

VILLAPP



José Javier Escudero Gómez

Adrián Muñoz Gámez

Andrés Pascual Contreras

Índice

| | |
|---|----|
| Manual de instalación | 3 |
| Requisitos Mínimos | 3 |
| Manual del usuario | 4 |
| Resumen de la aplicación..... | 4 |
| Funcionalidades | 5 |
| Requisitos de Orientación..... | 8 |
| Diseño con modelos y comentarios | 12 |
| Casos de Uso | 12 |
| Diagramas de Clases | 17 |
| Diagramas de Secuencia..... | 22 |
| Aportaciones de los integrantes, metodología de trabajo y conclusiones..... | 29 |



Manual de instalación

Requisitos Mínimos

- Versión de Android mínima: Se requiere al menos la API 19 correspondiente a la 4.4 (KitKat).
- Permisos Requeridos:
 - Acceso a la Red
 - Ubicación precisa basada en GPS
- Como instalar la APK: Es necesario cumplir los requisitos mínimos y permitir la instalación de aplicaciones de origen desconocido (Acceder a Ajustes, Seguridad y marcar la casilla Orígenes desconocidos).
- Como configurar el proyecto en Android Studio: En primer lugar, hay que acceder al SDK Manager de Android Studio, dentro del mismo ir a la pestaña SDK Tools y activar el Google Play Services (para poder utilizar Google Maps). Después es necesario instalar los emuladores de Genymotion (seguir el tutorial del siguiente enlace: <http://aegis.pe/como-configurar-genymotion-con-android-studio>) y flashear el kernel de los emuladores creados de Genymotion según la versión de Android que se esté emulando con el archivo .zip que se puede obtener del siguiente enlace: <http://opengapps.org/> (que contiene todo lo necesario para que funcionen correctamente las aplicaciones de Google además de instalarlas).
- Se ha habilitado una cuenta llamada “profesor”, de contraseña “profesor”, con algunos lugares visitados para poder probar todas las funcionalidades.

Manual del usuario

Resumen de la aplicación

Villapp es una aplicación turística centrada en los lugares más importantes de la Villa Madrid durante un período histórico no estrictamente acotado que abarca desde el reinado de los Reyes Católicos, pasando por el Siglo de Oro, hasta finales del siglo XVIII aproximadamente. La elección de los lugares viene marcada por su presencia en el mapa de la antigua Villa de Madrid de Teixeira.

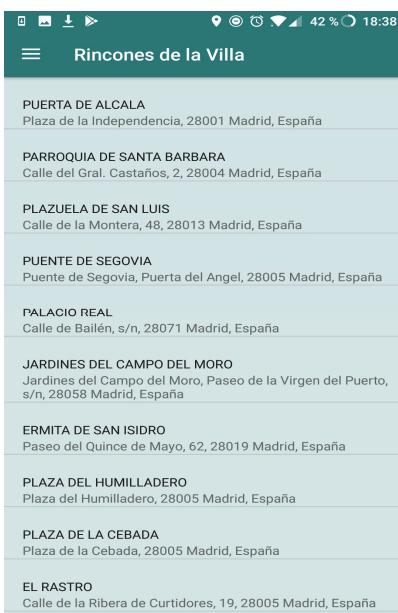
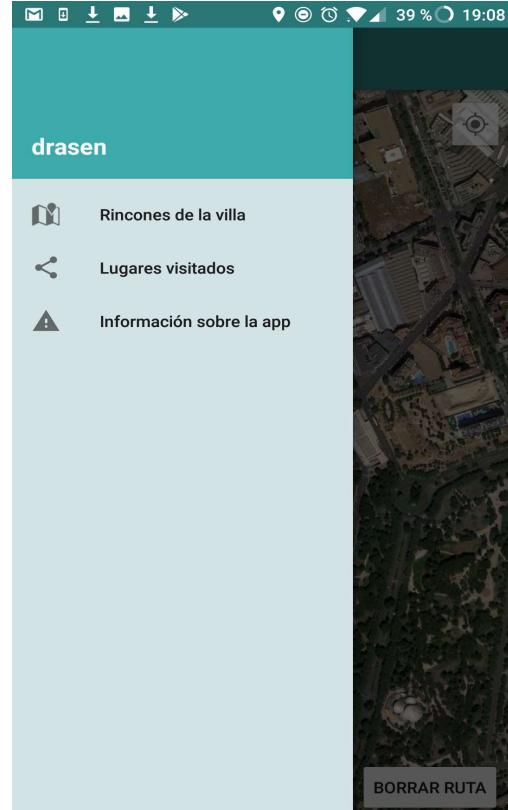


Topografía de la Villa de Madrid en el *Plano de Teixeira* completado en 1656 (Wikipedia)

El objetivo ha sido incluir tan solo aquellos emplazamientos históricos que se encontraban ya presentes en el Madrid de aquella época y que aún hoy en día se pueden visitar por encontrarse prácticamente intactos o bien por mantener alguna referencia de la época.

Funcionalidades

- **Panel lateral:** Desde el panel lateral de la aplicación podremos acceder al listado de sitios de la villa totales que se pueden visitar, a los sitios que el usuario ya ha visitado y a la información sobre la app (ver imágenes inferiores).



Lugares visitables

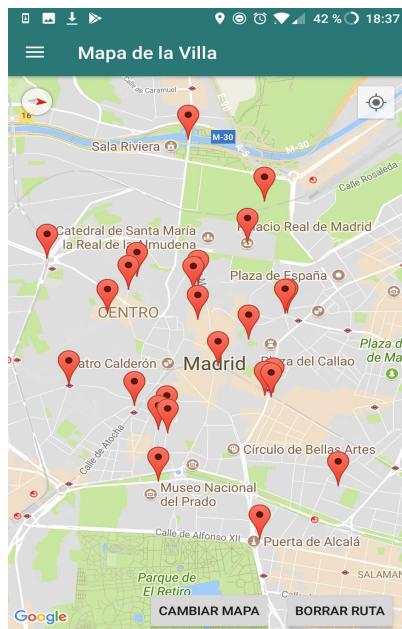
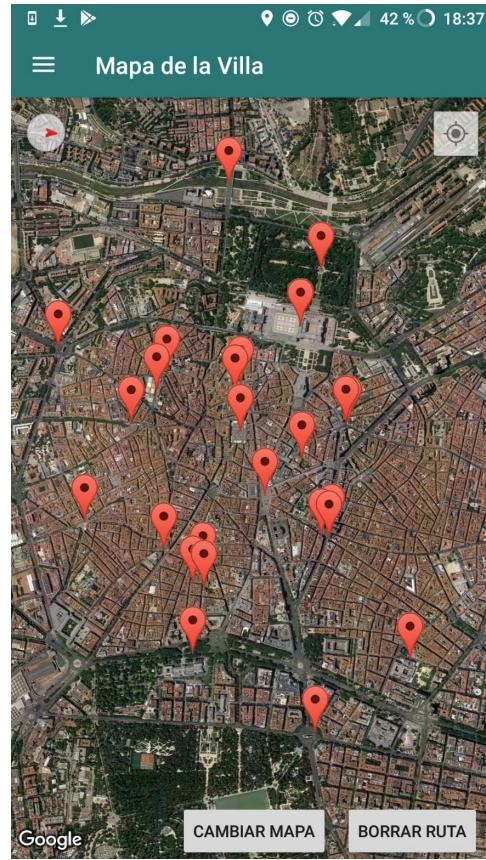


Lugares ya visitados

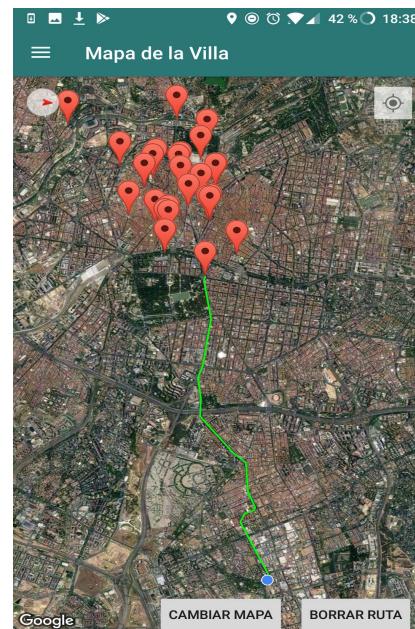


Acerca de

- Sistema de navegación:** Esta funcionalidad constituye el corazón de la aplicación. El sistema despliega un mapa en la pantalla del Madrid actual en modo satélite (por defecto) que el usuario puede cambiar por el modo callejero¹. Sobre este mapa se despliegan los puntos de interés que pueden ser visitados. Estos puntos son obtenidos desde una API propia, desarrollada ex profeso para albergar los lugares concretos de la aplicación. El usuario estará localizado en todo momento en el mapa mediante el sistema de GPS (de no estar activo este sistema se fijara la vista en uno de los puntos del mapa) y además podrá fijar una ruta desde un punto de visita a otro². Pinchando en cualquiera de estos puntos el usuario accederá a una descripción ilustrada del sitio centrada en la historia del lugar³.



(1) Callejero



(2) Establecer ruta



(3) Descripción

- **Sistema de valoraciones:** Para cada uno de los lugares que pueden ser visitados, el usuario podrá ver las opiniones vertidas sobre él por los demás usuarios así dar la suya propia en caso de que se encuentre próximo al lugar. En la imagen adyacente podemos observar los comentarios de los usuarios, mientras que en las imágenes inferiores podemos observar el proceso por el cual el usuario da su valoración al lugar.



Usuario dando su opinión

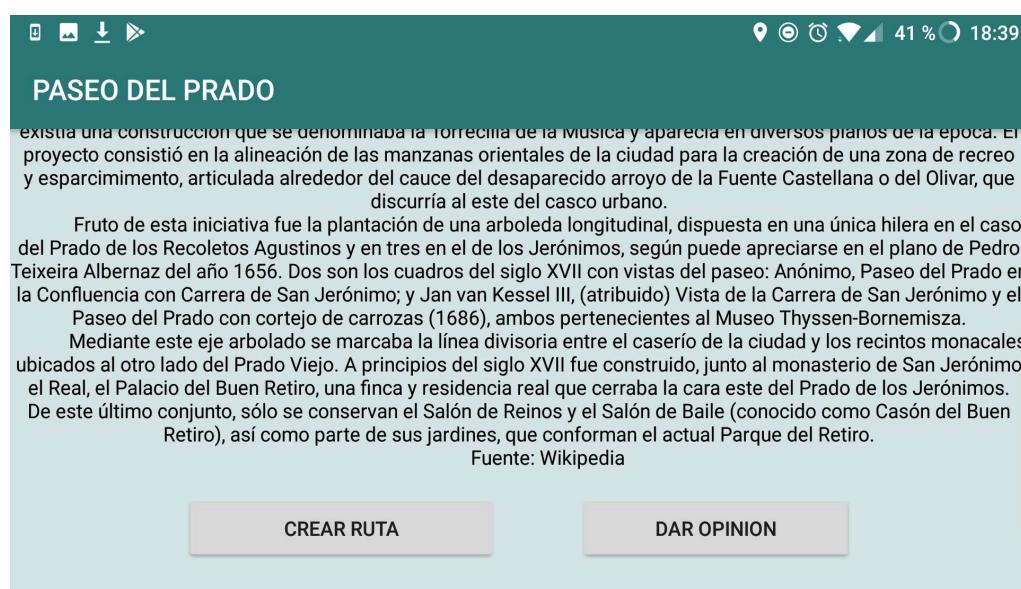


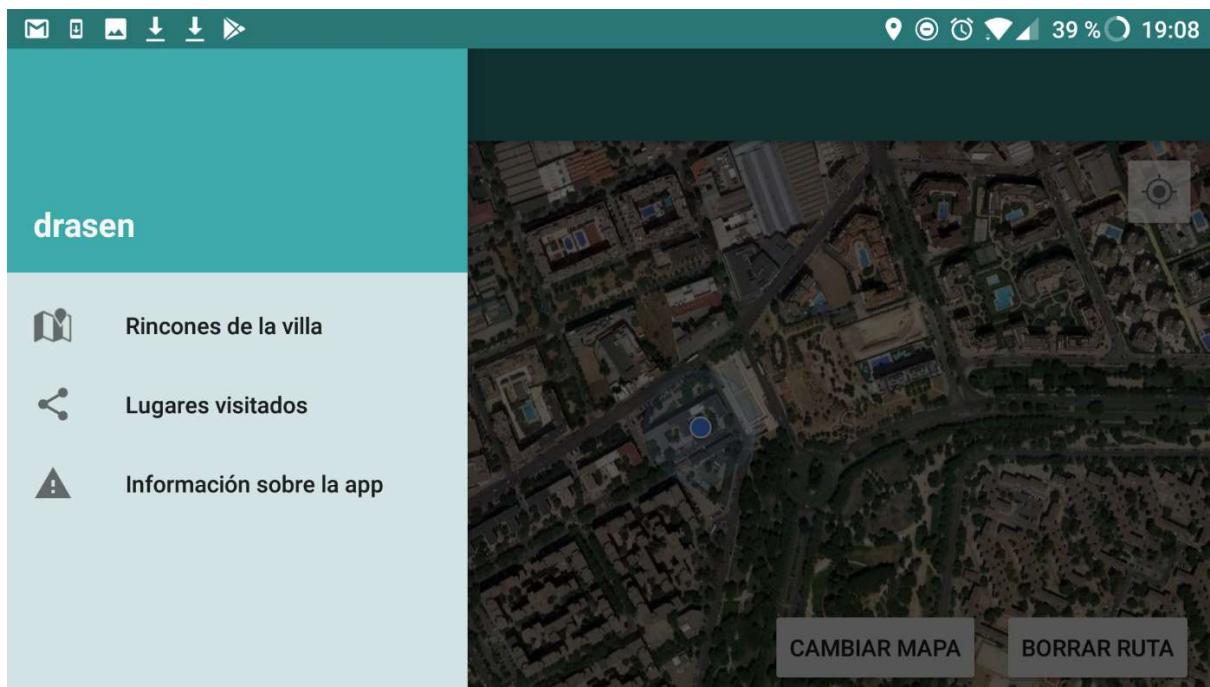
Opinión ya dada por usuario

(Las imágenes de opinión no se corresponden a la versión final, debido a que fueron tomadas en un dispositivo real y no estaba cerca de ningún punto de interés de la Gloriosa Villa de Madrid)

Requisitos de Orientación

En este apartado se mostrarán varias imágenes en las que se puede observar cómo se visualizan algunas de las pantallas expuestas en el apartado anterior cuando el dispositivo se gira horizontalmente.





The screenshot shows the "Acerca de" (About) page of the VillaApp. At the top, there is a dark header bar with icons and the time "18:39". Below it, the page has a teal header with the text "≡ Acerca de". The main content area features the title "VillaApp" in a large, bold, black font. Below the title, the text "Desarrolladores:" is followed by the names "José Javier Escudero Gómez - @efectoDoppler", "Andrés Pascual Contreras - @DrasenPC", and "Adrián Muñoz Gámez - @Nadrix". At the bottom of the page, there is a historical map of Madrid titled "LA VILLA DE MADRID CORTE DE LOS REYES CATÓlicos DE ESPAÑA".

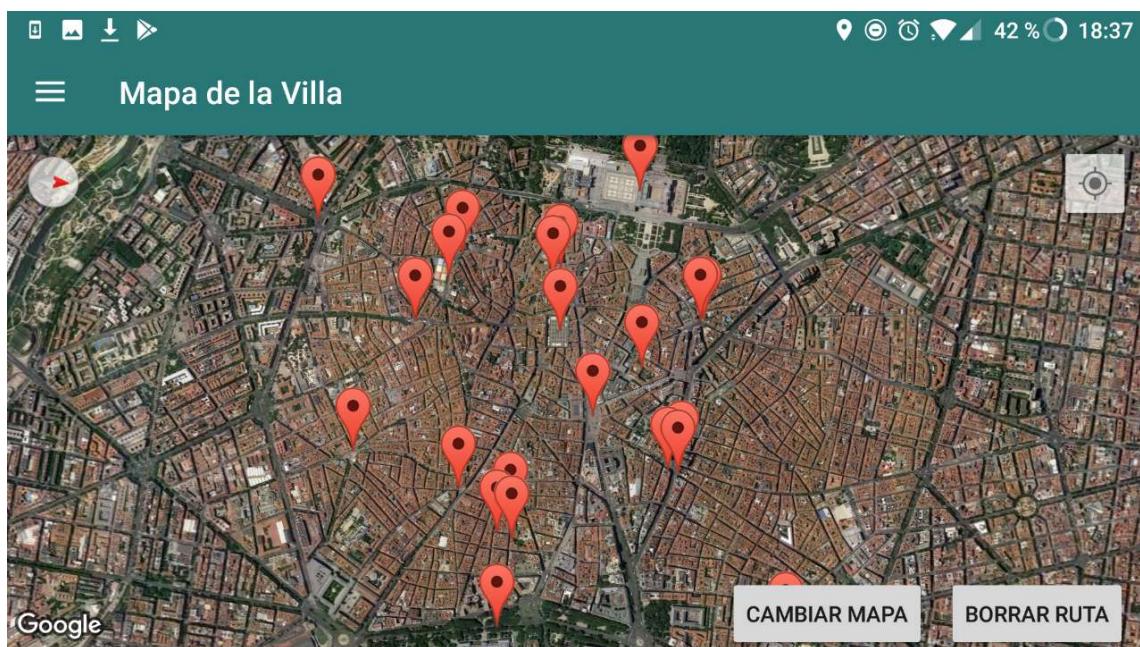
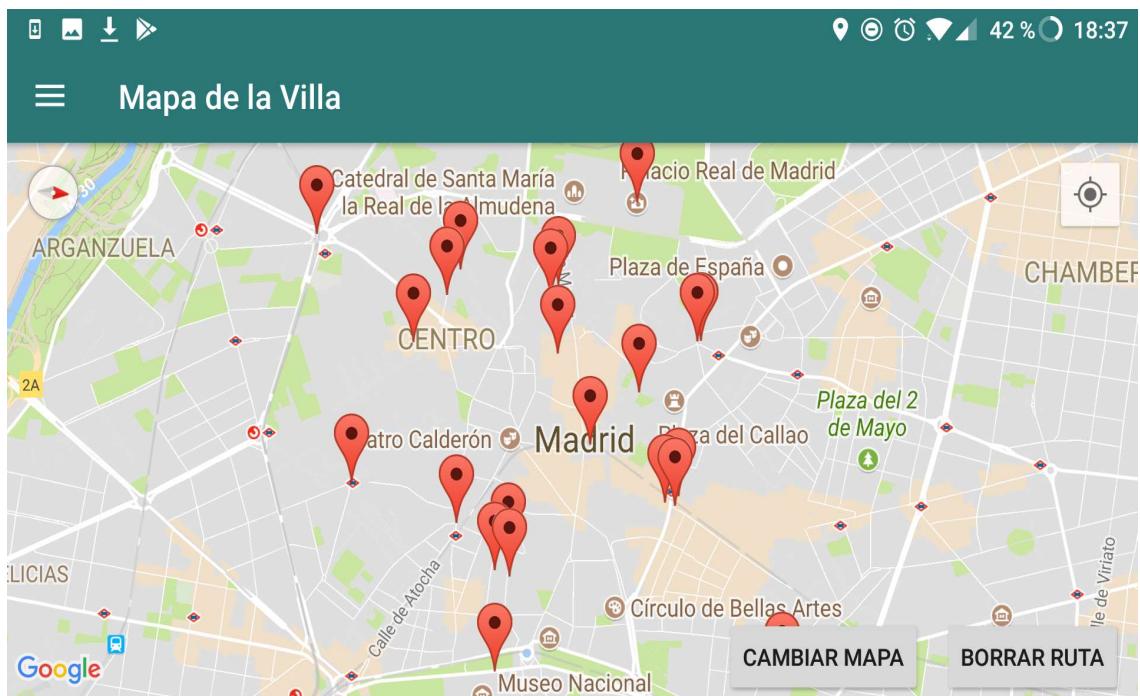
Desarrolladores:

José Javier Escudero Gómez - @efectoDoppler

Andrés Pascual Contreras - @DrasenPC

Adrián Muñoz Gámez - @Nadrix

LA VILLA DE MADRID CORTE DE LOS REYES CATÓlicos DE ESPAÑA





42% 18:38

≡ Rincones de la Villa

PUERTA DE ALCALA

Plaza de la Independencia, 28001 Madrid, España

PARROQUIA DE SANTA BARBARA

Calle del Gral. Castaños, 2, 28004 Madrid, España

PLAZUELA DE SAN LUIS

Calle de la Montera, 48, 28013 Madrid, España

PUENTE DE SEGOVIA

Puente de Segovia, Puerta del Angel, 28005 Madrid, España

PALACIO REAL

Calle de Bailén, s/n, 28071 Madrid, España

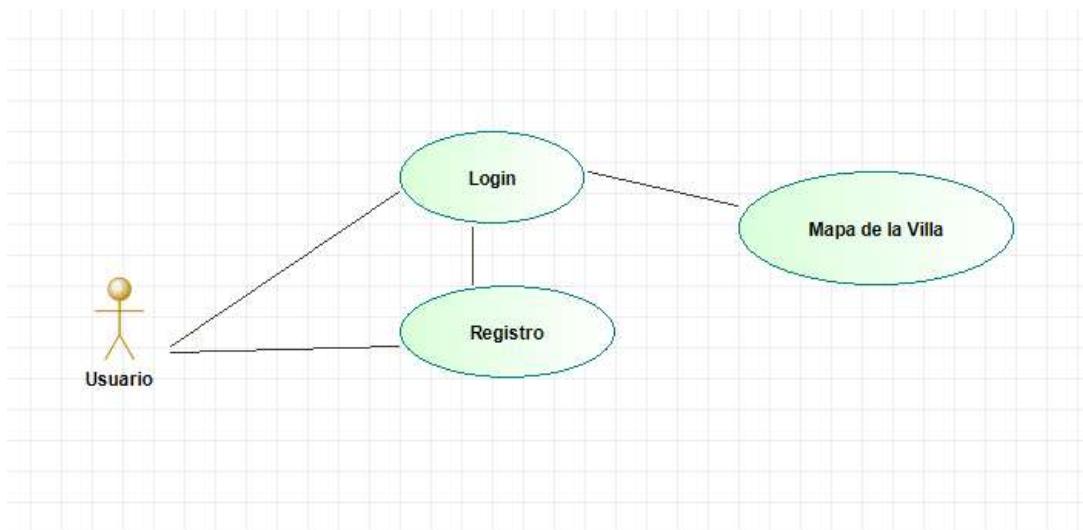


42% 18:38

≡ Rincones visitados

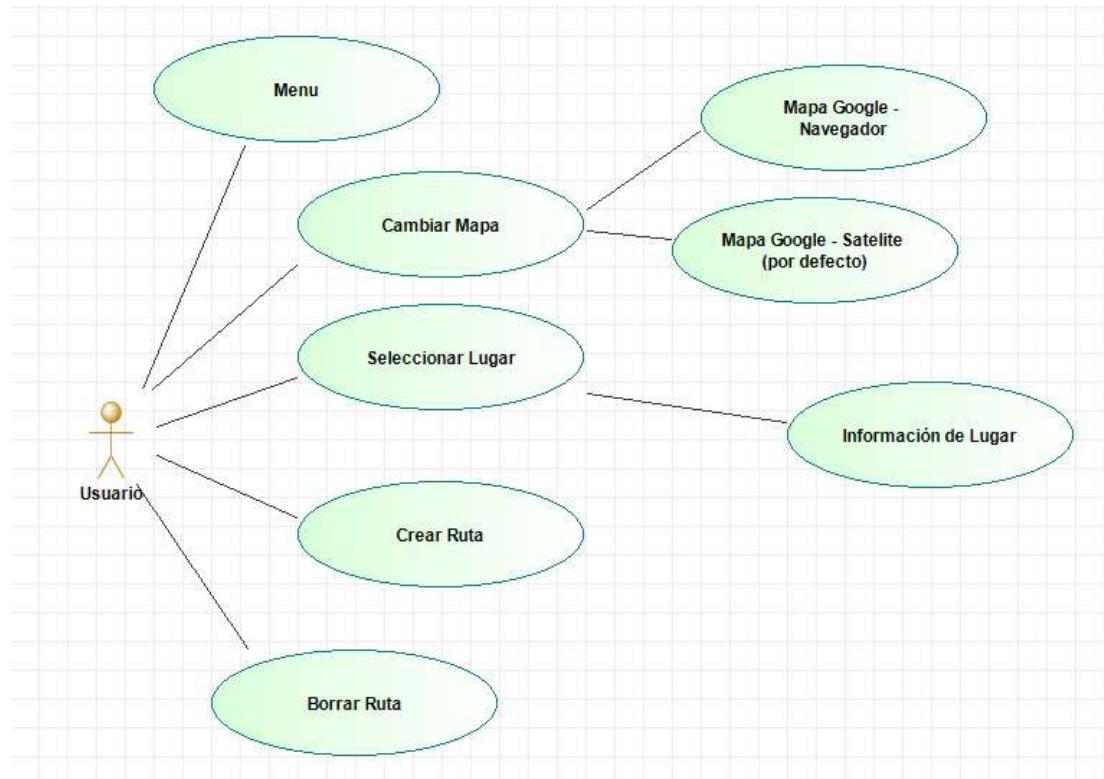
Diseño con modelos y comentarios

Casos de Uso



Casos de uso del Inicio de la aplicación

Al iniciar por primera vez aplicación (o si se han borrado los datos de la misma), el usuario llegará a la pantalla de Login desde donde podrá acceder a la app con sus credenciales, accediendo a la ventana principal (Mapa de la Villa). Si el usuario no está registrado, podrá acceder desde el Login a la pantalla de Registro, donde podrá registrarse en la aplicación para poder utilizarla. En posteriores ocasiones (salvo que se borren los datos de la app), el usuario entrará directamente a la ventana principal (Mapa de la Villa).

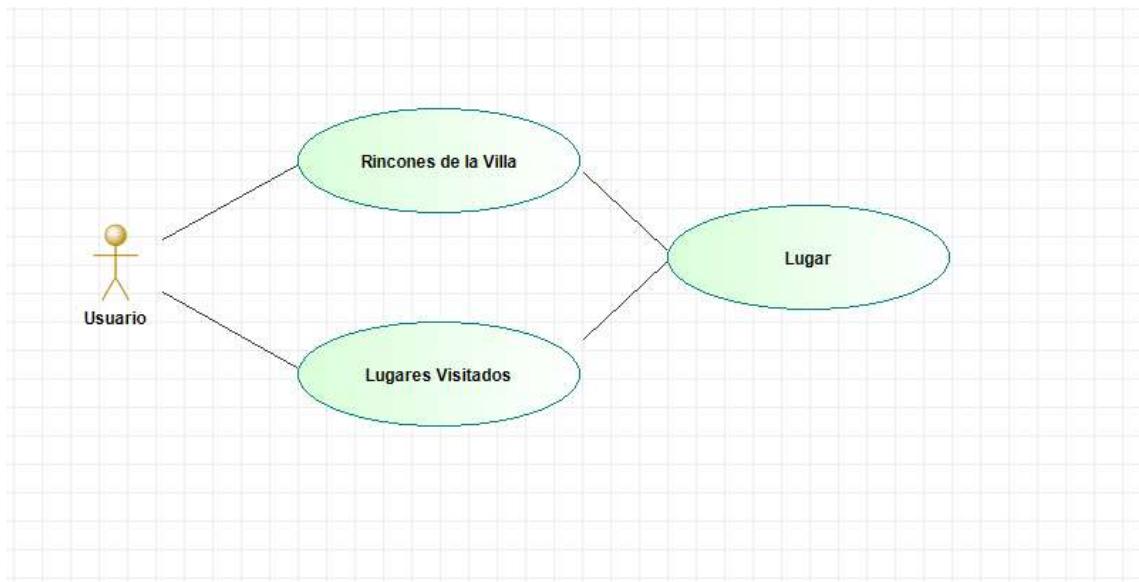


Casos de uso de la Ventana Principal (Mapa de la Villa)

En la ventana principal el usuario puede desplegar el Menú (Panel Lateral), cambiar de mapa entre la vista de satélite (con las fotografías de la ciudad tomadas desde arriba por satélite) y la vista de navegación de Google Maps.

También puede seleccionar cualquiera de los puntos de interés señalados en el mapa para acceder a la Información de dicho lugar.

Por último, el usuario puede crear una ruta entre dos puntos que le facilite la llegada al punto de interés de su elección o borrar una ruta creada para establecer otra diferente.



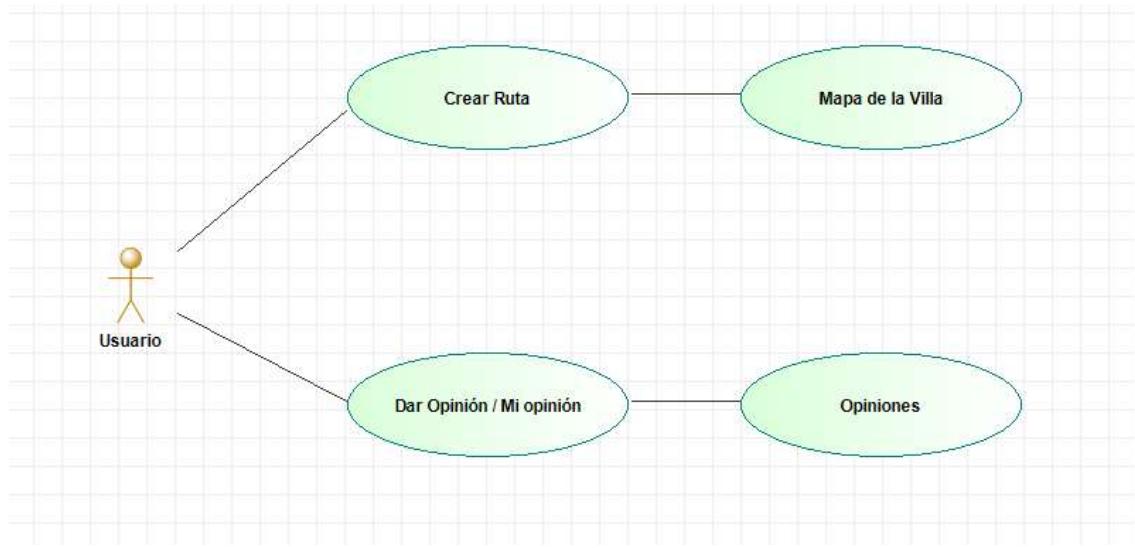
Casos de uso de Menú (Panel Lateral)

Mientras el usuario se encuentre en la ventana principal de la aplicación (Mapa de la Villa), podrá desplegar el menú, situado en la parte superior izquierda de la pantalla.

El menú se despliega en forma de panel lateral y ofrece las opciones de Rincón de la Villa y Lugares Visitados.

La primera opción, Rincones de la Villa, lleva al usuario a la lista completa de puntos de interés que ofrece la aplicación. Desde esta lista podremos acceder directamente a la información de un lugar concreto, de la misma forma que si lo hiciésemos pulsando sobre uno de los puntos en el Mapa de la Villa.

La segunda opción, Lugares Visitados, nos lleva también a una lista de lugares solo que, en este caso, aparecerán en la lista tan solo aquellos puntos de interés que ya hayamos visitado con anterioridad. Desde esta lista, al igual que desde la lista de Rincones de la Villa, el usuario podrá acceder directamente a la información del lugar pinchando sobre este de la misma forma que si lo hiciera en el Mapa de la Villa.

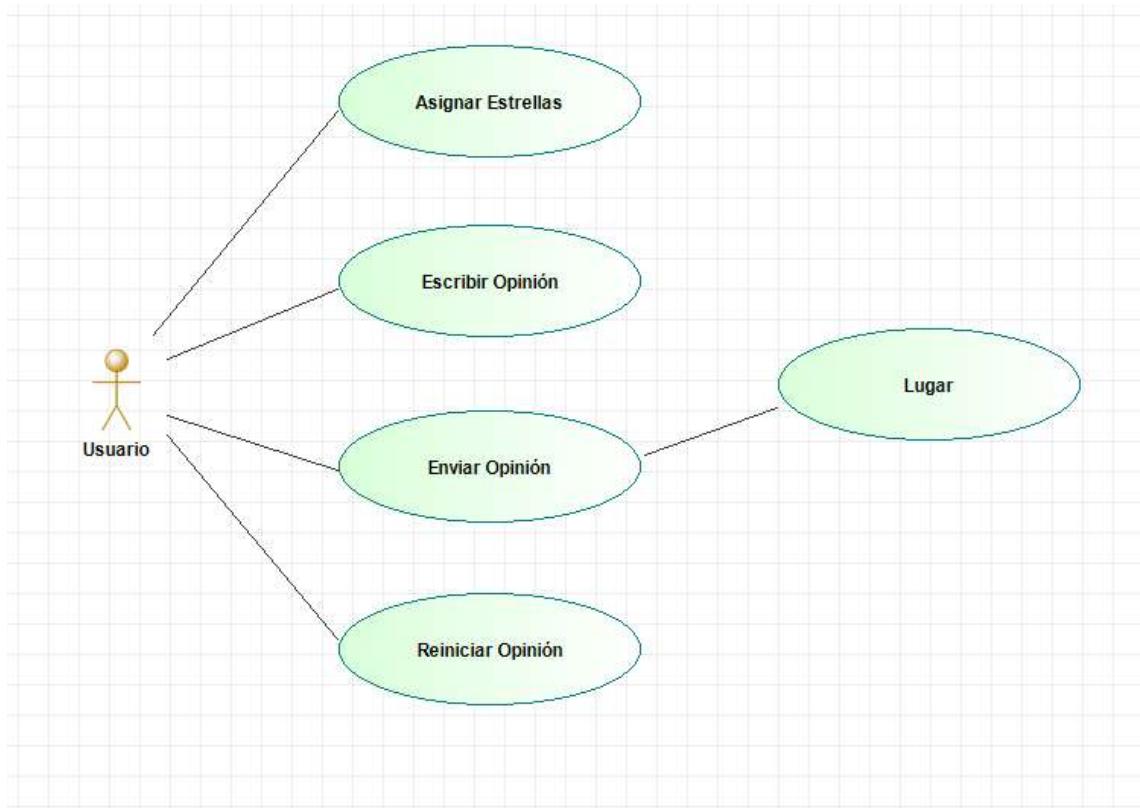


Casos de uso de Lugar

Desde la ventana de un Lugar concreto, el usuario podrá leer la información del mismo, así como observar una imagen, consultar su dirección y ver las opiniones arrojadas por otros usuarios acerca de dicho emplazamiento. Respecto a la funcionalidad, el usuario podrá crear una ruta ha dicho lugar, dar su opinión o consultarla en caso de que ya la haya dado.

El primer lugar, si el usuario selecciona Crear Ruta, se creará una ruta desde su ubicación actual hasta el lugar que estaba consultando. Funciona de la misma manera que si el usuario lo hiciera desde el botón Crear Ruta de la ventana principal (Mapa de la Villa).

Si el usuario selecciona Dar Opinión/Mi Opinión será direccionado a la ventana de opinión.



Casos de uso de Dar Opinión

En la ventana de Opinión, en el caso de que el usuario hubiera dado una opinión sobre el lugar con anterioridad, podrá observar con detalle la opinión que dio sobre dicho lugar en cuestión (número de estrellas que otorgó y opinión escrita vertida).

Si no ha dado una opinión sobre el lugar, en esta ventana el usuario podrá valorar el punto de interés con un sistema de valoración basado estrellas. Después el usuario podrá escribir una opinión sobre dicho lugar para concretar más a fondo qué le ha parecido el sitio, si lo recomienda a otros usuarios o cualquier opinión que decida verter sobre el mismo.

Por último, el usuario podrá enviar su opinión, quedando esta guardada y visible, a partir de ese momento, para el resto de usuarios o reiniciarla. Si opta por esta última opción, tanto el campo de estrellas como el de texto serán borrados para que pueda volver elaborar la opinión. Si decide enviar su opinión, será redireccionado de nuevo a la ventana del Lugar.

Diagramas de Clases

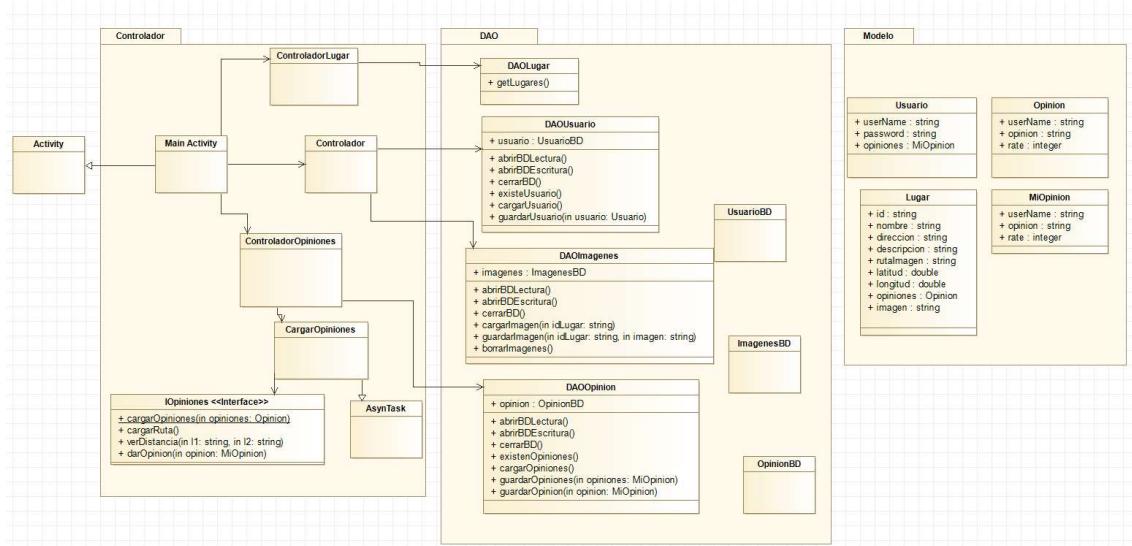


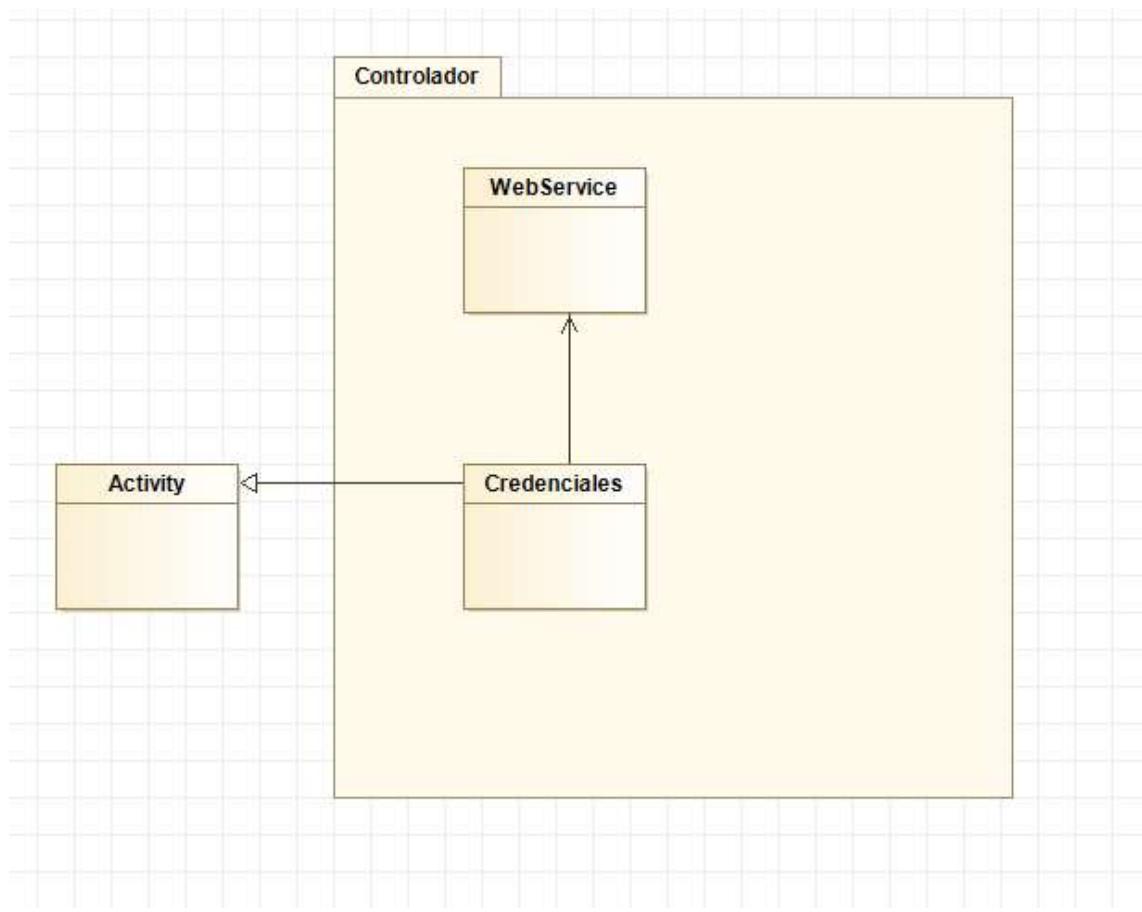
Diagrama Login Automático

En primer lugar, el Controlador de lugar carga todos los lugares disponibles en la API interna de la aplicación. Una vez cargados, el controlador comprobará si existe un usuario dentro del dispositivo llamando al DAO correspondiente (DAOUsuario) y este devolverá o bien un booleano falso (en caso de que no exista), o bien el usuario.

A continuación, se cargará la lista de lugares en el dispositivo llamando al DAO correspondiente (DAOLugar).

Después, y tan solo si existe el usuario, se llamará al ControladorOpiniones para cargar las opiniones vertidas por el usuario. Primero, se hará una comprobación para ver si están en el dispositivo (DAOOpinion). En caso contrario, llamará a la AsyncTask para cargarlo desde la base de datos externa.

Por último, si el usuario existe, la actividad saltará a la actividad principal mediante un Intent. De no existir, se saltará a la actividad Credenciales, donde el usuario podrá registrarse (en caso de que sea la primera vez que usa la aplicación) o loguearse (en caso de que ya esté registrado, pero sus datos no se encuentren guardados en el dispositivo). Una vez se ha realizado el salto se destruye la actividad para liberar espacio en la pila de actividades.



Esta actividad es muy simple debido a que únicamente posee un WebService que carga una web externa con un formulario de login y de registro. Una vez que se ha completado el formulario correspondiente (según el usuario deba registrarse o ya lo esté y tan solo deba loguear) los datos se envían a la parte nativa mediante javascript y se le envía a la actividad CargaDatosLogin. Una vez se ha realizado el salto se destruye la actividad para liberar espacio en la pila de actividades.

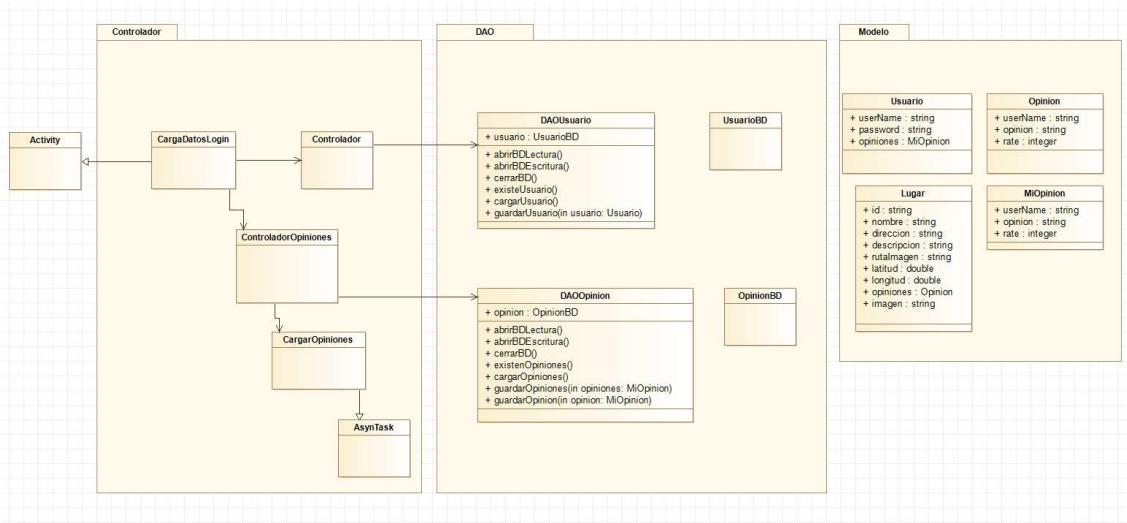


Diagrama CargaDatosLogin

Una vez se han recibido los datos obtenidos en la actividad credenciales, se llama al controlador para almacenar el usuario en la base de datos interna (DAOUsuario). Una vez guardado se cargan las opiniones del usuario de la base de datos externa mediante un AsyncTask y se almacenan en la base de datos interna (DAOOpiniones).

Por último se salta a la actividad principal mediante un Intent. Una vez se ha realizado el salto se destruye la actividad para liberar espacio en la pila de actividades.

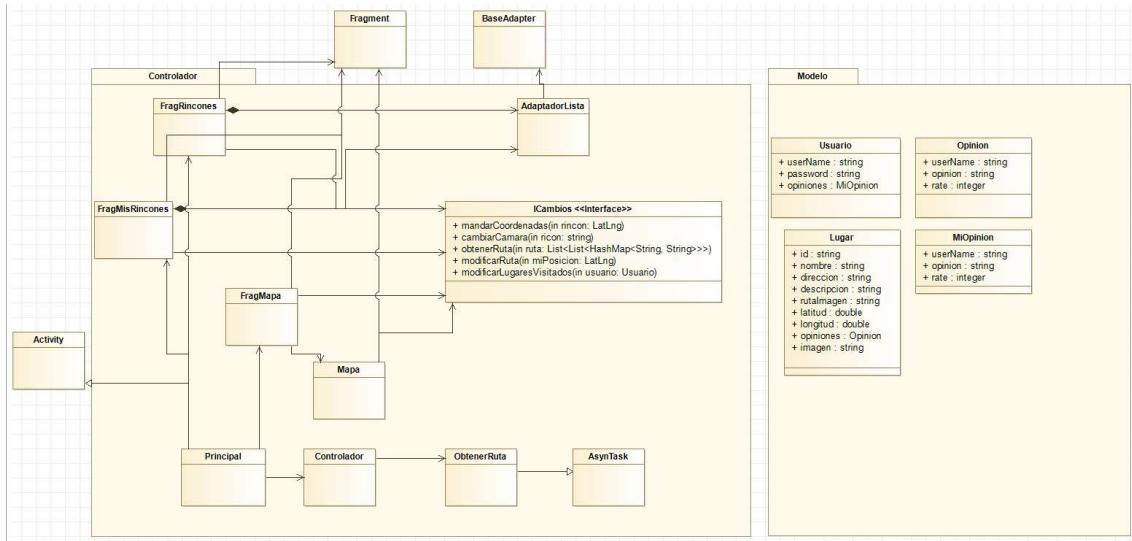


Diagrama Actividad Principal

Esta actividad constituye el núcleo básico de la funcionalidad de la aplicación, desde donde se accede a todas las otras actividades.

Una vez obtenidos los datos enviados desde CargaDatosLogin la actividad carga el fragmento del mapa (FragMapa), el cual constituye la vista principal de la aplicación. Desde dicho fragmento se puede seleccionar un lugar marcado en el mapa y saltar a la actividad LugarClicado (para obtener más información sobre el mismo).

Si abrimos el Menú (Panel Lateral) podremos acceder a la lista de lugares disponibles (FragRincones) o bien a la lista de lugares que ya hemos visitado (FragMisRincones). Ambos fragmentos se componen por un adaptador (AdaptadorLista), que se encarga de mostrar los lugares de manera visualmente adecuada.

Todos los fragmentos poseen una interfaz (ICambios) la cual sirve para comunicar los cambios realizados (crear una ruta entre la posición actual y un lugar, cambiar el tipo de mapa o recibir una opinión desde la actividad LugarClicado).

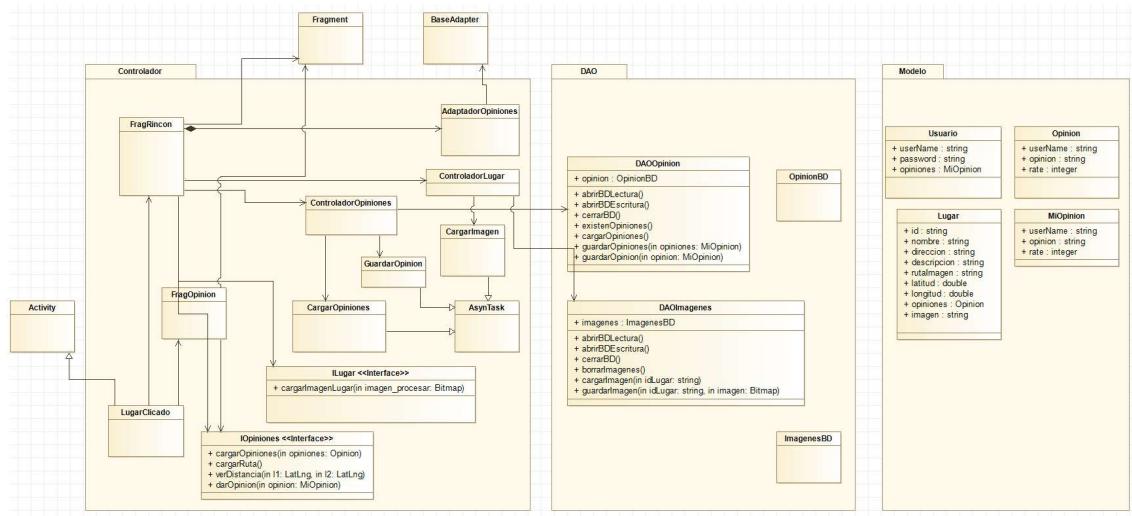


Diagrama de LugarClicado

Una vez obtenidos los datos mandados desde la Actividad Principal se carga el fragmento FragRincon desde donde se muestra toda la información turística del lugar seleccionado previamente (AsyncTask CargarImagen o de DAOImagenes, si está en la cache del dispositivo) y las opiniones de otros usuarios, si las hubiera (cagarOpinion). Desde este fragmento también se puede solicitar crear una ruta en el mapa.

Si no hemos estado previamente en ese lugar podremos dar una opinión, y si hemos estado ya podremos consultar nuestra opinión. Todo esto se hace en FragOpinion.

Si no hemos dado una opinión, desde FragOpinion se llamará al ControladorOpinion y este guardará la opinión en la base de datos interna (DAOOpinion) y en la base de datos externa (AsyncTask GuardarOpinion).

Ambos fragmentos tienen dos interfaces (IOpciones e ILugar) que actúan como observadores para notificar cambios a la actividad LugarClicado.

Por último, cuando se abandona esta actividad, se elimina de la pila de actividades de Android.

Diagramas de Secuencia

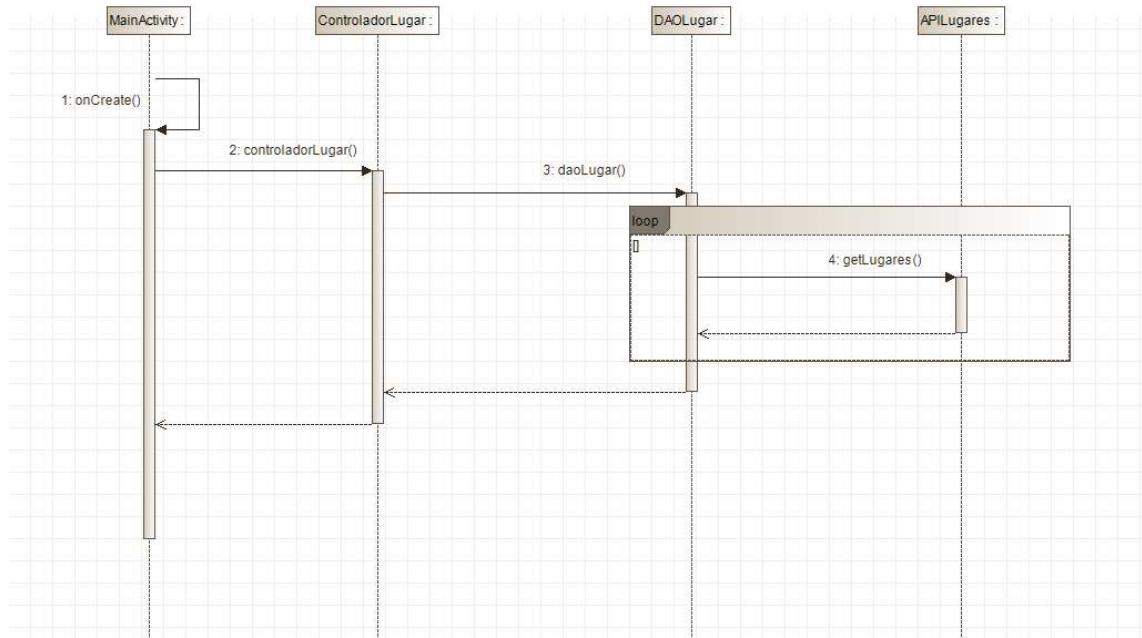


Diagrama de Secuencia Cargar Lugares

Una vez creada la `MainActivity` (1) se inicia el `ControladorLugar` (2).

Su constructor por defecto se encarga de crear a `DAOLugar` (3) y, una vez creado, este accede a la API interna donde almacenamos todos los lugares y los obtiene (4).

Cuando hemos obtenido todos los lugares, estos van retornando hasta volver a la actividad.

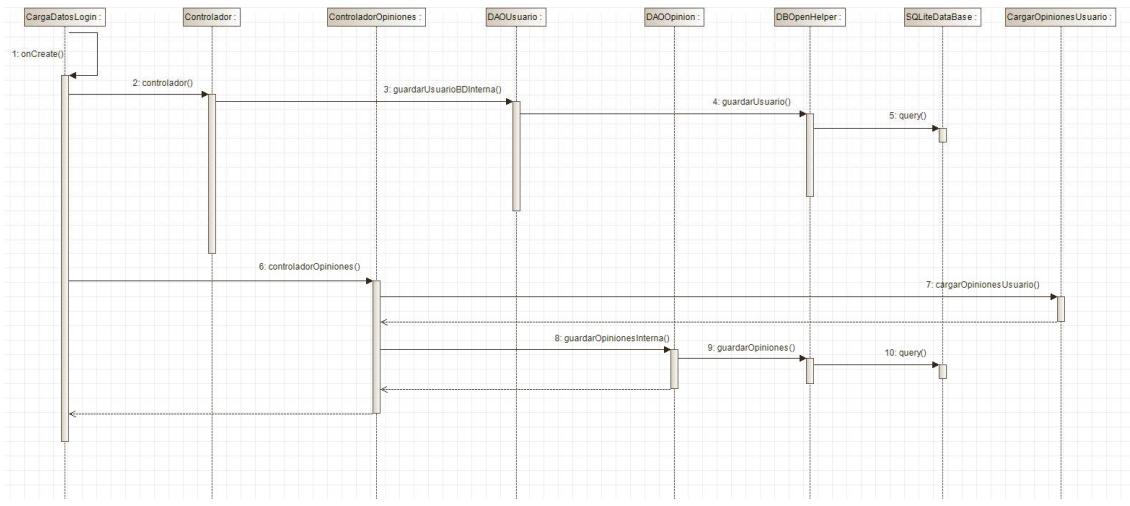


Diagrama de Secuencia Carga Datos Login

Una vez cargada la aplicación con los datos obtenidos de la actividad anterior (1), se llama a `controlador` (2) y se guardan las credenciales del usuario en la base de datos interna del usuario mediante `DAOUsuario` (3).

A continuación, se llama a `controladorOpiniones` (6) y, mediante un servicio web, se cargan las opiniones del usuario desde la base de datos externa (7).

Después de la recepción de los datos por parte del servicio web, estos se almacenan en la memoria interna del dispositivo mediante `DAOOpiniones` (8). Cuando todo este proceso ha finalizado, se mandan de vuelta a la actividad todos los datos obtenidos y se envían a la siguiente actividad.

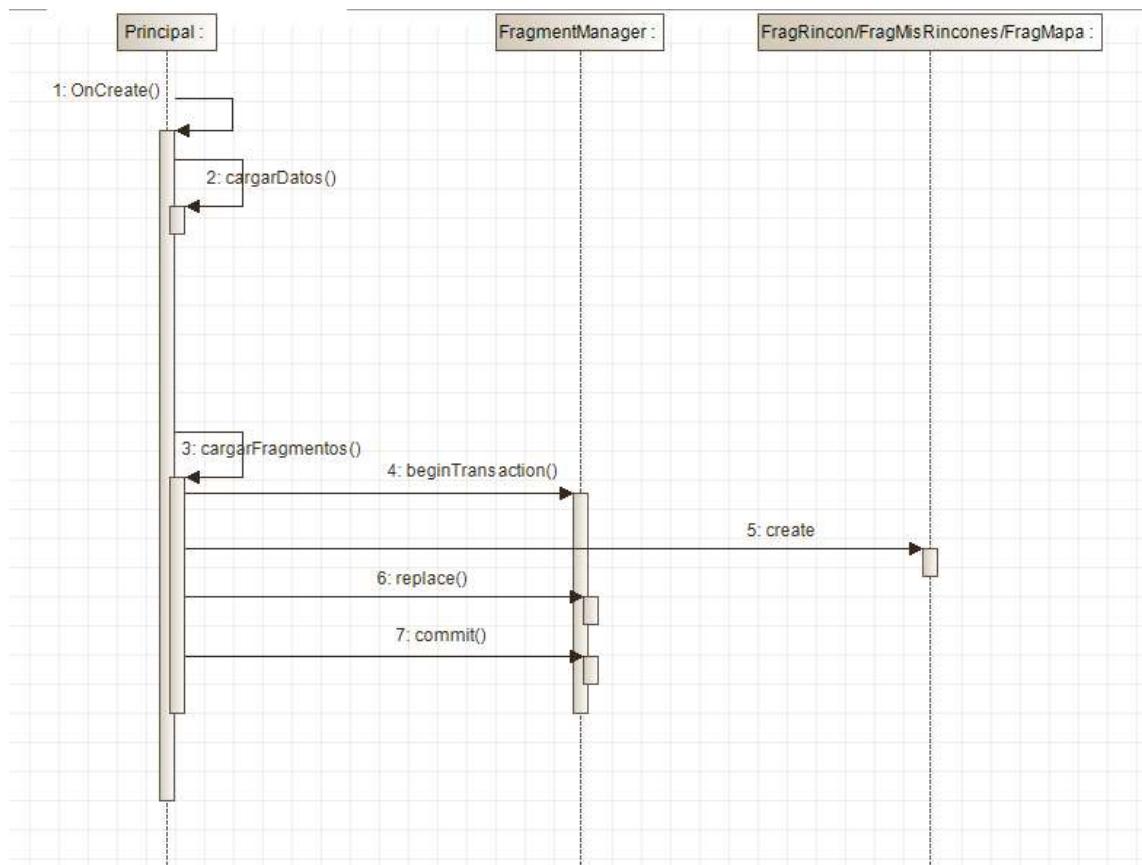


Diagrama de Secuencia Fragmentos en Actividad Principal

La actividad principal, una vez ha sido creada (1) y ha cargado los datos obtenidos mediante un bundle (2), procede a cargar los fragmentos (3). Para ello, se inicializa un FragmentManager (4) y se le añaden los fragmentos necesarios (5 y 6).

Por último, se confirma la transacción del FragmentManager (7).

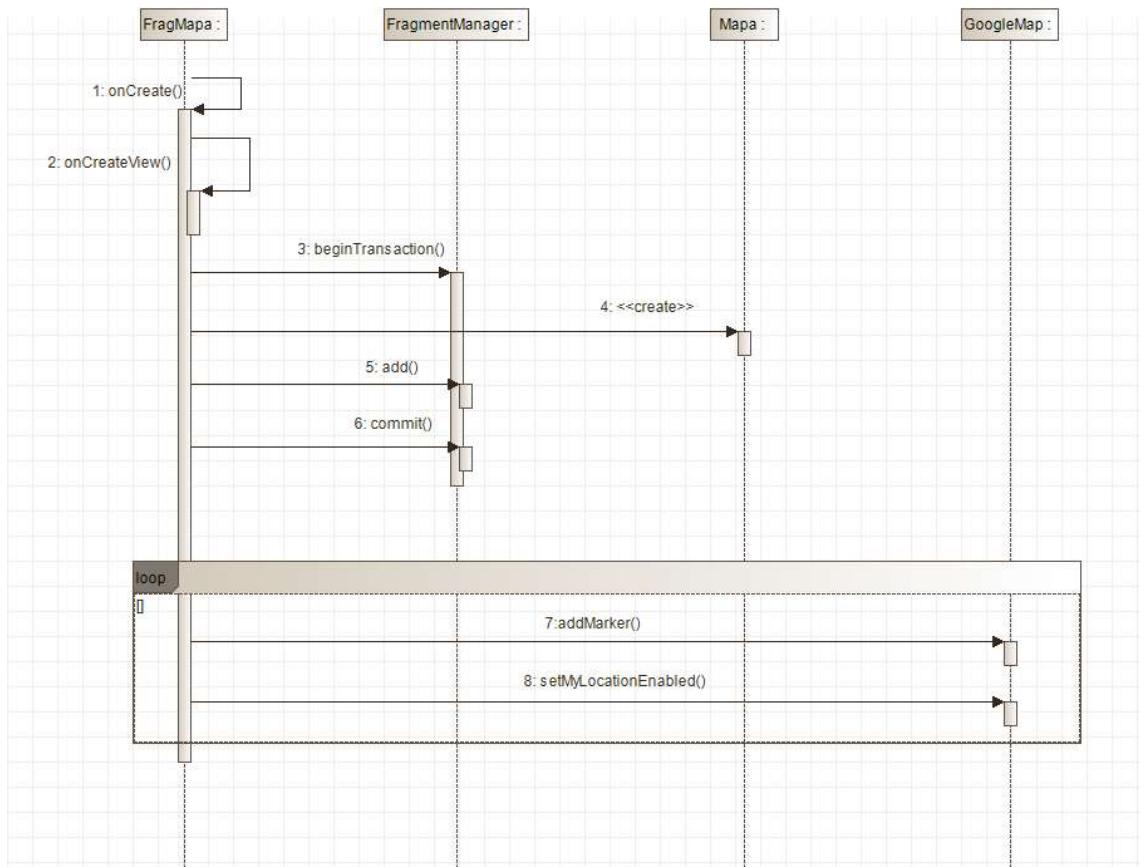


Diagrama Cargar Lugares en el Mapa

Una vez creado el FragMapa (1 y 2) se le añade el fragmento Mapa (4) mediante el FragmentManager. Para ello, se inicia una transacción (3) donde se incluye el Mapa (5) y por último se confirma (6).

Cuando ya se ha integrado el Mapa se colocan en él todos los lugares (7) y se fija la propiedad de que se muestre la ubicación del usuario (8).

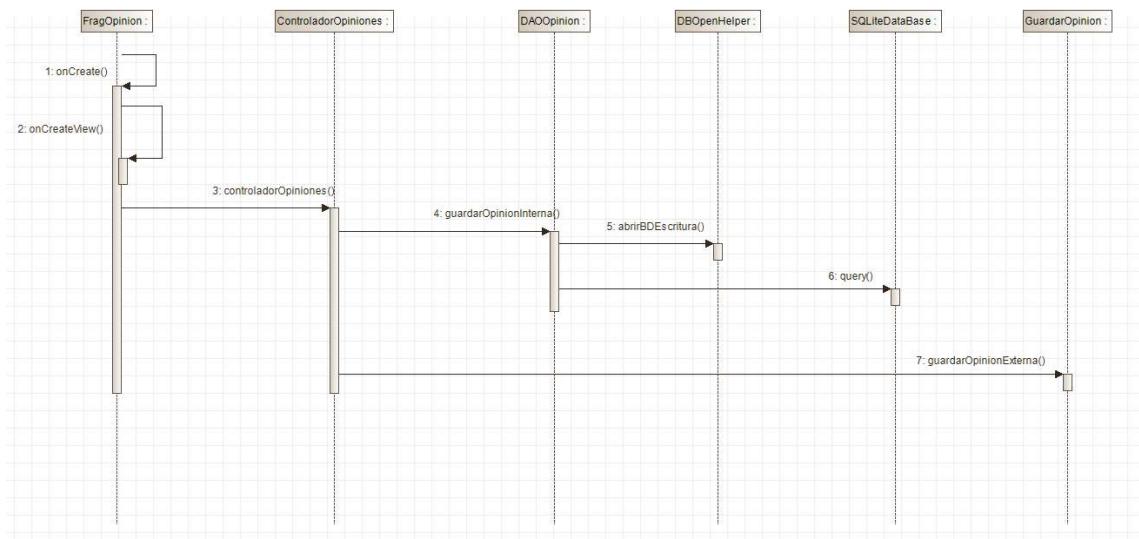


Diagrama de Secuencia de Dar Opinión

Una vez creado el fragmento de opinión (1, y 2) se procede a dar una opinión sobre el lugar. Cuando la opinión ha sido enviada por el usuario, se llama a `ControladorOpiniones` (3) y este se encargará de almacenarlo tanto en la base de datos interna, a través del `DAOOpinion` (4), como en el servidor alojado en la red (7).

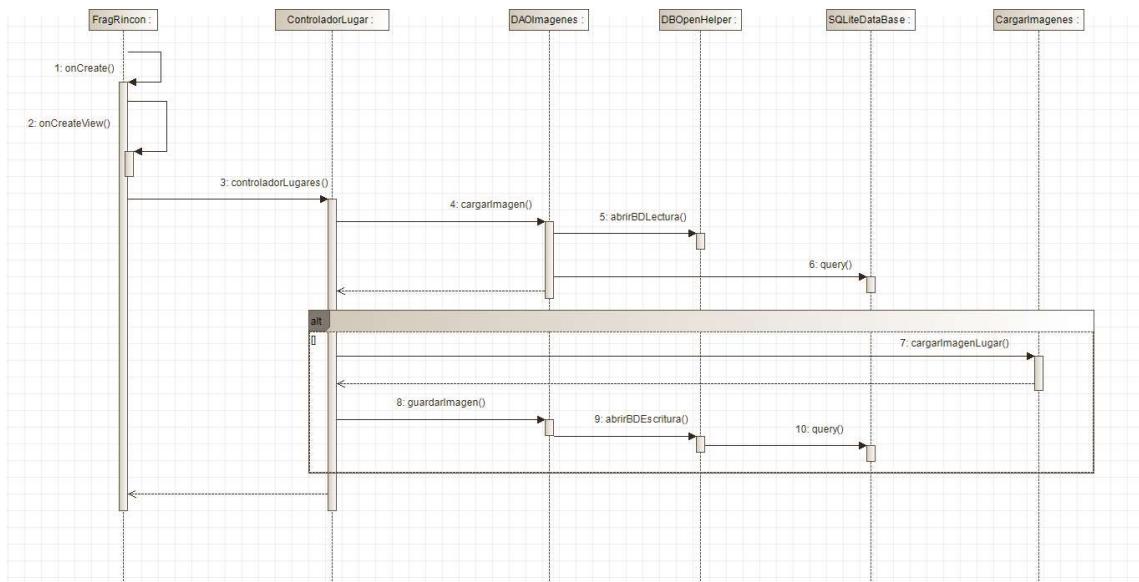


Diagrama de Secuencia Cargar Imágenes

Una vez creado el fragmento FragRincon (1 y 2), este crea el ControladorLugar (3) y trata de obtener la imagen del lugar de interés donde se encuentra el usuario.

Para ello, en primer lugar, buscará en la base de datos interna (que se usa de caché de imágenes) y tratará de cargarla (4 y 5). En el caso de que no la encuentre, la cargará del servidor web (7) y la almacenará en la base de datos interna (8).

Cuando ya se ha cargado la imagen, esta se retorna al fragmento y se mostrará en pantalla.

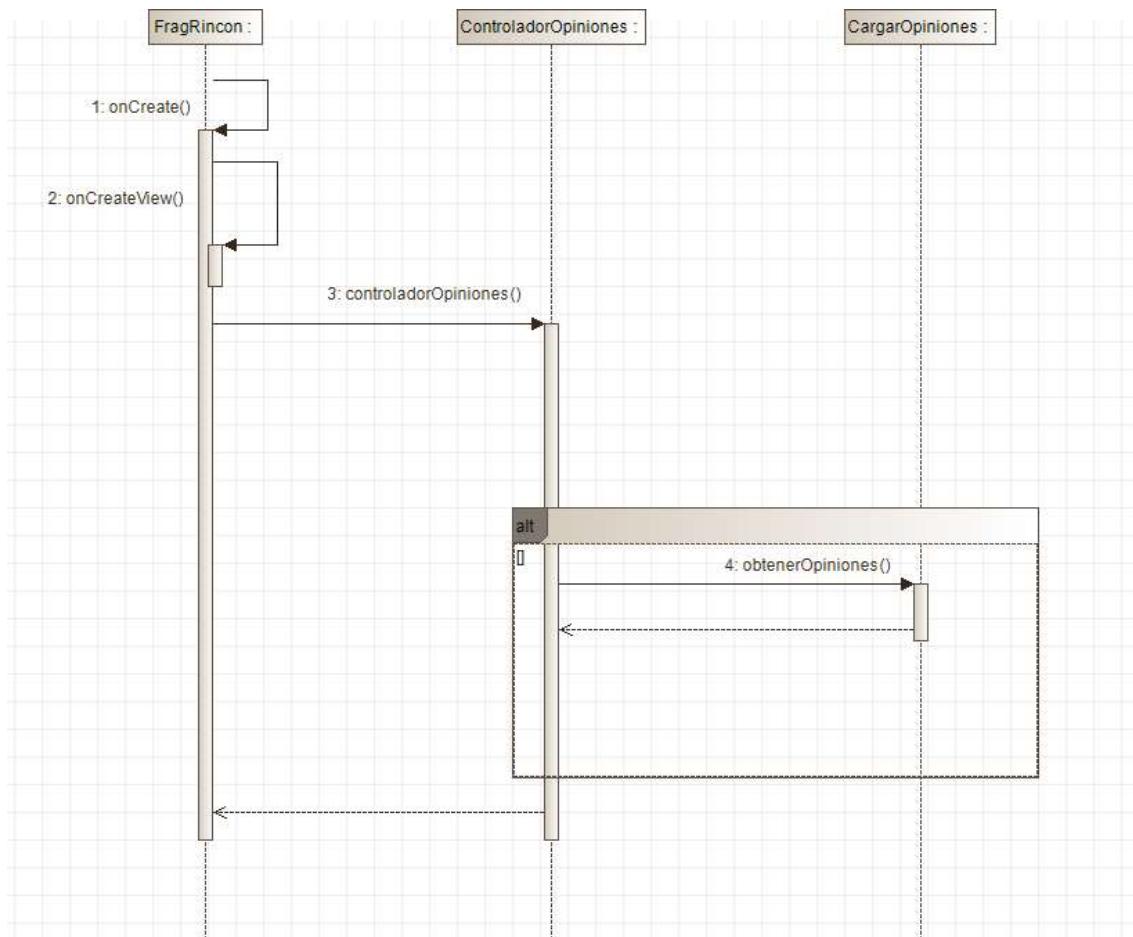


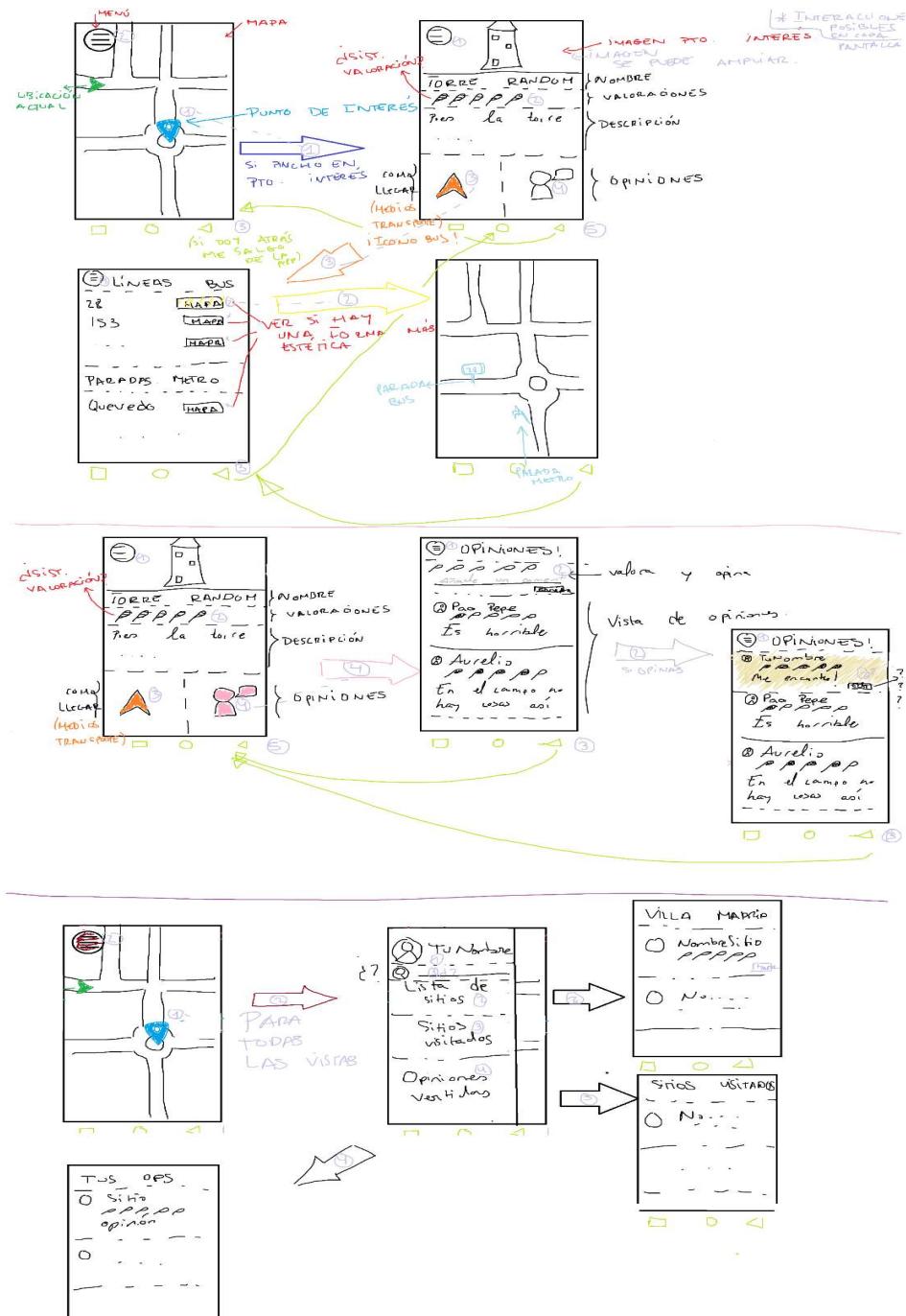
Diagrama de Secuencia de Cargar Opiniones

Una vez que se ha cargado el fragmento con éxito (1 y 2), se crea el controlador de opinión (3). En este punto, tan solo si el lugar no tiene cargada ninguna opinión, llamará al WebService y tratará de cargar los datos alojados en el servidor (4).

Al contrario que en el diagrama Cargar Imágenes los datos obtenidos no se cachearán en el interior del dispositivo y, por tanto, se perderán al cerrar la aplicación.

Aportaciones de los integrantes, metodología de trabajo y conclusiones

Durante las dos primeras semanas de proyecto se diseñó todo el servicio externo (base de datos, formularios de login y registro) así como se comenzó a maquetar cómo serían las diferentes vistas y el flujo básico de datos de la aplicación.



A partir de ese momento, se trabajó entre todos los integrantes para obtener un núcleo funcional de la aplicación, teniendo como objetivo la presentación preliminar.

Tras acabar el núcleo se procedió a terminar funcionalidades y a pulir el diseño y otros pequeños detalles de la app.

Respecto a la metodología de trabajo, se ha utilizado Android Studio para el desarrollo Nativo, Atom para el desarrollo web, GitHub para el control de versiones, PostMan (depuración PHP) y FileZilla (FTP) para la gestión en el servidor. En el aspecto de documentación, a parte de las herramientas de ofimática habituales, se ha utilizado Modelio (UML) para la creación de los distintos diagramas que aparecen en el presente documento (diagramas de uso, de clases y de secuencia).

Se trabajó paralelamente utilizando Skype y su funcionalidad para compartir pantalla como principal método de comunicación entre los miembros del equipo de desarrollo.

Por último, se han obtenido las siguientes conclusiones:

- La información respecto a diversos aspectos de Android está muy dispersa.
- El rendimiento de los emuladores de Android, comparado con un dispositivo real, deja mucho que desear.
- La documentación entregada en clase es insuficiente y debe complementarse continuamente con la documentación oficial online de Android.
- Tras completar el proyecto es posible afirmar que el equipo sea capaz de enfrentarse a proyectos más grandes en Android gracias a la experiencia adquirida.