemos varios informes (iniciación, finalización y validación) sobre la misma tarea, todos ellos tendrán el mismo identificador.
- `ini_fin`: marcador de tipo de informe. Puede tomar valores "ini" (si es un reporte de iniciación), "fin" (finalización) y "validar" (si se trata de una validación).
- `datos_ok`: registro de estado del envío de datos. Si este campo vale false, quiere decir que los datos del correspondiente formulario no han sido recibidos correctamente en el servidor, y por tanto el sistema tratará de reenviar este informe cuando corresponda. Si vale true, significa que los datos han llegado con éxito al servidor.
- `comments`: comentarios adicionales añadidos por el usuario en el formulario.
- `date` y `time`: fecha y hora en la que el informe fue redactado. Para evitar el falseamiento de esta información, son tomados automáticamente por el dispositivo y no se permite su modificación.
- `lat` y `lng`: coordenadas de geolocalización (latitud, longitud). Son también tomadas automáticamente por el dispositivo.
- `img_imageX`: estos tres campos almacenan la ruta y nombre (en el sistema de archivos local) de las imágenes adjuntas al reporte. Sirve para poder visualizar estas imágenes junto con el resto del informe posteriormente, además de para poder transferirlas.
- `hay_img`: registro de estado que indica si en el correspondiente informe se incluye, al menos, una imagen. En caso de que no haya imágenes adjuntas este campo valdrá false, de otro modo valdrá true.
- `img_ok`: registro de estado cuya función es similar a "datos_ok", pero relativo a las imágenes. Indica, por tanto, si las imágenes asociadas al reporte correspondiente han sido correctamente recibidas en el servidor (true) o si será necesario enviarlas posteriormente (false). La necesidad de dos registros de estado (uno para datos y otro para imágenes) viene impuesta por la separación en la transmisión de ambos elementos, ya que la transmisión de datos es más prioritaria y no requiere consideraciones aplicables a la transmisión de imágenes.

B.1.3. Tabla de nuevas tareas

NUEVAS TAREAS						
new_id	tareaID	descripcion	fecha_conocimiento	datos_ok	bloqueID	sectorID
contratoID	claseID	tipoID	subtipoID	provincia	carreteraID	calzadaID
PKinicial	PKfinal	fuentes_informacion	plazo_llegada	fecha_llegada	actuaciones_complementarias	observaciones
proc_gen	proc_esp	latitud	longitud	zoom	img_image1	img_image2
img_image3	userID_alta	ambitoID	ladoID	orientacionID	hay_img	img_ok

Tabla 3. Alta de nuevas tareas

En esta tabla quedarán registradas todas aquellas tareas que damos de alta rellenando el formulario correspondiente.

Como podemos observar, esta tabla contiene gran cantidad de campos ya que describe todos los detalles necesarios para dar de alta una nueva tarea en el servidor.

- new_id: es un identificador local de la tarea, pero no tendrá ninguna validez en el servidor.
- tareaID: es el identificador de tarea que aparecerá en la lista de tareas. Es exclusivo de cada tarea.
- descripción: detalles adicionales de la tarea.
- fecha_conocimiento: fecha tomada automáticamente al dar de alta una nueva incidencia.
- datos_ok: registro de estado cuya función es la misma que en el apartado anterior ([B.1.2. Tabla de reportes de actuación](#)).
- bloqueID: identificador de bloque. Bloque es la distinción entre las tareas programables y no programables (sólo damos de alta las no programables).
- sectorID: identificador de sector. Los sectores son las unidades en las que se divide la carretera.
- contratoID: identificador del contrato que regula la gestión de la carretera.
- claseID: identificador de la clase de la nueva incidencia.
- tipoID: identificador del tipo de la nueva incidencia.
- subtipoID: identificador del subtipo de la nueva incidencia.
- provincia: nombre de la provincia donde se originó la incidencia.
- carreteraID: identificador de la carretera donde se originó la incidencia.
- calzadaID: identificador de en qué plataforma (derecha, izquierda, mediana) se

originó la incidencia.

- PKinicial: punto kilométrico del inicio de la incidencia.
- PKfinal: punto kilométrico del final de la incidencia.
- fuente_informacion: indica quién informó de la incidencia.
- plazo_llegada: tiempo disponible (minutos) para realizar las actuaciones necesarias. Ese tiempo comienza a consumirse en cuanto se da de alta la tarea.
- fecha_llegada: fecha y hora en la que el operario deberá tener finalizada la actuación. Equivale a fecha_conocimiento + plazo_llegada.
- actuaciones_complementarias: protocolo de actuación establecido para una clase, tipo y subtipo.
- observaciones: otra información a aportar por el operario.
- proc_gen: procedimiento general de actuación del sector en el que se originó la incidencia.
- proc_esp: procedimiento específico de actuación asignado a un sector y un contrato.
- latitud y longitud: coordenadas de geolocalización de la incidencia.
- zoom: nivel de zoom con el que aparecerá el mapa de Google Maps que muestra la situación de la incidencia en la ficha de la lista de tareas.
- img_imageX: imágenes asociadas a la nueva incidencia. Guardan el nombre y la ruta de las imágenes para poder ser enviadas más tarde.
- userID_alta: identificador del usuario que da de alta la incidencia.
- ambitoID: identificador del ámbito.
- ladoID: identificador del lado.
- orientacionID: identificador de la orientación. Junto con el ámbito y el lado, conforma la georeferencia transversal.
- hay_img: registro de estado que vale true si alguno de los campos img_imageX contiene la ruta de una imagen. Vale false si todos los campos están vacíos (es decir, el formulario no tiene imágenes adjuntas).
- img_ok: registro de estado que indica si las imágenes han sido correctamente enviadas al servidor.

B.1.4. Tabla de la lista de tareas

Esta será la tabla que almacena la lista de tareas que nos llega del servidor. Es la que contiene una mayor cantidad de campos ya que es la que más información necesita almacenar. Muchos campos son idénticos a los del apartado anterior ([B.1.3. Tabla de alta de nuevas tareas](#)) y por tanto sólo se comentarán aquellos que no estuvieran presentes en dicho apartado.

- inicio_actuacion: registro de estado que vale true si la incidencia había sido iniciada cuando descargamos la lista del servidor por última vez.

- `periodo_transcurrido`: minutos que han pasado desde la fecha de conocimiento hasta el momento actual.
- `dif_temporal`: control relativo a las diferencias horarias para evitar manipulaciones por parte de los usuarios. En pruebas (sin implementar).
- `comprobacion_presencia`: verificación de presencia física en la realización de la tarea. En pruebas (sin implementar).
- `userID_baja`: identificador del usuario que finalizó la tarea.
- `fin_actuacion`: registro de estado que vale true si la incidencia había sido finalizada cuando descargamos la lista del servidor por última vez.
- `calzada`: texto que nombra la parte de la vía en la que sucedió la incidencia (derecha, izquierda, mediana...).
- `validada`: registro de estado que vale true si la incidencia había sido validada cuando descargamos la lista del servidor por última vez.
- `pk_hito_ini`: hito kilométrico inicial.
- `pk_dst_hito_ini`: distancia al hito kilométrico inicial.
- `pk_hito_fin`: hito kilométrico final.
- `pk_dst_hito_fin`: distancia al hito kilométrico final.
- `tipo`: texto que nombra el tipo de incidencia. Esta información se complementa con su identificador (`tipoID`).
- `bloque`: texto que nombra el bloque de la incidencia. Esta información se complementa con su identificador (`bloqueID`).
- `clase`: texto que nombra la clase de incidencia. Esta información se complementa con su identificador (`claseID`).
- `contrato`: texto que nombra el contrato que regula la explotación de la carretera donde sucedió la incidencia. Esta información se complementa con su identificador (`contratoID`).
- `sector`: texto que nombra el sector donde sucedió la incidencia. Esta información se complementa con su identificador (`sectorID`).
- `permissionID`: identificador de los permisos de los usuarios que pueden actuar sobre la incidencia. Generalmente se relaciona con el sector.
- `carretera`: texto que describe el sector donde sucedió la incidencia. Esta información se complementa con su identificador (`sectorID`).
- `antelacion_aviso`: indicador relativo a las tareas programadas. Sirve para mostrar un aviso antes de que aparezca la tarea programada.
- `fecha_inicial`: indicador relativo a las tareas programadas. Fecha en la que se realizó la programación de la tarea.
- `modo_repeticion`: indicador relativo a las tareas programadas. Frecuencia con la que se repite una tarea programada (anual, mensual...).
- `img_rep_ini_status` y `img_rep_fin_status`: registros de estado que controlan si

el reporte de iniciación y finalización, respectivamente, fueron redactados o no, bien sea porque al descargar la lista del servidor ya estaban redactados o bien porque el usuario los ha redactado sin haber podido enviar la información al servidor. Controlan los iconos que muestran el estado de los reportes (terminado o no) en la lista de tareas y en la ficha.

B.2. Caché local

Al igual que los navegadores de Internet, las aplicaciones Android permiten el almacenamiento de datos en una memoria caché local. Existen dos tipos en esta memoria, las variables de sesión y las variables locales. Aunque dichos nombres son una herencia del uso de este tipo de variables en los navegadores, siguen teniendo vigencia para las aplicaciones móviles

Mientras que las variables de sesión, como su nombre indica, están asociadas a la sesión del navegador y su contenido se volatilizará cuando cerremos dicha pestaña del navegador, las segundas permanecen almacenadas hasta que el sistema reciba la orden de borrarlas (el usuario puede programar esta acción). De esta manera, con la memoria caché local conseguimos un almacenamiento permanente muy útil que será utilizado a modo de variables globales en la aplicación.

Las variables más relevantes de este tipo se comentan a continuación:

- `localStorage.pfc_sector_val`: guarda el sector seleccionado por el usuario en la configuración.
- `localStorage.pfc_tramos_val`: guarda los tramos seleccionados por el usuario en la configuración.
- `localStorage.pfc_solo_wifi`: indica si las imágenes se enviarán sólo cuando esté disponible una red wifi o por el contrario se tratará de utilizar también una red 3G.
- `localStorage.pfc_f_act_lista`: frecuencia de actualización automática de la lista.
- `localStorage.pfc_auto_lista`: indica si la actualización automática de la lista está activada o no.
- `localStorage.pfc_f_act_envios`: frecuencia de envío automático de la información pendiente.
- `localStorage.pfc_auto_envios`: indica si el envío automático de la información pendiente está activado o no.
- `localStorage.pfc_inicio`: indica si es la primera vez que ejecutamos la aplicación tras instalarla (para realizar los ajustes iniciales).
- `localStorage.pfc_tabla2_existe`: indica si la lista de tareas no existe o existe pero está vacía.
- `localStorage.pfc_img_width`: contiene el valor, en píxels, del ancho de las imágenes tomadas por la cámara en el entorno de la aplicación.
- `localStorage.pfc_img_height`: contiene el valor, en píxels, del alto de las imágenes tomadas por la cámara en el entorno de la aplicación.

- `localStorage.pfc_fuente_infoID`: para el usuario registrado, contiene la lista de agentes que pueden dar de alta una nueva tarea.
- `localStorage.pfc_userID`: identificador del usuario registrado.
- `localStorage.pfc_permisos`: permisos del usuario registrado.
- `localStorage.pfc_reporte_hecho`: indica si el informe seleccionado ya había sido rellenado.
- `localStorage.pfc_inicio`: indica si es la primera vez que ejecutamos la aplicación.

B.3. Librería de la aplicación

Como ya se ha adelantado anteriormente, durante el desarrollo del código de la aplicación de este PFC se fueron creando una serie de funciones genéricas que podrían aprovecharse en futuros trabajos. Otras, se crearon a partir de la generalización de diversos módulos muy similares que cumplían funciones análogas (aunque con ligeras diferencias). Simplemente añadiendo parámetros de configuración a la función, obteníamos un módulo de gran potencia dada su versatilidad, de modo que el programador puede controlar el funcionamiento de esa función. Esas diversas funciones de la misma familia se sustituyeron por unas pocas que diferían ligeramente en su comportamiento en función de los parámetros al ser llamadas.

A continuación se citan algunas de esas funciones, explicando su cometido y el por qué de su relevancia.

B.3.1. Funciones genéricas para la gestión de la base de datos local

insert_Sql (`table_name`, `columns_names`, `data_values`, `tx`)

Inserta una nueva fila de datos en la tabla.

table_name: string con el nombre de la tabla.

columns_names: array que contiene los nombres de las columnas que queremos rellenar.

data_values: array con los valores a insertar. Debe estar ordenado con respecto a "*columns_names*".

tx: objeto "tx" de la transacción.

create_Sql (`table_name`, `columns_names`, `tx`)

Crea una nueva tabla si no existía previamente.

table_name: string con el nombre de la tabla.

columns_names: array que contiene los nombres de las columnas de nuestra nueva tabla.

tx: objeto "tx" de la transacción.

save (`table_name`, `columns_names`, `saved_values`, `save_callback_success`, `save_callback_error`)

Esta función intenta insertar un conjunto de datos en una tabla y si tiene éxito ejecuta la función *save_callback_success*. Si falla, llama a la función *save_callback_error* para capturar el error.

table_name: string con el nombre de la tabla donde guardaremos los datos.

columns_names: array que contiene los nombres de las columnas que queremos rellenar.

saved_values: array con los valores a insertar. Debe estar ordenado con respecto a "columns_names".

save_callback_success: función llamada si tenemos éxito al guardar la información.

save_callback_error: función llamada si algo va mal durante la ejecución de "save". Recibirá un parámetro err, que contiene las propiedades "code" y "message".

extract (*table_name*, *columns_names*, *where_clause*, *extract_callback_success*, *extract_callback_error*)

Esta función extrae información de una tabla, teniendo la opción de incluir una cláusula condicional. Si la consulta tiene éxito se ejecutará la función *extract_callback_success*. Si falla, llama a la función *extract_callback_error* para capturar el error.

table_name: string con el nombre de la tabla de donde extraemos los datos.

columns_names: array que contiene los nombres de las columnas de donde queremos extraer.

where_clause: en principio, es un string que indica las condiciones que deben de cumplir los datos a extraer (p. ej.: campo ID=12). Si no se desea una búsqueda condicional, "where_clause" deberá tomar el valor booleano "false".

extract_callback_success: función llamada si tenemos éxito al extraer la información. Recibirá tres parámetros:

- *extracted_info_array*: array bidimensional de índice numérico con la información extraída.
- *extracted_indexes*: array de índice numérico con los índices (nombres de las columnas).
- *extracted_info_obj*: objeto de índice numérico en su primera dimensión y asociativo en su segunda (con los índices "extracted_indexes") que contiene la información extraída.

extract_callback_error: función llamada si algún error ocurre durante la extracción. Recibirá un parámetro err, que contiene las propiedades "code" y "message".

drop (*array_tables*, *drop_callback_success*, *drop_callback_error*)

Esta función elimina, si existen, las tablas recibidas como parámetro.

array_tables: array de strings que contiene los nombres de las tablas a eliminar.

drop_callback_success: función de callback llamada en caso de que la operación tenga éxito.

drop_callback_error: función de callback llamada en caso de que la eliminación falle.

B.3.2. Funciones específicas

auto_login (*success_callback*, *tipo_llamada*, *no_conexion_callback*)

Esta función realiza un registro transparente al usuario, de manera que continuará ejecutando la función que se le pasa como parámetro (*success_callback*) si es capaz de hacer el login correctamente.

success_callback: función llamada si el registro automático tiene éxito.

tipo_llamada: modo en el que trabaja la función (inicio, activo y pasivo o *background*)

no_conexion_callback: función llamada en caso de que no exista conexión a internet.

success_save (*tipo_llamada*, *tipo_llamada_auto_login*, *objetivo*, *envio_success_callback*)

Función que se ejecuta después de haber guardado los datos del formulario con éxito. Extrae la información asociada a los formularios que no han sido enviados y realiza el envío de los mismos.

tipo_llamada: modo en el que se ejecutará la función (nuevo envío, envío general, automático).

tipo_llamada_autologin: cuando llamemos a la función de *auto_login*, modo en el que se llamará ésta.

objetivo: indica si la información a enviar se trata de informes de iniciación/finalización, de una nueva alta o de una validación.

envio_success_callback: función llamada tras realizar el envío.

comprobar_envio_imagenes (*tipo_llamada*, *tipo_llamada_auto_login*, *objetivo*, *envio_success_callback*)

En esta función se comprueba si hay que enviar imágenes, es decir, por un lado hay que verificar que la conexión y configuración permiten el envío de imágenes y por otro lado hay que comprobar si hay imágenes en nuestra BD que deban ser enviadas. Después, llama a la función encargada de enviar las imágenes.

Los parámetros de esta función cumplen el mismo papel que en la anterior (*success_save*).

enviar_imagenes (*img_info_array*, *parametros_img*, *tipo_llamada*, *tipo_llamada_auto_login*, *objetivo*, *envio_success_callback*)

Esta función realiza el envío al servidor de las imágenes y procesa su respuesta para después llamar a la función de actualización de la base de datos local.

img_info_array: contiene las rutas a las imágenes que han de enviarse.

parametros_img: objeto que contiene una serie de parámetros sobre las imágenes que han de enviarse.

El resto de parámetros cumplen el mismo papel que en las anteriores funciones.

update_data (respuesta, parametros, tipo_llamada, tipo_llamada_auto_login, objetivo, envio_success_callback)

En esta función se realiza la actualización de la base de datos local, marcando como enviado aquello que ha sido recibido con éxito en el servidor.

respuesta: es la respuesta que nos llega del servidor, con ella determinaremos que datos han sido registrados en el mismo.

parametros: objeto que contiene una serie de parámetros sobre la información enviada.

envio_success_callback: función llamada al finalizar la ejecución.

El resto de parámetros cumplen el mismo papel que en las anteriores funciones.

Modos de ejecución

Hemos visto que estas funciones incluían parámetros que eran descritos como "modo en el que se ejecutará la función". La razón de estos parámetros es disponer de una manera de modificar ligeramente el comportamiento de la función dependiendo del contexto en el que se ejecute.

Pongamos como ejemplo la función de envío de la información (*success_save*). Existen tres posibles escenarios en el que dicha función se ejecutará:

- El usuario ha rellenado un formulario y ha pulsado en enviar. En este caso, mientras se produce el envío se muestra en pantalla un mensaje de espera, y al finalizar se informa del éxito o fracaso del intento y se vuelve a la pantalla anterior.
- El usuario, desde el submenú de opciones, ha seleccionado enviar toda la información de golpe. En este caso, necesitamos seguir mostrando un mensaje mientras se producen los envíos. Pero sería deseable que no se mostrara un mensaje indicando éxito o fracaso por cada uno de los formularios pendientes (en caso de tener un gran número esto sería tedioso) sino mostrar solamente un mensaje al final, con un resumen de la información que ha podido ser enviada y los errores producidos. Permanecemos en la misma pantalla.
- Envío automático de la información. Esta tarea por definición se realiza de manera transparente al usuario para no interferir con la actividad que éste realizase en el momento. Por tanto, no debe mostrar ningún tipo de mensaje (ni siquiera aunque no haya conexión, en ese momento simplemente abortará el intento) ni tampoco cambiar de pantalla. Debe ser indetectable para el usuario.

B.3.3. Funciones de propósito general

asToNum (arr)

Esta función toma un array asociativo (o un objeto) y devuelve un array igual pero de

índice numérico.

purga_array (arr1,pos,arr2)

Esta función toma un array numérico y elimina sus filas cuyo elemento en la posición "pos" vale "false".

get_ind (arr)

Esta función toma un array asociativo (o un objeto) y devuelve un array numérico con los nombres de sus índices.

tx_server (tx_url, tx_type, tx_dataType, tx_data, tx_server_success_callback, tx_server_error_callback).

Esta función realiza una llamada ajax al servidor para enviar datos.

tx_url: dirección URL del servidor.

tx_type: método de envío (POST/GET).

tx_dataType: formato de respuesta (JSON, html, script...)

tx_data: datos a enviar.

tx_success_callback: función llamada en caso de que el envío tenga éxito.

tx_error_callback: función llamada en caso de que el envío falle.

tx_server_images (image_URI, image_parameters, url_server_images, tx_server_images_success_callback, tx_server_images_error_callback)

Esta función realiza una llamada AJAX al servidor para enviar imágenes.

image_URI: ruta local donde se encuentra la imagen a enviar.

image_parameters: objeto que contiene información adicional sobre la imagen.

url_server_images: dirección URL del servidor.

tx_server_images_success_callback: función llamada en caso de que el envío tenga éxito.

tx_server_images_error_callback: función llamada en caso de que el envío falle.

get_fecha_hora ()

Devuelve la fecha y la hora locales (del cliente).

geoloc (tipo_llamada,callback)

Obtiene las coordenadas de la geolocalización y las coloca en <div id="lat"> y <div id="lng">, respectivamente.

tipo_llamada: modo en el que se ejecutará la función.

callback: función llamada en caso de conseguir establecer la geolocalización.

google_maps (lat,long,zoom)

Esta función nos inserta un mapa de google maps en <div id="mapa">.

lat: latitud del marcador que aparecerá en el centro del mapa.

long: longitud del marcador que aparecerá en el centro del mapa.

zoom: nivel de zoom que mostrará el mapa creado.