

**Aplicación para la gestión de incidencias en el equipamiento urbano de Sevilla.**

***Ramón Bajona***

***Antonio Guardia***

***Ignacio Martínez***

Resumen del trabajo

Aplicación desarrollada para dispositivos Android y página web, destinada a facilitar a los ciudadanos de Sevilla un canal de comunicación con el Ayuntamiento. Mediante esta aplicación los ciudadanos podrán poner en conocimiento del Ayuntamiento de Sevilla, las incidencias y/o problemas derivados del mantenimiento del mobiliario urbano. Los ciudadanos, tomarán una fotografía, indicarán la ubicación en el mapa y añadirán una breve descripción del problema, esta información se almacenará en una base de datos central. Por otro lado, los servicios municipales de mantenimiento, desde una sección privada de la web, podrán ver todas las incidencias enviadas por los ciudadanos e intentar solucionarlas en la mayor brevedad posible.

**Índice**

Introducción

Contexto y justificación del trabajo

Objetivos del trabajo

Enfoque y método seguido

Planificación del Trabajo

Breve sumario de productos obtenidos

Usuarios y contexto de uso

Perfiles de usuario

Contextos de usos

Análisis de tareas

Casos de uso

Implementación

Implementación web

Implementación app

Implementación API

Conclusiones

Conclusiones científicas

Conclusiones personales

Agradecimientos

1. **Introducción**
   1. **Contexto y justificación del Trabajo**

En cualquier localidad, uno de los principales problemas del Ayuntamiento es el mantenimiento de todas las infraestructuras. Dejando de lado los posibles problemas económicos que dificulten esta misión, el Ayuntamiento se encuentra con una gran dificultad, la que supone la detección de los problemas que debe solucionar, especialmente si son de poca importancia: una farola fundida, un banco de un parque en mal estado, desperfectos en la calle o acera, papeleras rotas…, etc.

Actualmente, estos problemas se detectan mediante inspecciones realizadas por el propio personal de mantenimiento del Ayuntamiento, o atendiendo a las quejas que pueden realizar los propios habitantes de Sevilla en los plenos ordinarios, o personándose ante los órganos competentes correspondientes. Estos métodos a menudo se presentan ineficaces y consumen un gasto de personal demasiado elevado.

Como solución, el servicio de Respuesta Urbana (072 REUR) o el Servicio de Parques y Jardines del Ayuntamiento de Sevilla, según competencias, acudiendo a la colaboración ciudadana mediante la aplicación móvil y página web, permitir a cualquier habitante de Sevilla la comunicación de cualquier incidencia en cualquier momento. Dichas incidencias se almacenarán en una base de datos centralizada, pudiendo el personal de mantenimiento acceder a ellas y comprometerse en la medida de lo posible a dar respuesta a las mismas en un plazo de 72 horas.

Como ventajas añadidas, con el uso de una aplicación móvil y una web de estas características, se fomenta la participación ciudadana en los asuntos municipales, y al proporcionar un feedback del estado de las quejas a lo largo del tiempo, el ciudadano se siente involucrado y escuchado por su ciudad.

Existen aplicaciones similares que podrían cumplir con los requerimientos que se desean, pero están enfocadas para municipios de mayor tamaño como pueden ser Madrid o Barcelona. Además muchas de estas aplicaciones no disponen de un organismo público que se encargue de dar respuesta a las incidencias enviadas por los usuarios. Por esto se decidió optar por la realización de una aplicación y una web sencilla, de fácil uso y mantenimiento, orientada a las necesidades de un municipio de menor tamaño.

* 1. **Objetivos del Trabajo**

El objetivo de este trabajo es el desarrollo de una aplicación Android, una página web y una base de datos centralizada, con la que pueda comunicarse la aplicación y la web mediante servicios API REST. La aplicación estará dirigida a dos tipos de usuarios, por un lado a un usuario ciudadano deSevilla y por otro a un usuario miembro del equipo de mantenimiento del Ayuntamiento.

* 1. **Enfoque y método seguido**

Como ya se ha mencionado anteriormente, existen aplicaciones similares en el mercado. El enfoque que se seguirá será crear una aplicación y web similar a las existentes pero a menor escala, orientada a su uso en una ciudad del tamaño de Sevilla.

* 1. **Planificación del Trabajo**

La planificación seguida en el proyecto es la siguiente:

|  |  |  |
| --- | --- | --- |
| **Tarea** | **Tarea secundaria** | **Duración** |
| **Formulario app** |  |  |
|  | Subir foto | 5 horas |
|  | Localización | 2 horas |
|  | Campos del formulario | 2 horas |
|  | Enviar información | 3 horas |
| **Mapa incidencias app** |  |  |
|  | Mostrar incidencias | 4 horas |
|  | Mostrar mapa acotado | 1 hora |
| **Base de datos** |  |  |
|  | Tabla Usuario | 2 horas |
|  | Tabla incidencia | 2 horas |
| **API** |  |  |
|  | GET | 2 horas |
|  | POST | 2 horas |
|  | PUT | 2 horas |
|  | DELETE | 1 hora |
| **Login app** |  |  |
|  | Conexión API | 3 horas |
|  | Comprobaciones | 3 horas |
|  | Formulario | 2 horas |
| **Formulario Registro app** |  |  |
|  | Conexión API | 1 hora |
|  | Comprobaciones | 3 horas |
|  | Formulario | 2 horas |
| **Mis incidencias app** |  |  |
|  | Mostrar info | 2 horas |
|  | Mostrar estado | 2 horas |
| **Perfil app** |  |  |
|  | Formulario | 2 horas |
|  | Conexión API | 1 hora |
|  | Comprobaciones | 2 horas |
| **Formulario web** |  |  |
|  | Subir foto | 4 horas |
|  | Localización | 2 horas |
|  | Campos del formulario | 2 horas |
|  | Enviar información | 2 horas |
| **Login web** |  |  |
|  | Conexión API | 1 horas |
|  | Comprobaciones | 2 horas |
|  | Formulario | 2 horas |
| **Formulario Registro web** |  |  |
|  | Conexión API | 1 hora |
|  | Comprobaciones | 2 horas |
|  | Formulario | 2 horas |
| **Mis incidencias web** |  |  |
|  | Mostrar info | 2 horas |
|  | Mostrar estado | 2 horas |
| **Perfil web** |  |  |
|  | Formulario | 2 horas |
|  | Conexión API | 1 hora |
|  | Comprobaciones | 2 hora |
| **Subir incidencia app** |  |  |
|  | Formulario | 2 hora |
|  | Conexión API | 1 hora |
|  | Comprobaciones | 2 horas |
| **Subir incidencia app** |  |  |
|  | Formulario | 2 horas |
|  | Conexión API | 1 hora |
|  | Comprobaciones | 2 horas |
| **Diseño** |  |  |
|  | App | 3 horas |
|  | Web | 8 horas |
| **Inicio web** |  |  |
|  | Menú inicio | 5 horas |
| **Documentación** |  |  |
|  | Documentación | 4 horas |

Para la realización del proyecto, se contará con las siguientes

herramientas:

* Android Studio
* Netbeans
* MySQL
* Python
  1. **Breve sumario de productos obtenidos**
* Código fuente de la aplicación móvil.
* Código fuente de la página web.
* Ejecutable (.apk) de la aplicación.
* Estructura de la base de datos del servidor.
* Memoria del proyecto.

1. **Usuarios y contexto de uso**

En este punto, vamos a analizar cuáles son las características de los usuarios, sus objetivos y contexto de uso de la aplicación, con el fin de detectar las necesidades para satisfacer a los usuarios. Para ello, haremos un estudio de quiénes serán los usuarios de la aplicación, qué tareas realizarán y cómo utilizarán el sistema.

* 1. **Perfiles de usuario**

Nos encontramos con dos perfiles de usuarios que usarán la aplicación:

Miembros del equipo de mantenimiento del Ayuntamiento

- Franja de edad de los 20 a los 60 años.

- Acostumbrados al uso del móvil en su entorno de trabajo, envío de mensajes, correo electrónico, etc.

- No se prevén problemas en su adaptación al uso de la aplicación.

- Generalmente será siempre la misma persona.

Ciudadanos del municipio

Distintas franjas de edad:

- De 15 a 25 años: Son los más acostumbrados al uso de aplicaciones móviles y nuevas tecnologías. Es el rango de edad que presenta menos interés en los problemas del municipio.

- De 25 a 50 años: En su mayor parte disponen de un Smartphone, y lo usan habitualmente. No están tan acostumbrados a los cambios como el rango anterior, pero se adaptan a las nuevas tecnologías, con un breve periodo de aprendizaje.

- Mayores de 50 años: Es el rango de edad más difícil para la implantación de una aplicación móvil, ya que no suelen estar acostumbrados a las nuevas tecnologías. Hay que cuidar las interfaces, de manera que presenten iconos grandes, no contenga mucha información y sean lo más sencillos posibles.

* 1. **Contextos de Uso**

Se han identificado los siguientes contextos de uso:

- Un usuario, miembro del equipo de mantenimiento del Ayuntamiento usará la aplicación a lo largo de su jornada laboral para dar solución a las incidencias que indiquen los ciudadanos.

- Un ciudadano cualquiera, en cualquier momento del día, puede observar un problema en el mobiliario urbano de la ciudad y usar la aplicación para comunicarlo.

Independientemente del contexto de uso, para su funcionamiento la aplicación requerirá que el dispositivo móvil disponga de conexión a internet.

* 1. **Análisis de tareas**

Estas son las tareas que pueden realizar los usuarios con la aplicación y la web:

-Iniciar Sesión (Login): El usuario deberá introducir su correo electrónico y su contraseña para acceder a la app/web.

-Registro de usuario: en el caso que un usuario acceda por primera vez a la app/web deberá rellenar un formulario de registro con sus datos personales tales como nombre, apellidos, email y contraseña. Una vez validados los datos se procederá a registrar al nuevo usuario, el cual podrá acceder a la app/web por medio del inicio de sesión.

- Comunicar un problema: Al usuario se le indicará que proporcione una fotografía del problema, bien tomándola en ese momento, o bien cogiéndola de la galería. Seguidamente, se le mostrará un mapa del municipio, centrado en la posición actual (de ser posible), para que indique la posición del problema o introducir manualmente la dirección. Por último, se le pedirá una breve descripción del problema y se almacenarán todos estos datos.

- Consultar el estado de un problema: La aplicación le presentará al usuario una lista con los problemas pendientes, (informados por él, si es un usuario normal, o por cualquiera si forma parte del equipo de mantenimiento) en esta lista podrá ver datos asociados a la incidencia.

-Consultar incidencias en el mapa (solo disponible desde la app): Un usuario podrá consultar las incidencias en un mapa.

- Cambiar el estado de un problema (Solo disponible desde la web): Un usuario miembro del equipo de mantenimiento podrá cambiar el estado de un problema entre: enviado, en curso, resuelto o descartado.

-Consultar correos electrónicos del equipo de ReparaSevilla (solo disponible desde la app): Un usuario podrá consultar los correos electrónicos de ReparaSevilla.

-Cambiar los datos de usuario: Un usuario podrá modificar sus datos personales tales como el email, contraseña para acceder a la app/web, nombre, apellidos y foto de perfil.

- Cerrar sesión: Un usuario podrá terminar la sesión.

* 1. **Casos de uso**

Estas son los casos de uso que pueden realizar los usuarios con la aplicación y la web:

**Aplicación Móvil ReparaSevilla**

A continuación se detallan el conjunto de casos de uso que intervienen en

la aplicación móvil.

Caso de uso **Iniciar sesión (Login):**

* + Resumen de la funcionalidad: El usuario deberá introducir su correo electrónico y su contraseña para acceder a la app.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Conocer los datos para la conexión.
  + Poscondición: Se inicia la sesión.
  + Proceso normal principal:
  1. El Usuario introduce su email y contraseña.
  2. El Usuario pulsa el botón de ‘Iniciar sesión’ para enviar los credenciales a ReparaSevilla.
  3. ReparaSevilla devuelve una conexión válida.
  + Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error de conexión.

Caso de uso **Registro de usuario:**

* + Resumen de la funcionalidad: en el caso que un usuario acceda por primera vez a la app deberá rellenar un formulario de registro con sus datos personales tales como nombre, apellidos, email y contraseña. Una vez validados los datos se procederá a registrar al nuevo usuario, el cual podrá acceder a la app por medio del inicio de sesión.
  + Actores: Usuario.
  + Casos de uso relacionados: Iniciar sesión (Login).
  + Precondición: El usuario no puede estar ya registrado.
  + Poscondición: Se registra el nuevo usuario.
  + Proceso normal principal:

1. El nuevo Usuario introduce su email, contraseña nombre y apellidos.
2. El Usuario pulsa el botón de ‘Registrarse’ para enviar los datos a ReparaSevilla.
3. ReparaSevilla registra el nuevo Usuario.
   * Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error al registrar el nuevo Usuario.

Caso de uso **Subir incidencia:**

* + Resumen de la funcionalidad: Al usuario se le indicará que proporcione una fotografía del problema, bien tomándola en ese momento, o bien cogiéndola de la galería. Seguidamente, se le mostrará un mapa del municipio, centrado en la posición actual (de ser posible), para que indique la posición del problema o introducir manualmente la dirección. Por último, se le pedirá una breve descripción del problema y se almacenarán todos estos datos.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Estar logueado en la app.
  + Poscondición: Subir una nueva incidencia.
  + Proceso normal principal:

1. El Usuario introduce la foto del problema y la descripción de éste.
2. ReparaSevilla tomará la localización del dispositivo o el usuario introduce la dirección manualmente.
3. El Usuario pulsa el botón de ‘Subir incidencia’ para enviar los datos a ReparaSevilla.
4. ReparaSevilla registra la incidencia.
   * Alternativas de proceso y excepciones:

4a. ReparaSevilla devuelve un error al subir la incidencia.

Caso de uso **Mis incidencias:**

* + Resumen de la funcionalidad: La aplicación le presentará al usuario una lista con los problemas informados por él, donde podrá ver datos asociados a la incidencia.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Haber introducido una incidencia previamente.
  + Poscondición: Ninguna.
  + Proceso normal principal:

1. El Usuario consulta la dirección, descripción y estado de las incidencias subidas por él.
   * Alternativas de proceso y excepciones:

1a. ReparaSevilla devuelve un error de conexión.

Caso de uso **Mapa de Incidencias:**

* + Resumen de la funcionalidad: Un usuario podrá ver la posición de las incidencias enviadas por todos los Usuarios de ReparaSevilla en un mapa.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Ninguna.
  + Poscondición: Ninguna.
  + Proceso normal principal:

1. El Usuario consulta la posición de todas las incidencias en el mapa.
   * Alternativas de proceso y excepciones:

1a. ReparaSevilla devuelve un error de conexión.

Caso de uso **Mi perfil:**

* + Resumen de la funcionalidad: Un usuario podrá modificar sus datos personales tales como contraseña para acceder a la app, nombre, apellidos y foto de perfil.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Conocer los datos para del Usuario.
  + Poscondición: Modificar los datos del Usuario.
  + Proceso normal principal:

1. El Usuario modifica los datos deseados.
2. El Usuario pulsa el botón de ‘Modificar’ para enviar los nuevos datos a ReparaSevilla.
3. ReparaSevilla modifica los datos del Usuario.
   * Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error al modificar los datos.

Caso de uso **Contáctanos:**

* + Resumen de la funcionalidad: Un usuario podrá consultar los correos electrónicos de ReparaSevilla.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Ninguna.
  + Poscondición: Ninguna.
  + Proceso normal principal:

1. El Usuario consulta los emails.
   * Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error de conexión.

Caso de uso **Cerrar sesión:**

* + Resumen de la funcionalidad: Un usuario podrá terminar la sesión.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Iniciar Sesión (Login).
  + Precondición: Tener iniciada la sesión de Usuario.
  + Poscondición: Terminar la sesión.
  + Proceso normal principal:

1. El Usuario pulsa sobre el botón ‘Cerrar sesión’.
2. El Usuario es redirigido al caso de uso Iniciar Sesión (Login).
   * Alternativas de proceso y excepciones:

2a. ReparaSevilla devuelve un error al cerrar sesión.

**Página Web ReparaSevilla**

A continuación se detallan el conjunto de casos de uso que intervienen en la página web.

Caso de uso **Iniciar sesión (Login):**

* + Resumen de la funcionalidad: El usuario deberá introducir su correo electrónico y su contraseña para acceder a la web.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Conocer los datos para la conexión.
  + Poscondición: Se inicia la sesión.
  + Proceso normal principal:

1. El Usuario introduce su email y contraseña.
2. El Usuario pulsa el botón de ‘Iniciar sesión’ para enviar los credenciales a ReparaSevilla.
3. ReparaSevilla devuelve una conexión válida.
   * Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error de conexión.

Caso de uso **Registro de usuario:**

* + Resumen de la funcionalidad: en el caso que un usuario acceda por primera vez a la web deberá rellenar un formulario de registro con sus datos personales tales como nombre, apellidos, email y contraseña. Una vez validados los datos se procederá a registrar al nuevo usuario, el cual podrá acceder a la web por medio del inicio de sesión.
  + Actores: Usuario.
  + Casos de uso relacionados: Iniciar sesión (Login).
  + Precondición: El usuario no puede estar ya registrado.
  + Poscondición: Se registra el nuevo usuario.
  + Proceso normal principal:

1. El nuevo Usuario introduce su email, contraseña nombre y apellidos.
2. El Usuario pulsa el botón de ‘Registrarse’ para enviar los datos a ReparaSevilla.
3. ReparaSevilla registra el nuevo Usuario.
   * Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error al registrar el nuevo Usuario.

Caso de uso **Subir incidencia:**

* + Resumen de la funcionalidad: Al usuario se le indicará que proporcione una fotografía del problema, cogiéndola desde una ruta de su dispositivo. Seguidamente, se le mostrará un mapa del municipio, centrado en la posición actual (de ser posible), para que indique la posición del problema o introducir manualmente la dirección. Por último, se le pedirá una breve descripción del problema y se almacenarán todos estos datos.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Estar logueado en la web.
  + Poscondición: Subir una nueva incidencia.
  + Proceso normal principal:

1. El Usuario introduce la foto del problema y la descripción de éste.
2. ReparaSevilla tomará la localización del dispositivo o el usuario introduce la dirección manualmente.
3. El Usuario pulsa el botón de ‘Subir incidencia’ para enviar los datos a ReparaSevilla.
4. ReparaSevilla registra la incidencia.
   * Alternativas de proceso y excepciones:

4a. ReparaSevilla devuelve un error al subir la incidencia.

Caso de uso **Mis incidencias:**

* + Resumen de la funcionalidad: La web le presentará al usuario una lista con los problemas informados por él, donde podrá ver datos asociados a la incidencia.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Haber introducido una incidencia previamente.
  + Poscondición: Ninguna.
  + Proceso normal principal:

1. El Usuario consulta la dirección, descripción y estado de las incidencias subidas por él.
   * Alternativas de proceso y excepciones:

1a. ReparaSevilla devuelve un error de conexión.

Caso de uso **Mi perfil:**

* + Resumen de la funcionalidad: Un usuario podrá modificar sus datos personales tales como contraseña para acceder a la web, nombre, apellidos y foto de perfil.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Conocer los datos para del Usuario.
  + Poscondición: Modificar los datos del Usuario.
  + Proceso normal principal:

1. El Usuario modifica los datos deseados.
2. El Usuario pulsa el botón de ‘Modificar’ para enviar los nuevos datos a ReparaSevilla.
3. ReparaSevilla modifica los datos del Usuario.
   * Alternativas de proceso y excepciones:

3a. ReparaSevilla devuelve un error al modificar los datos.

Caso de uso **Panel ADMIN:**

* + Resumen de la funcionalidad: Un usuario miembro del equipo de mantenimiento (ADMIN) podrá cambiar el estado de cualquier incidencia entre: descartado, en revisión, en reparación y reparado.
  + Actores: Admin.
  + Casos de uso relacionados: Ninguno
  + Precondición: Ser un usuario Admin.
  + Poscondición: Cambiar estado de una incidencia.
  + Proceso normal principal:

1. El Admin consulta una incidencia.
2. El Admin selecciona un estado para la incidencia.
3. El Admin pulsa el botón ‘Modificar estado’.
4. ReparaSevilla modifica el estado de la incidencia.
   * Alternativas de proceso y excepciones:

4a. ReparaSevilla devuelve un error al modificar el estado de la incidencia.

Caso de uso **Cerrar sesión:**

* + Resumen de la funcionalidad: Un usuario podrá terminar la sesión.
  + Actores: Usuario/Admin.
  + Casos de uso relacionados: Iniciar Sesión (Login).
  + Precondición: Tener iniciada la sesión de Usuario.
  + Poscondición: Terminar la sesión.
  + Proceso normal principal:

1. El Usuario pulsa sobre el botón ‘Cerrar sesión’.
2. El Usuario es redirigido al caso de uso Iniciar Sesión (Login).
   * Alternativas de proceso y excepciones:

2a. ReparaSevilla devuelve un error al cerrar sesión.

**3. Implementación**

**3.1 Implementación web**

Index.py

Es nuestro archivo principal de la web hay que iniciar este archivo con python index.py y el se encarga de montar el servidor web y establecer los métodos con las rutas hacia cada pagina de la web. Aquí también realizamos la mayoría de las peticiones a la api para obtener los resultados antes de cargar las páginas.

login.py

Aquí cargamos en variables el contenido html y css y lo devolvemos a index.py.

formularioRegistro.py

En este archivo cargamos las variables html y css con el contenido del formulario y lo devolvemos a index.py.

Inicio.py

En este archivo solo cargamos en variables el contenido html y css y lo devolvemos a index.py para que cuando vallamos a la ruta de inicio se cargue este html.

subirIncidencias.py

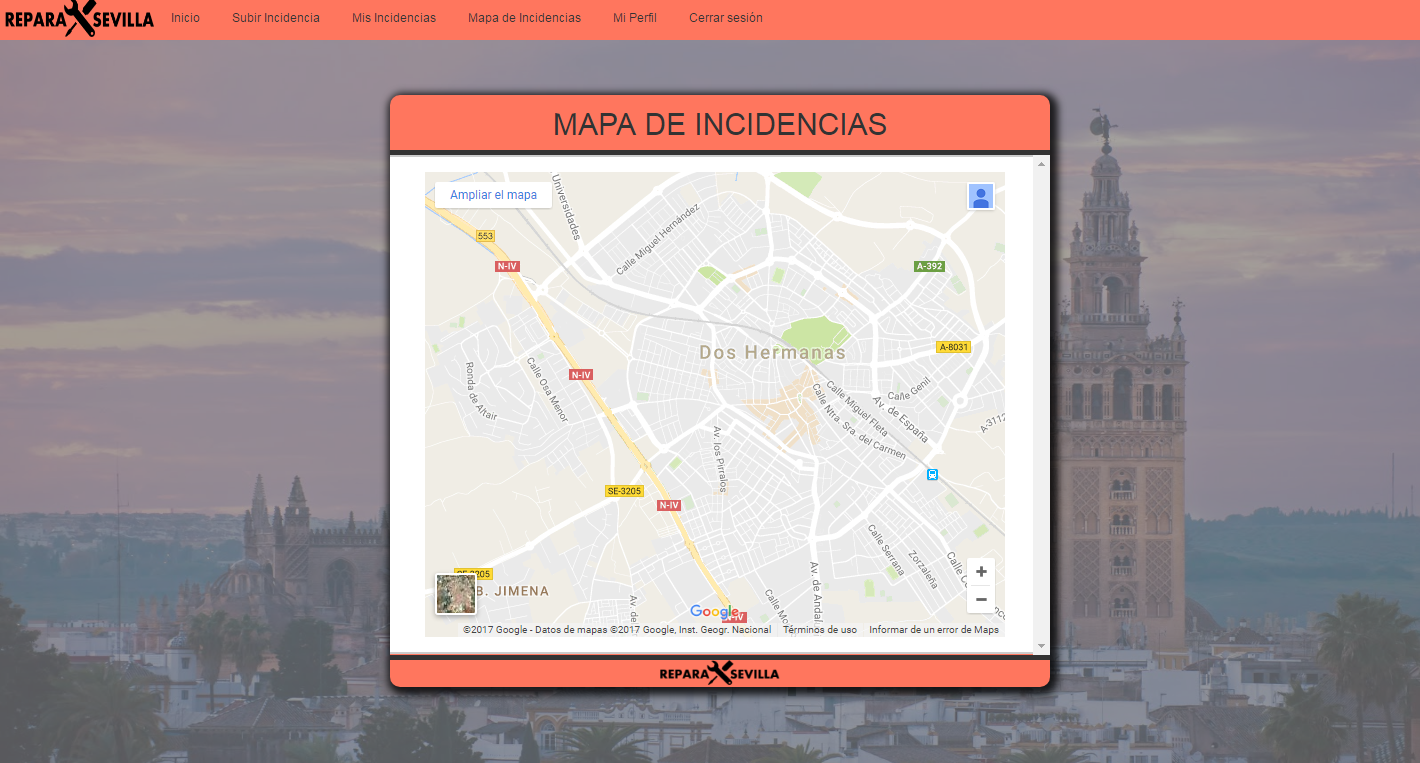
Aquí cargamos en variables el contenido html y css y lo devolvemos a index.py.

misIncidencias.py

En este archivo además de cargar las variables html y css con el contenido de la web mandamos una petición a la api para poder ir rellenando el html que generamos con los datos.

mapaIncidencias.py (Funcionalidad disponible próximamente)

En este archivo cargamos en variables el html , el css y el javascript en el cual debería cargar las macas de las incidencias en el mapa.



editarPerfil.py

En este archivo además de cargar las variables html y css con el contenido web mandamos una petición a la api para rellenar los campos del formulario con los datos del usuario.

adminIncidencias.py (Funcionalidad disponible próximamente)

Aquí cargamos el contenido html y css de la web en variables y lo devolvemos a index.py.

**3.2 Implementación app móvil**

HttpClient.java

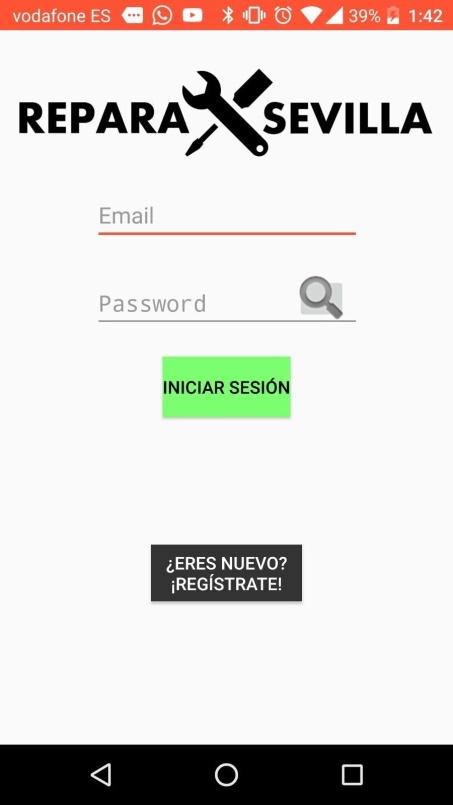
Esta clase se encarga de mandar peticiones post a la api y recibir la informacion que nos manda esta, es una clase por defecto de la librería HttpClient.

OperacionesApi.java

En la clase de operaciones api hacemos los diferentes metodos para gestionar las peticiones post y recibir los json, tambien se convierten estos json en los diferentes tipos de datos que vallamos a necesitar

login.java

Aqui recojemos los datos de los editText para mas tarde recoger de la api los emails y comprobar que existe, si existe entonces recoje los datos del usuario de la api con el email para verificar que la contraseña es igual

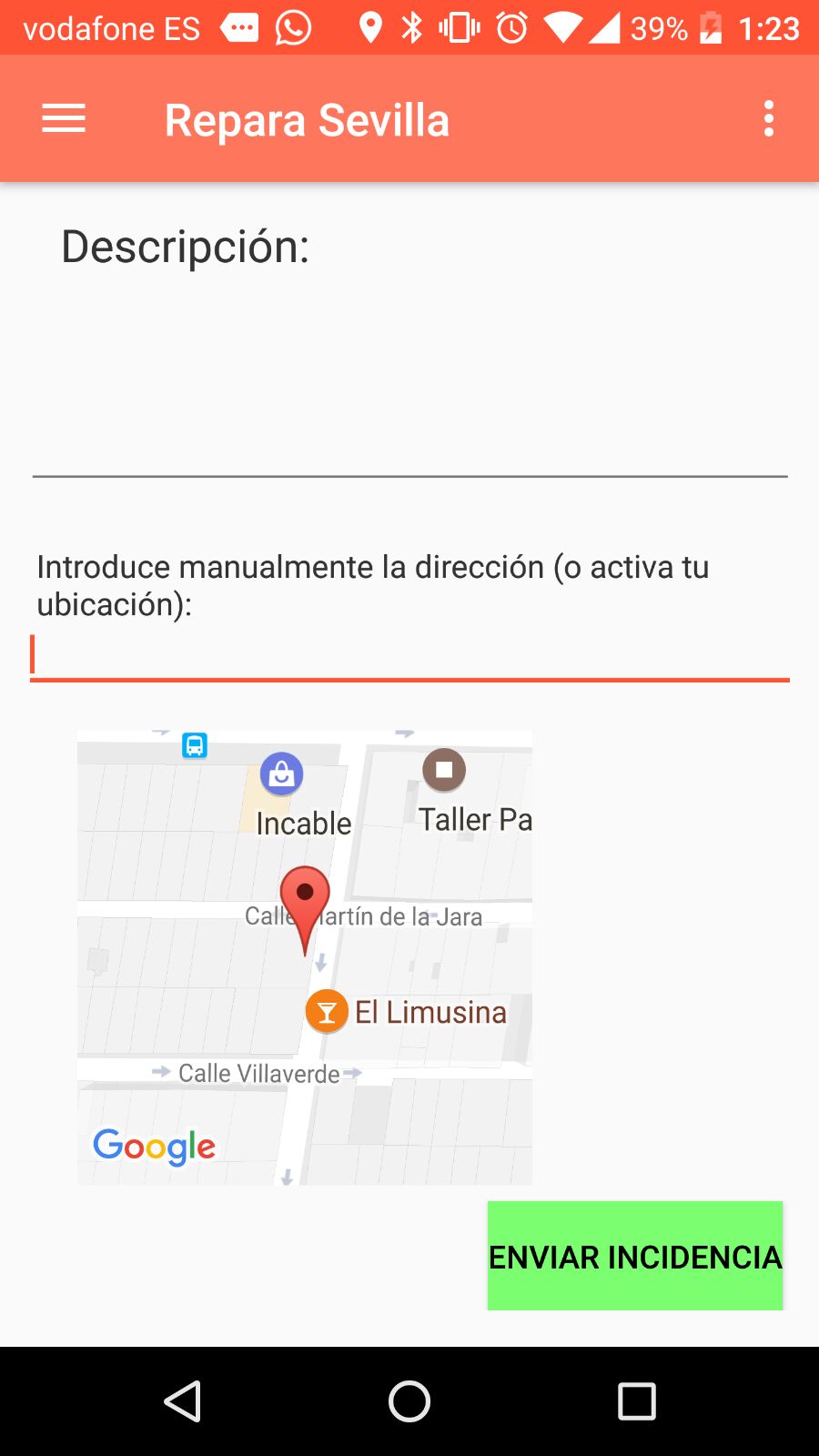


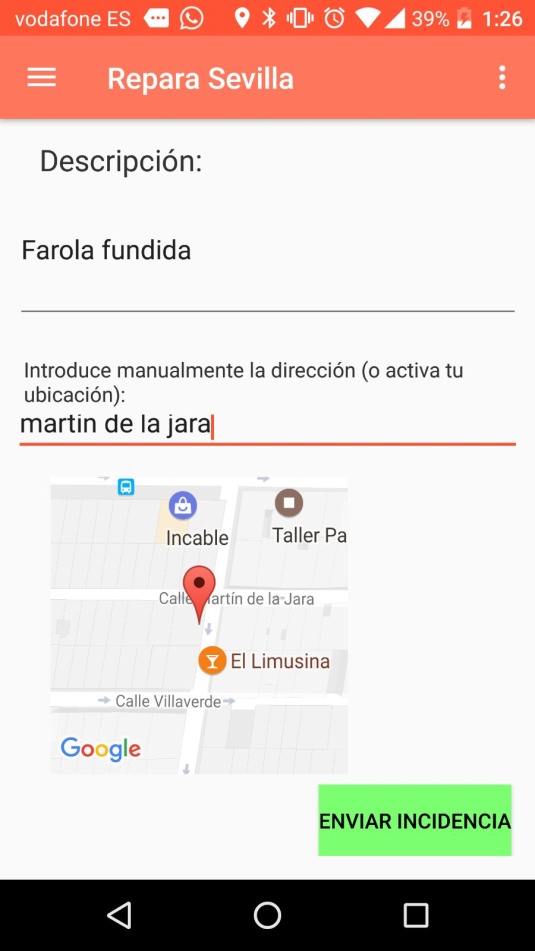
pantallaPrincipal.java

Esta clase recoge el objeto usuario que envía el login para asignárselo como propiedad de activity y va instanciando los diferentes fragments de la aplicación pasándoles el objeto usuario

subirIncidencias.java

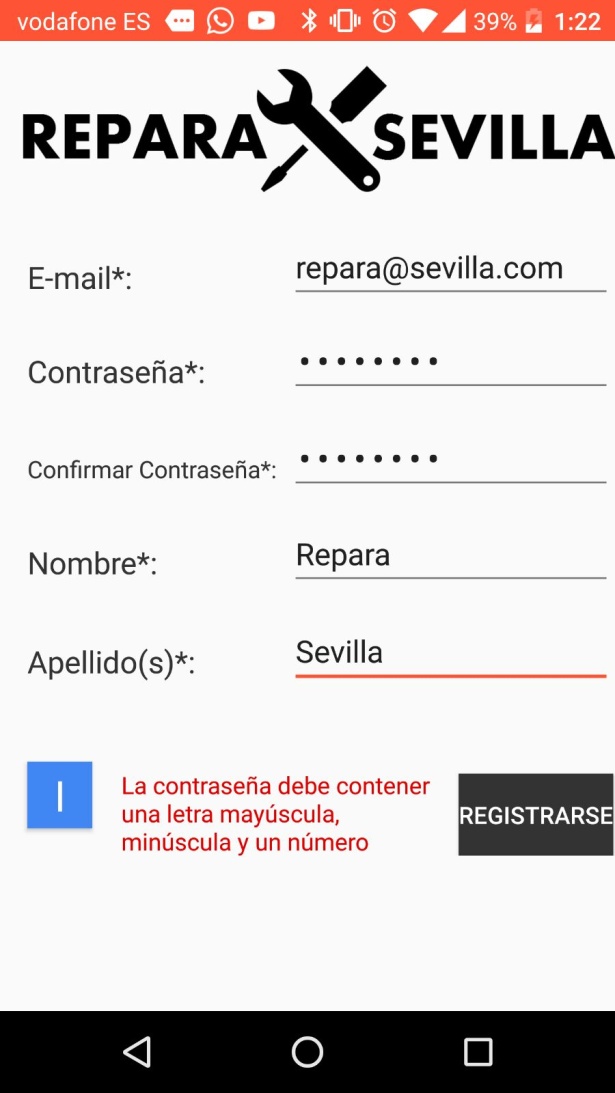
Esta clase se encarga de coger los datos del formulario subirIncidencias para mandarlos con una petición post a la api, esta petición se hace a partir del correo que obtuvimos en el login.





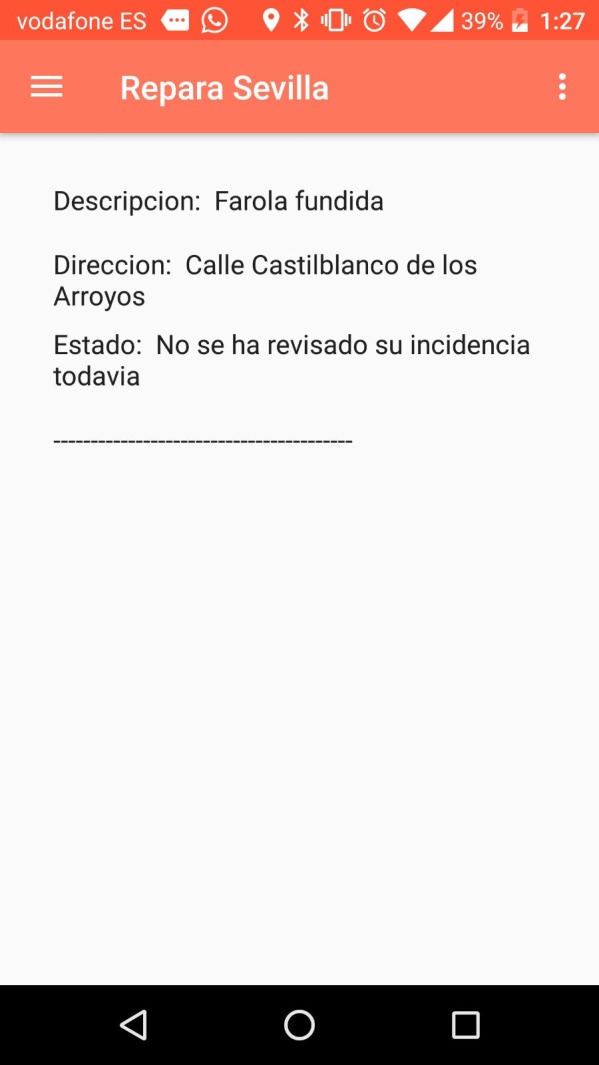
NuevoUsuario.java

Esta clase se encarga de recoger los datos del formulario de registro y hacer las diferentes comprobaciones para mandar una peticion post con el nuevo usuario.



MisIncidencias.java

En esta clase hacemos una peticion post a la api para a partir del correo que nos llega desde pantalla principal recibir un json con todas las incidencias del usuario.



miPerfiljava

Esta clase se encarga de recoger los datos del formulario de miPerfil para mandar una peticion post a la api con la que actualizar el usuario



map\_formulario.java

Aqui mostramos el mapa con las incidencias que obtenemos con una peticion post a la api en la que recibimos todas las incidencias del correo que nos viene desde pantallaPrincipal



contacto.java

En esta clase solo instanciamos un layout



Usuario.java

Esta es la clase con la que se hace el objeto usuario

Incidencias.java

Esta es la clase con la que se hace el objeto incidencias

3.3 **Implementación API**

api.py

En el archivo api.py esta lo que sería nuestra api restful método tiene una ruta asignada y dependiendo de la ruta a la que se haga la petición realiza las sentencias a la base de datos y manda devuelta los objetos json correspondientes

**4 Conclusiones**

**4.1 Conclusiones científicas**

Como todo proyecto, el principal objetivo es satisfacer las necesidades del

cliente, en este caso, la Escuela Profesional Vedruna, que pedía el desarrollo de un producto capaz de utilizar los conocimientos adquiridos durante los dos años del Grado Superior de Desarrollo de Aplicaciones Multiplataforma. Para lograr dicho objetivo, se ha recurrido a una planificación validada por el consultor del proyecto, Eliseo Molero. Las tareas marcadas se han cumplido en el plazo previsto por lo que no ha existido una desviación de tiempo en el desarrollo del producto.

Por otro lado, este desarrollo ha implicado el aprendizaje de conocimientos sobre las tecnologías vinculadas a Android.

**4.2 Conclusiones personales**

Desde el punto de vista del grupo, el proyecto ha sido un gran reto para nosotros.

El Proyecto de Desarrollo de Aplicaciones Multiplataforma es la culminación de un largo trabajo, horas y horas de esfuerzo traducidos en una asignatura en la que se deben poner en práctica todos los conocimientos adquiridos durante los dos años del grado.

Aun teniendo en cuenta la gran variedad de opciones a escoger, consideramos oportuno hacer un proyecto basado en Android y Python para poder ampliar nuestros conocimientos técnicos en ambos campos.

Estos últimos meses han implicado un gran esfuerzo personal por parte de todos los componentes del grupo. Días enteros de duro trabajo, noches largas delante de la pantalla, falta de sueño... pero ha merecido la pena, ha sido una experiencia satisfactoria para todos.

**5. Agradecimientos**

Tenemos que darle las gracias de todo corazón a nuestro consultor del proyecto y profesor, Eliseo Molero, nuestros profesores Ana Coello, Reyes Urbano, Noelia Cardoso, Lorena Silvia Díaz, Aurora Sánchez y Juan Antonio Ayuga por su empeño y dedicación durante estos dos años, y a todos los componentes de la Escuela Profesional Vedruna, por acogernos como a uno más en su gran familia.

A todos ellos, GRACIAS.