

NEARBY

DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

PREPARADO POR:

JOSÉ MIGUEL MORA RODRÍGUEZ

KARINA ZELEDÓN PINELL

DYLAN RODRÍGUEZ BARBOZA

ABRIL, 2018

Contenido

Propósito	4
Requerimientos	5
Funcionalidades	5
UX, UI (wireframes)	7
Descripción de diseño de alto nivel	23
Diagrama de arquitectura	23
Diagrama de base de datos	24
Descripción de los web services	26
Listar usuarios	26
Crear usuario	26
Consultar un usuario	26
Editar un usuario	26
Eliminar un usuario	26
Listar categorías	27
Crear categoría	27
Consultar una categoría	27
Editar una categoría	27
Eliminar una categoría	27
Listar subcategorías	28
Crear subcategoría	28
Consultar una subcategoría	28
Editar una subcategoría	28
Eliminar una subcategoría	28
Listar comentarios	29
Crear comentario	29
Consultar un comentario	29
Editar un comentario	29
Eliminar un comentario	29
Listar eventos	29
Crear evento	30
Consultar un evento	30
Editar un evento	30
Eliminar un evento	30
Listar grupos	30
Crear grupo	31
Consultar un grupo	31
Editar un grupo	31
Eliminar un grupo	31

Listar usuarios asociados a evento	32
Crear usuario asociado a evento	32
Listar usuarios asociados a grupo	32
Crear usuario asociado a grupo	32
Listar categorías asociadas a un usuario	32
Crear categoría asociada a un usuario	33
Listar subcategorías asociadas a un usuario	33
Crear subcategoría asociada a un usuario	33
Asistir a un evento	33
Estar interesado en un evento	33
Obtener eventos cercanos	34
Obtener eventos cercanos por categoría	34
Obtener eventos para un usuario	34
Obtener un usuario de django	34
Obtener usuario actual	34
Obtener grupos por categoría	35
Obtener eventos por categoría	35
Interacción con sistemas externos	36
Mixpanel	36
Fabric	36
Facebook	37
Google Places	37
Google Maps	37
Cloudinary	37

Propósito

Nearby es una aplicación desarrollada para dispositivos android. Consiste en un medio para la creación de eventos, en donde se pueda especificar el lugar en el cual se llevará a cabo el evento. Consta de quince ventanas, mediante las cuales se pueden hacer acciones como: ver eventos, ver grupos que poseen eventos, ver el perfil del usuario, registrarse en la aplicación, visualizar los grupos y eventos de un usuario. En la aplicación, se hace uso de: el GPS y acceso a la memoria interna del dispositivo. Se hace conexión con el API de Facebook, Google Places y Google Maps. La aplicación tiene conexión con un API propio, desarrollado en django rest framework, en el cual se encuentra la lógica y el consumo de los datos de la aplicación.

La aplicación, inicialmente tuvo un alcance alto, indebidamente validado, que consistía de una gran cantidad de funcionalidades, por lo que se decidió mostrar la primera versión, sin la implementación de todas las funcionalidades planteadas en el prototipo, pero con el propósito de ampliar las funcionalidades en versiones posteriores, incluso para otras plataformas diferentes a android.

Requerimientos

Se requería que la aplicación cumpliera con los siguientes requisitos:

- Desarrollo en Android nativo (Java o Kotlin).
- Desarrollo de backend propio (no Baas), que sea accesible por dominio público.
- El backend debe tener un backoffice web de administración, con su propia autenticación.
- Debe usar user login.
- Deben manejar autenticación y autorización a nivel de API.
- Conexión con al menos dos APIs externos (sin contar el backend propio).
- Usar una de las siguientes características del teléfono: GPS, acelerómetro, cámara, compás.
- El app debe tener al menos 15 activities.
- Deben manejar métricas de uso con Mixpanel o Fabric.

Los requerimientos específicos que debía cumplir la aplicación, de acuerdo al alcance definido, son los siguientes:

- Hacer un inicio de sesión propio.
- Hacer un inicio de sesión con Facebook.
- Registrar un usuario en la aplicación.
- Seleccionar las categorías del usuario.
- Seleccionar las subcategorías del usuario.
- Visualizar eventos.
- Visualizar eventos más cercanos.
- Filtrar eventos por categoría.
- Crear un nuevo evento.
- Visualizar un evento.
- Comentar un evento.
- Dar la opción de asistir o estar interesado en un evento.

- Listar grupos de eventos.
- Visualizar un grupo de eventos.
- Filtrar eventos por asistencia o interés.
- Visualizar notificaciones de eventos creados para un grupo al que pertenezca un usuario.
- Visualizar el perfil del usuario.
- Listar eventos del usuario.
- Listar grupos del usuario.

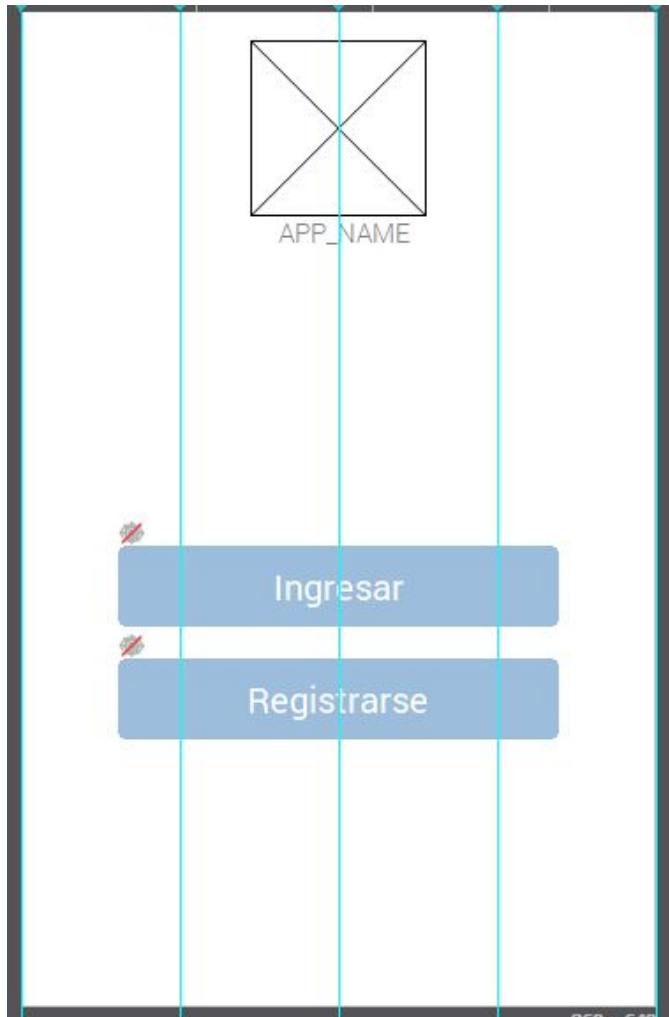
Funcionalidades

.Para esta versión de la aplicación, se tomaron en cuenta las siguientes funcionalidades:

- Registrarse en la aplicación.
- Inicio de sesión propio.
- Autenticación por tokens.
- Seleccionar las categorías de un usuario.
- Seleccionar las subcategorías de un usuario.
- Listar eventos.
- Listar eventos, dada una categoría.
- Listar eventos más cercanos.
- Hacer un inicio de sesión con Facebook.
- Crear un evento, en el cual se incluya una imagen que lo ilustre y la ubicación del evento.
- Visualizar un evento, en el cual se puedan ver sus comentarios respectivos e información general.
- Comentar un evento.
- Visualizar un grupo de eventos, en el cual se pueda ver su información general.
- Listar grupos de eventos.
- Poder votar como usuario, si se va a asistir o estar interesado en un evento.
- Filtrar los eventos por el voto que hizo un usuario en el evento, es decir, que se puede separar la visualización de los eventos a los que asistirá de los eventos en los que está interesado.
- Visualizar notificaciones, estas serán los nuevos eventos que se agreguen a un grupo al que pertenezca un usuario.
- Visualizar el perfil del usuario.
- Cambiar foto y nombre del perfil.
- Visualizar los eventos de un usuario en su perfil.
- Visualizar los grupos de un usuario en su perfil.

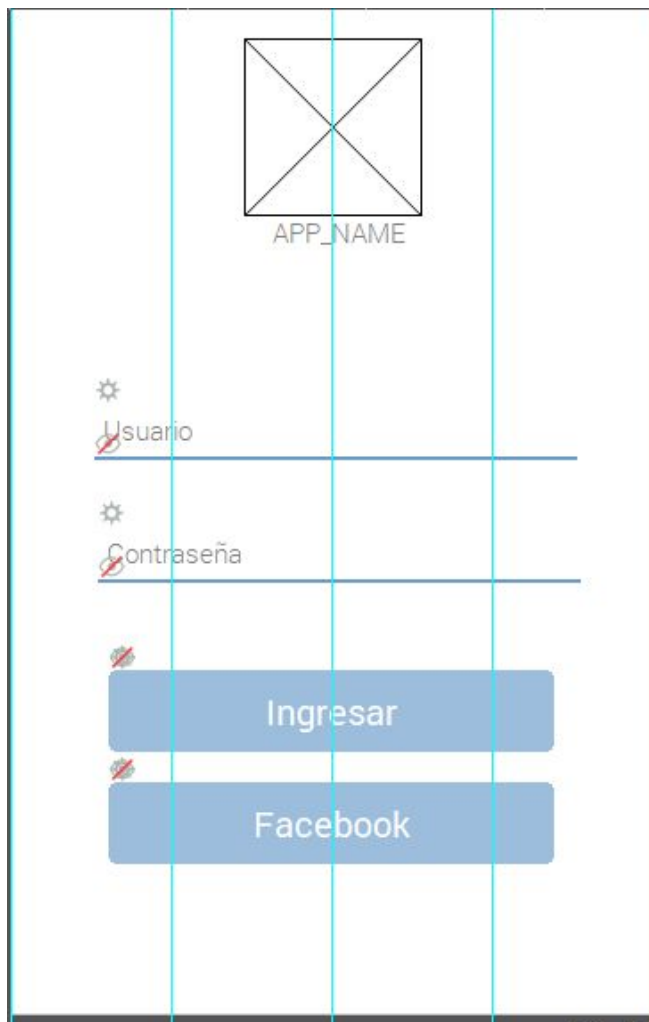
UX, UI (wireframes)

A continuación, se muestran los wireframes que se propusieron originalmente para la aplicación:






















Principal

En esta primer activity, se muestra la pantalla de inicio de la aplicación, la cual está delimitada por el nombre de la aplicación (parte superior de la pantalla) y dos botones: uno encargado de ingresar, que dirigirá a otro activity para digitar los datos del usuario, y otro encargado de registrar, en el cual saldrá un pop up para inscribirse con facebook o manualmente.



Inicio de sesión

En esta segunda activity, se permite ingresar los datos, para ingresar a la aplicación, puede ingresar con credenciales de la aplicación o bien, con la cuenta de facebook registrada.

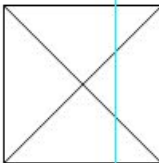
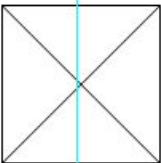
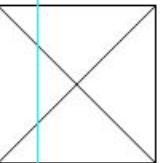
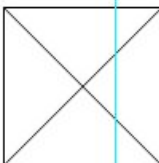
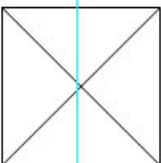
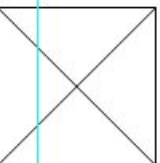
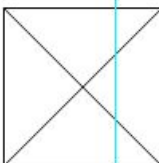
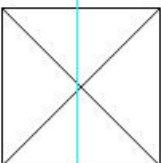
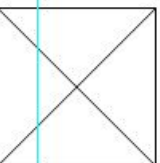
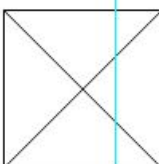
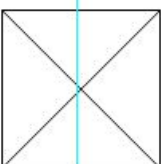
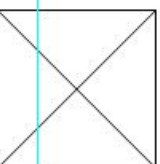
			
	Correo		
			
	Contraseña		
			
	Confirmar Contraseña		
	Fecha Nacimiento		
	 Día	  Mes	  Año
			
	 Femenino		
	<div>Registrarse</div>		


Registro

En esta actividad, el usuario será capaz de registrarse en el sistema, deberá ingresar datos como el correo electrónico, contraseña, fecha de nacimiento y género.

			<div>Saltar</div>
--	--	--	-------------------


Seleccione las categorías que le interesan

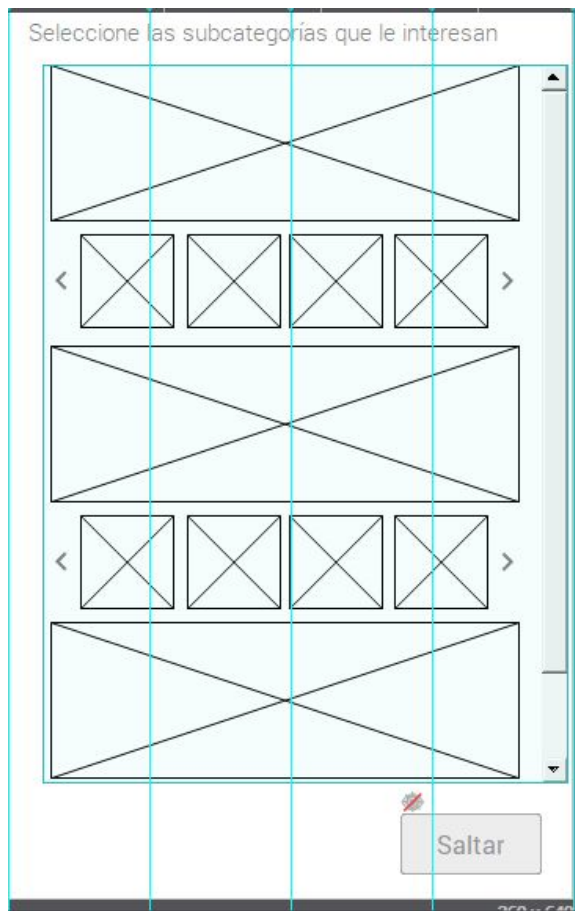
		
		
		
		

 Continuar

Categorías de interés

En esta actividad, el usuario podrá seleccionar los temas de interés, de manera que la aplicación los pueda tomar en cuenta para presentarle eventos en su perfil. Se permite que el usuario evite esta sección de la aplicación, mediante el botón de saltar.

 Saltar



Subcategorías de interés

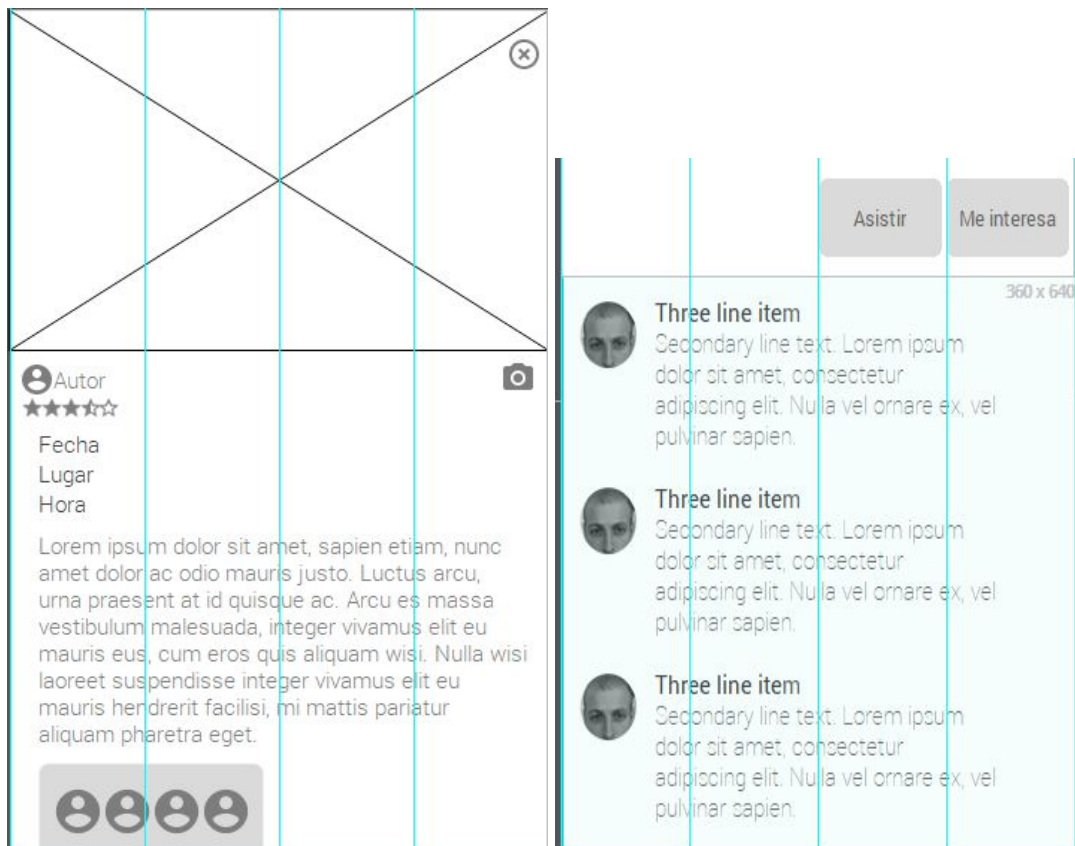
En esta actividad, el usuario podrá seleccionar las subcategorías de los temas de interés, mostrados anteriormente, de manera que la aplicación los pueda tomar en cuenta para presentarle eventos en su perfil. Se permite que el usuario evite esta sección de la aplicación, mediante el botón de saltar. Se puede deslizar hacia la derecha o izquierda en los cuadros de subcategorías (más pequeños), para mostrar más contenido.





Menú principal

En este activity se muestra un menú de navegación en la parte inferior, en el resto del activity se muestra un buscador de eventos, un filtro de eventos (por día, hora y distancia), una sección para ver eventos cerca del usuario (debajo del filtro) y categorías que ha gustado a la derecha de este. Debajo de estos, aparecen eventos, en el cuál se presentarán sus respectivos datos, el símbolo de la cámara representará la subcategoría a la que pertenece. Finalmente, se pueden agregar nuevos eventos (ya sea individual o grupalmente) mediante el botón con el símbolo “+”.



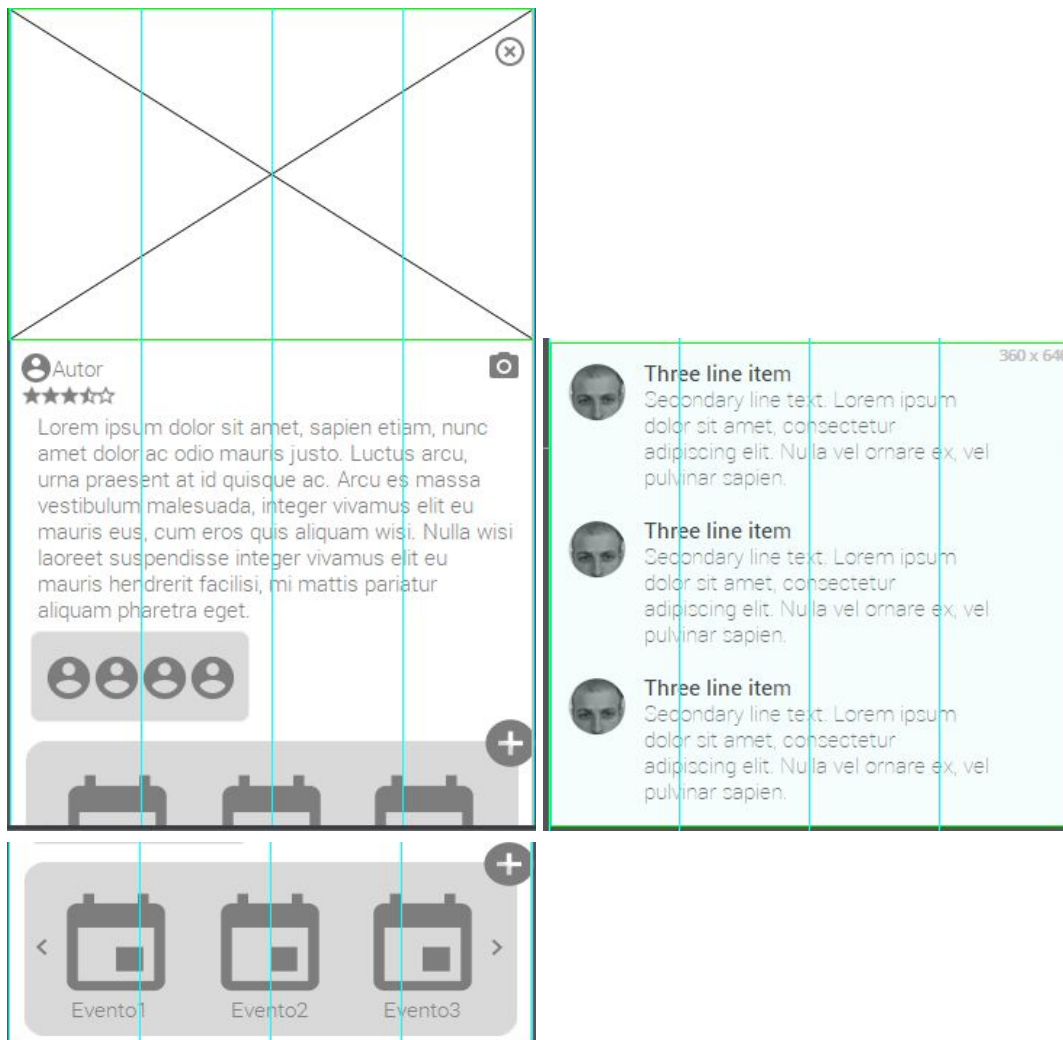
Detalle de evento

En este activity se muestra el detalle de un evento, con datos como el autor, fecha, lugar (uso de google maps), descripción, una imagen, calificación, comentarios. Y una sección que muestra los participantes (cuadro con perfiles) el cual mostrará un pop up con los participantes del evento. Además, se presenta la oportunidad de asistir o mostrarse interesado por el evento. El símbolo de “x” dentro de la imagen, permite volver al menú principal.



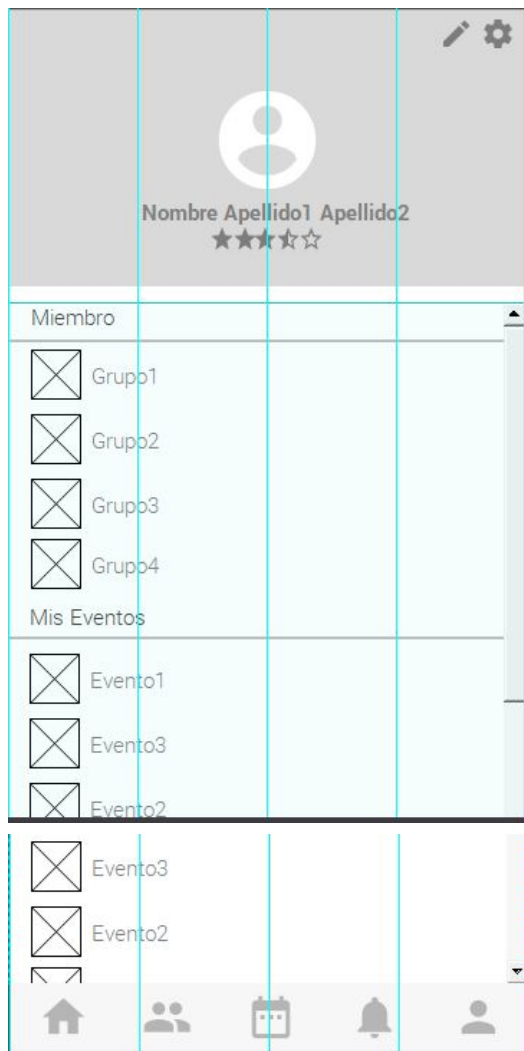
Grupos

En este activity se muestra los grupos, al inicio aparecen los grupos a los que se pertenece y el buscador funciona como el activity de menú principal, solo que este buscará grupos. Este activity se activa cuando se da click al segundo botón del menú de desplazamiento en la parte inferior. El símbolo de “+” permite agregar un nuevo grupo.



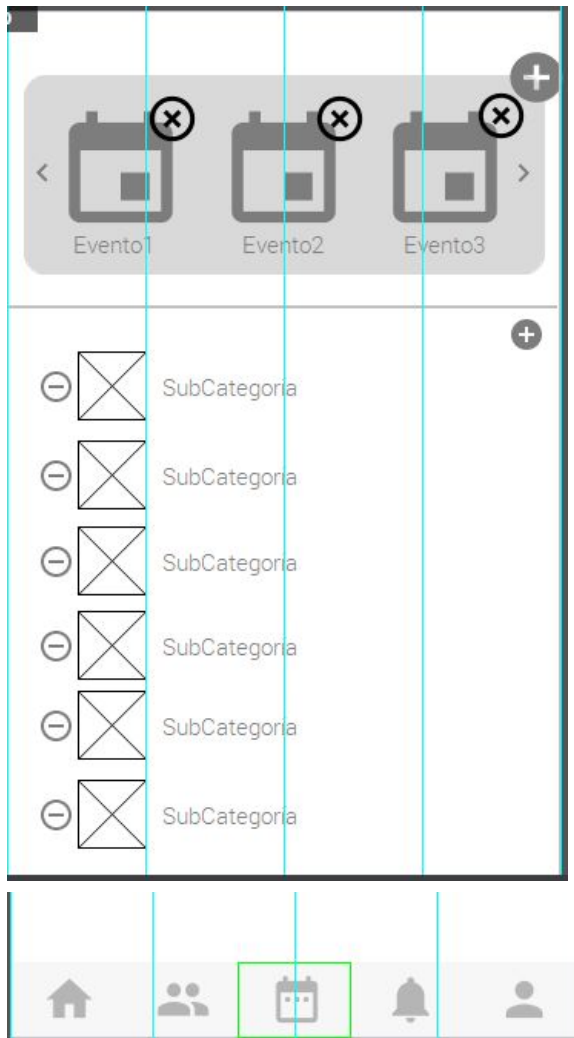
Detalle del grupo

En este activity se muestra la imagen del grupo, el autor del grupo, calificación del autor, descripción del grupo, lista de miembros, subcategoría a la que pertenece y eventos que posee el grupo, con la posibilidad de añadir más, mediante el símbolo de “+”, el autor debe aprobarlos para que sean publicados. Además, se muestran comentarios para dicho grupo. El símbolo de “x” en la imagen de la parte superior, hace que se devuelva al menú principal.



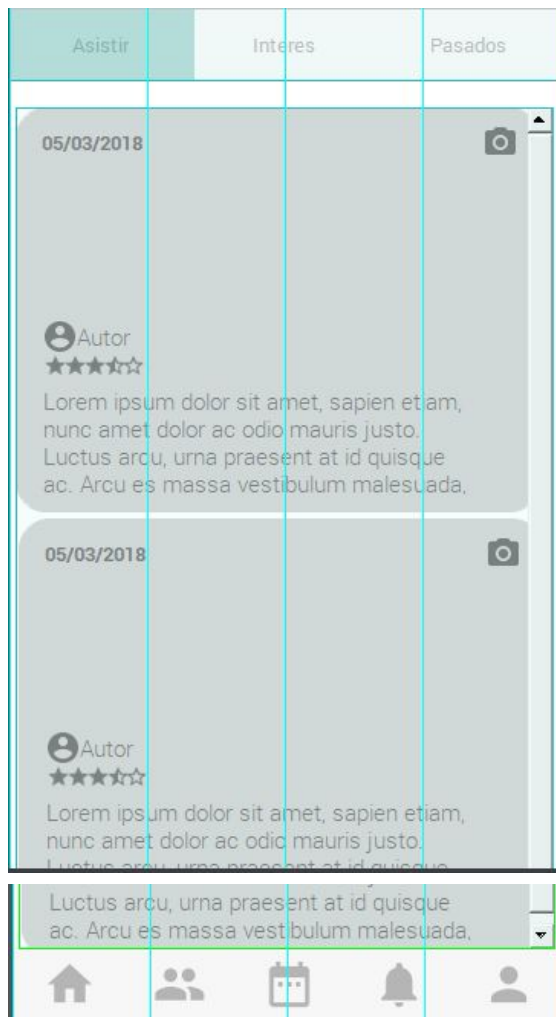
Perfil

En este activity se muestran los datos del usuario, como lo es su foto, nombre, apellidos, calificación, grupo de los que es miembro, eventos que ha creado, además, la opción de editar la opción de editar el perfil (lápiz) y configuración para configurar las categorías (a la par del lápiz). Este activity se muestra al seleccionar el quinto elemento del menú de navegación de la parte inferior.







Configuración de perfil

En este activity se muestra la lista de categorías en la parte superior (se pueden eliminar o agregar). Al seleccionar una categoría, se presentará una lista de subcategorías de la categoría, estas se pueden agregar o eliminar. En el prototipo se muestran como eventos, fue un error, en realidad son categorías.



Calendario

En este activity se muestra la lista de eventos a los que el usuario asistirá, a los que el usuario está interesado y los que han pasado. Este activity se muestra al seleccionar el tercer elemento del menú de navegación de la parte inferior.

Menciones	Grupos	Administrador
 <p>Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa</p>		
 <p>Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa</p>		
 <p> Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque </p>		<input checked="" type="checkbox"/> 

Notificaciones

En este activity se muestran notificaciones. La primera notificación es relacionada con las menciones, la segunda notificación está relacionada con grupos y la tercera notificación está relacionado con administradores (nuevos eventos, donde se aprueban o no).

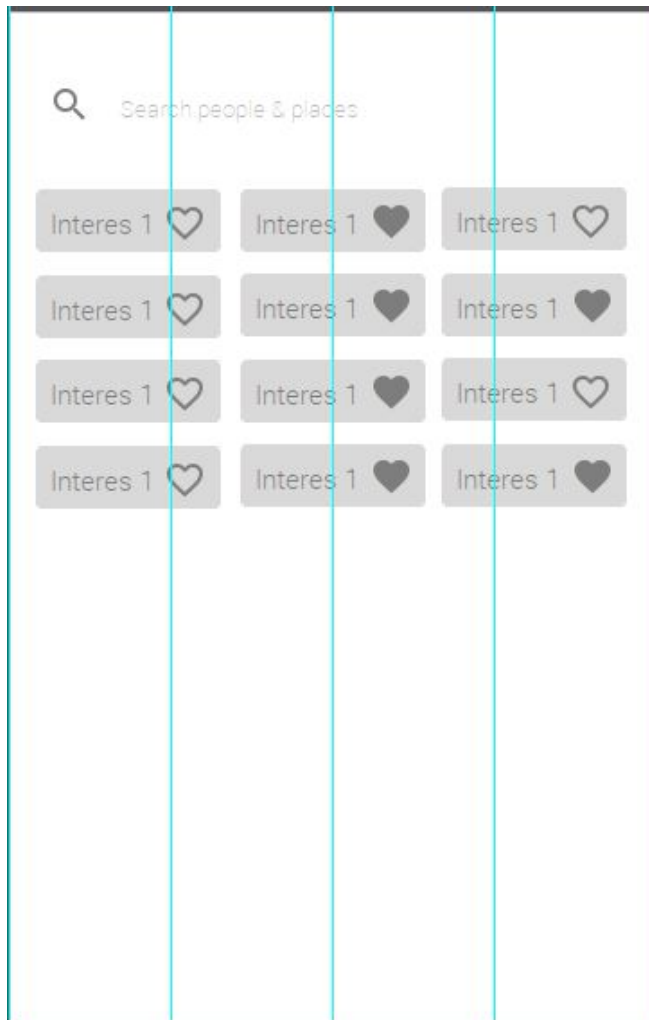
The image shows a mobile application interface for adding a new group. The main form is divided into three sections: a top section for a profile picture (indicated by a diagonal line), a middle section for the group name and description, and a bottom section for adding interests. The bottom section contains a button labeled "Agregar Intereses". The form is overlaid on a grid background. At the bottom of the screen, there is a navigation bar with a blue circle containing a white plus sign.

Form fields and elements:

- Top section: Profile picture placeholder (diagonal line).
- Middle section:
 - Settings icon (gear) and "Nombre del grupo" field.
 - "Escribe descripción" field.
- Bottom section:
 - "Agregar Intereses" button.
- Bottom navigation bar: Blue circle with a white plus sign (+).

Agregar nuevo grupo

En este activity se permite agregar un nuevo grupo, al dar datos como la imagen, nombre de grupo, descripción, y agregar intereses. Se agrega mediante el símbolo de “+”.



Agregar intereses a los eventos o grupos

En este activity se muestra una lista de intereses que se pueden agregar a un evento, puede usarse un buscador para facilitar la búsqueda.

Nombre

Escribe descripción

Día del evento

Hora

Escribir dirección

Agregar Intereses

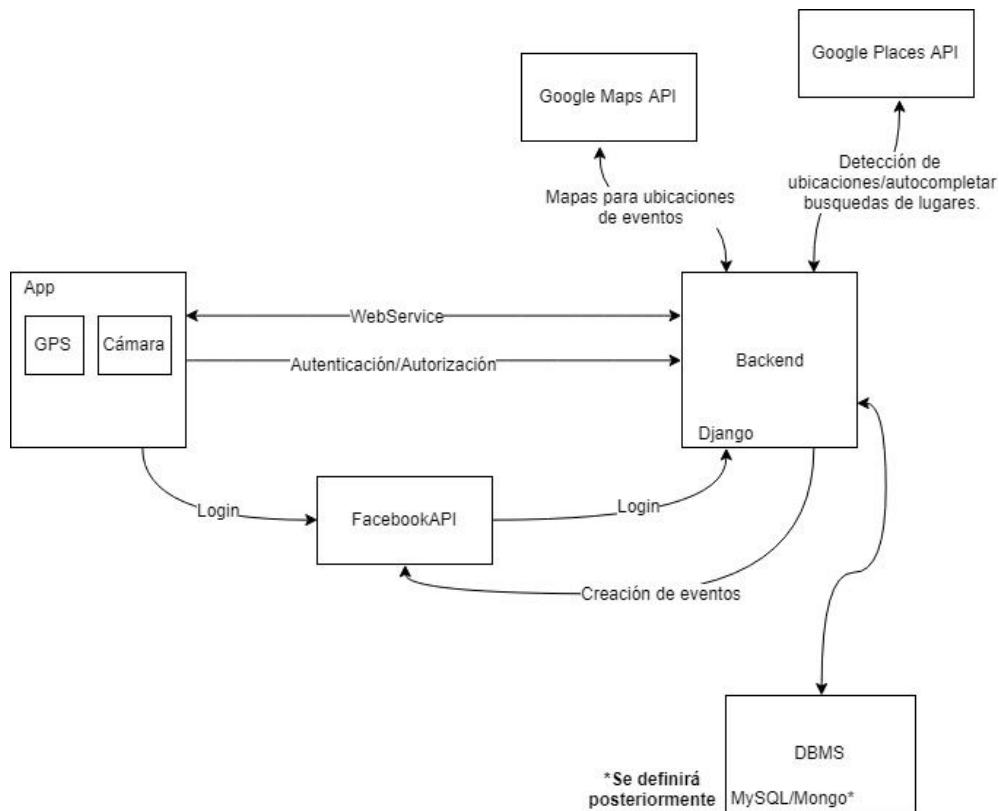
+

Agregar nuevo evento

En este activity se permite agregar un nuevo evento, al dar datos como la imagen, nombre de evento, descripción, y agregar intereses, fecha, hora, dirección (se utiliza google maps). Se agrega mediante el símbolo de “+”.

Descripción de diseño de alto nivel

Diagrama de arquitectura



Como se aprecia en la imagen, la aplicación consiste de un backend, el cual posee sus propias rutinas para el manejo de los datos, en la implementación final, el backend se encarga de guardar los valores de la latitud y la longitud de los lugares en donde se llevará a cabo el evento, mientras que la aplicación hace conexión con los API de Google Places y Google Maps.

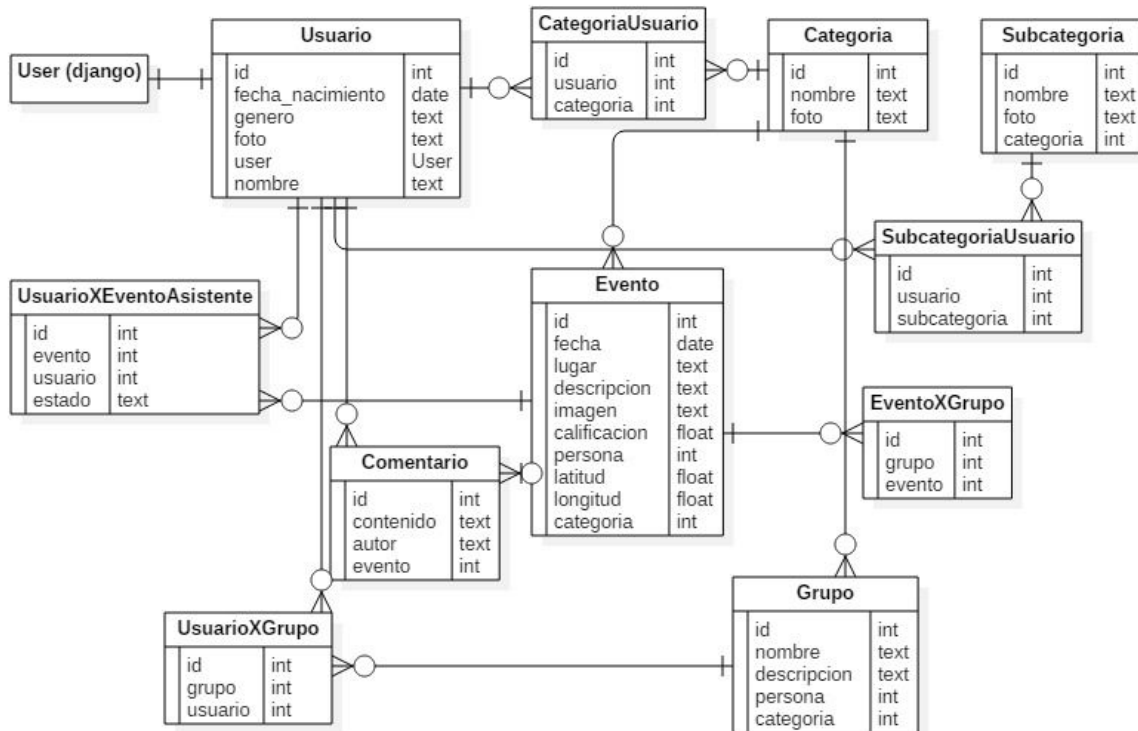
El backend fue desarrollado en python, django rest framework, en el cual se hace el uso de una base de datos sqlite, la cual es utilizada mediante serializadores, los cuales son los encargados de poseer la lógica del manejo de datos del backend, para ser procesados por una vista y posteriormente utilizar la base de datos para guardar cambios o retornar información.

La aplicación también se conecta con el API de Facebook para hacer el inicio de sesión.

Para poder hacer solicitudes al backend desarrollado en django, se debe iniciar sesión en el sistema, para obtener un token, el cual deberá ser utilizado para hacer las peticiones posteriores.

El backend se encuentra disponible en heroku, lugar desde el cual se podrá acceder a los datos desde cualquier dispositivo.

Diagrama de base de datos



En la imagen anterior, se pueden apreciar las siguientes entidades:

- **User (django):** es la tabla propia de django que se utiliza para realizar el registro e inicio de sesión propio de la aplicación.
- **Usuario:** es una tabla que tiene conexión con el User de django, de manera que se puedan agregar más columnas para el manejo de la aplicación.
- **Categoría:** es una tabla que guarda las categorías que se visualizan en la aplicación.
- **Subcategoría:** es una tabla que guarda las subcategorías que se visualizan en la aplicación.
- **CategoriaUsuario:** es una tabla que guarda una relación entre categorías y usuarios, de manera que este tenga asociadas algunas para el uso de la aplicación.
- **Evento:** es una tabla que guarda la información de un evento, este puede tener comentarios y puede estar asociado a grupos, por medio de la tabla EventoGrupo.
- **UsuarioXEventoAsistente:** es una tabla que se utiliza para guardar a los asistentes de un evento determinado.
- **Comentario:** es una tabla que guarda los comentarios para un evento.
- **SubcategoriaUsuario:** es una tabla que guarda las subcategorías para un usuario.
- **EventoXGrupo:** es una tabla que guarda los eventos para un determinado grupo.
- **Grupo:** es una tabla que guarda los grupos de la aplicación.
- **UsuarioXGrupo:** es una tabla que guarda a los usuarios pertenecientes a un grupo.

Descripción de los web services

A continuación, se describirán los web services utilizados en la aplicación.

Listar usuarios

- Descripción: lista todos los usuarios dentro del sistema.
- Url: api/usuarios/listar
- Parámetros: token.
- Método http: GET.

Crear usuario

- Descripción: crea un nuevo usuario en el sistema.
- Url: api/usuarios/crear
- Parámetros: token, nombre, fecha de nacimiento, género, foto.
- Método http: POST.

Consultar un usuario

- Descripción: consulta un usuario por un id.
- Url: api/usuarios/(?P<pk>\d+)/detalle
- Parámetros: token, id de usuario.
- Método http: GET.

Editar un usuario

- Descripción: edita un usuario en el sistema.
- Url: api/usuarios/(?P<pk>\d+)/editar
- Parámetros: token, id de usuario, nombre, fecha de nacimiento, género, foto.
- Método http: PUT.

Eliminar un usuario

- Descripción: elimina un usuario del sistema.
- Url: api/usuarios/(?P<pk>\d+)/eliminar
- Parámetros: token, id de usuario.
- Método http: DELETE.

Listar categorías

- Descripción: lista todas las categorías dentro del sistema.
- Url: `api/categorias/listar`
- Parámetros: token.
- Método http: GET.

Crear categoría

- Descripción: crea una nueva categoría en el sistema.
- Url: `api/categorias/crear`
- Parámetros: token, nombre de categoría, foto de categoría.
- Método http: POST.

Consultar una categoría

- Descripción: consulta una categoría por un id.
- Url: `api/categorias/(?P<pk>\d+)/detalle`
- Parámetros: token, id de categoría.
- Método http: GET.

Editar una categoría

- Descripción: edita una categoría.
- Url: `api/categorias/(?P<pk>\d+)/editar`
- Parámetros: token, id de categoría, nombre de categoría, foto de categoría.
- Método http: PUT.

Eliminar una categoría

- Descripción: elimina una categoría del sistema.
- Url: `api/categorias/(?P<pk>\d+)/eliminar`
- Parámetros: token, id de categoría.
- Método http: DELETE.

Listar subcategorías

- Descripción: lista las subcategorías del sistema.
- Url: `api/subcategorias/listar`
- Parámetros: token.
- Método http: GET.

Crear subcategoría

- Descripción: crea una subcategoría en el sistema.
- Url: `api/subcategorias/crear`
- Parámetros: token, nombre, foto, categoría.
- Método http: POST.

Consultar una subcategoría

- Descripción: obtiene una subcategoría.
- Url: `api/subcategorias/(?P<pk>\d+)/detalle`
- Parámetros: token, id.
- Método http: GET.

Editar una subcategoría

- Descripción: edita una subcategoría.
- Url: `api/subcategorias/(?P<pk>\d+)/editar`
- Parámetros: token, id, nombre, foto, categoría.
- Método http: PUT.

Eliminar una subcategoría

- Descripción: elimina una subcategoría.
- Url: `api/subcategorias/(?P<pk>\d+)/eliminar`
- Parámetros: token, id.
- Método http: DELETE.

Listar comentarios

- Descripción: lista los comentarios.
- Url: api/comentarios/listar
- Parámetros: token.
- Método http: GET.

Crear comentario

- Descripción: crea un nuevo comentario.
- Url: api/comentarios/crear
- Parámetros: token, contenido, autor, evento.
- Método http: POST.

Consultar un comentario

- Descripción: consulta un comentario.
- Url: api/comentarios/(?P<pk>\d+)/detalle
- Parámetros: token, id
- Método http: GET.

Editar un comentario

- Descripción: edita un comentario.
- Url: api/comentarios/(?P<pk>\d+)/editar
- Parámetros: token, id, contenido, autor, evento.
- Método http: PUT.

Eliminar un comentario

- Descripción: elimina un comentario.
- Url: api/comentarios/(?P<pk>\d+)/eliminar
- Parámetros: token, id.
- Método http: DELETE.

Listar eventos

- Descripción: lista los eventos.
- Url: api/eventos/listar
- Parámetros: token.
- Método http: GET.

Crear evento

- Descripción: crea un evento.
- Url: api/eventos/crear
- Parámetros: token, fecha, lugar, descripción, imagen, calificación, latitud, longitud, persona, categoría.
- Método http: POST.

Consultar un evento

- Descripción: consulta un evento.
- Url: api/eventos/(?P<pk>\d+)/detalle
- Parámetros: token, id.
- Método http: GET.

Editar un evento

- Descripción: edita un evento.
- Url: api/eventos/(?P<pk>\d+)/editar
- Parámetros: token, id, fecha, lugar, descripción, imagen, calificación, latitud, longitud, persona, categoría.
- Método http: PUT.

Eliminar un evento

- Descripción: elimina un evento.
- Url: api/eventos/(?P<pk>\d+)/eliminar
- Parámetros: token, id.
- Método http: DELETE.

Listar grupos

- Descripción: lista los grupos.
- Url: api/grupos/listar
- Parámetros: token.
- Método http: GET.

Crear grupo

- Descripción: crea un nuevo grupo.
- Url: `api/grupos/crear`
- Parámetros: token, nombre, descripción, persona, categoría.
- Método http: POST.

Consultar un grupo

- Descripción: consultar un grupo.
- Url: `api/grupos/(?P<pk>\d+)/detalle`
- Parámetros: token, id.
- Método http: GET.

Editar un grupo

- Descripción: edita un grupo.
- Url: `api/grupos/(?P<pk>\d+)/editar`
- Parámetros: token, id, nombre, foto, categoría, persona.
- Método http: PUT.

Eliminar un grupo

- Descripción: elimina un grupo.
- Url: `api/grupos/(?P<pk>\d+)/eliminar`
- Parámetros: token, id.
- Método http: DELETE.

Listar usuarios asociados a evento

- Descripción: lista usuarios asociados a un evento.
- Url: `api/usuarioevento/listar`
- Parámetros: token.
- Método http: GET.

Crear usuario asociado a evento

- Descripción: crea un usuario asociado a un evento.
- Url: `api/usuarioevento/crear`
- Parámetros: token, usuario, evento.
- Método http: POST.

Listar usuarios asociados a grupo

- Descripción: lista los usuarios asociados a un grupo.
- Url: `api/usuariogrupo/listar`
- Parámetros: token.
- Método http: GET.

Crear usuario asociado a grupo

- Descripción: crea un usuario asociado a un grupo.
- Url: `api/usuariogrupo/crear`
- Parámetros: token, usuario, grupo.
- Método http: POST.

Listar categorías asociadas a un usuario

- Descripción: lista las categorías asociadas a un usuario.
- Url: `api/categoriausuario/listar`
- Parámetros: token.
- Método http: GET.

Crear categoría asociada a un usuario

- Descripción: crea una categoría asociada a un usuario.
- Url: `api/categoriausuario/crear`
- Parámetros: token, categoría, usuario.
- Método http: POST.

Listar subcategorías asociadas a un usuario

- Descripción: lista las subcategorías asociadas a un usuario.
- Url: `api/subcategoriausuario/listar`
- Parámetros: token.
- Método http: GET.

Crear subcategoría asociada a un usuario

- Descripción: crea una subcategoría asociada a un usuario.
- Url: `api/subcategoriausuario/crear`
- Parámetros: token, subcategoría, usuario.
- Método http: POST.

Asistir a un evento

- Descripción: asocia un usuario a un evento con la acción de asistir.
- Url: `api/usuarioevento/asistir`
- Parámetros: token, usuario.
- Método http: POST.

Estar interesado en un evento

- Descripción: asocia un usuario a un evento con la acción de estar interesado.
- Url: `api/usuarioevento/interesado`
- Parámetros: token, usuario.
- Método http: POST.

Obtener eventos cercanos

- Descripción: obtiene los eventos más cercanos
- Url: api/eventocerca
- Parámetros: token, latitud, longitud.
- Método http: GET.

Obtener eventos cercanos por categoría

- Descripción: obtiene los eventos más cercanos, dada una categoría.
- Url: api/ecercacategoria
- Parámetros: token, latitud, longitud, categoría.
- Método http: GET.

Obtener eventos para un usuario

- Descripción: obtiene los eventos para un usuario.
- Url: api/usuarioevento/obtenerEvento
- Parámetros: token, usuario.
- Método http: GET.

Obtener un usuario de django

- Descripción: obtiene un usuario de django.
- Url: api/user/get
- Parámetros: token, id.
- Método http: GET.

Obtener usuario actual

- Descripción: obtiene el usuario que esté iniciado en el sistema.
- Url: api/usuarioactual
- Parámetros: token.
- Método http: GET.

Obtener grupos por categoría

- Descripción: obtiene los grupos, dada una categoría.
- Url: `api/grupos/categoria`
- Parámetros: token, categoría.
- Método http: GET.






Obtener eventos por categoría

- Descripción: obtiene los eventos, dada una categoría.
- Url: `api/eventos/categoria`
- Parámetros: token, categoría.
- Método http: GET.

Interacción con sistemas externos

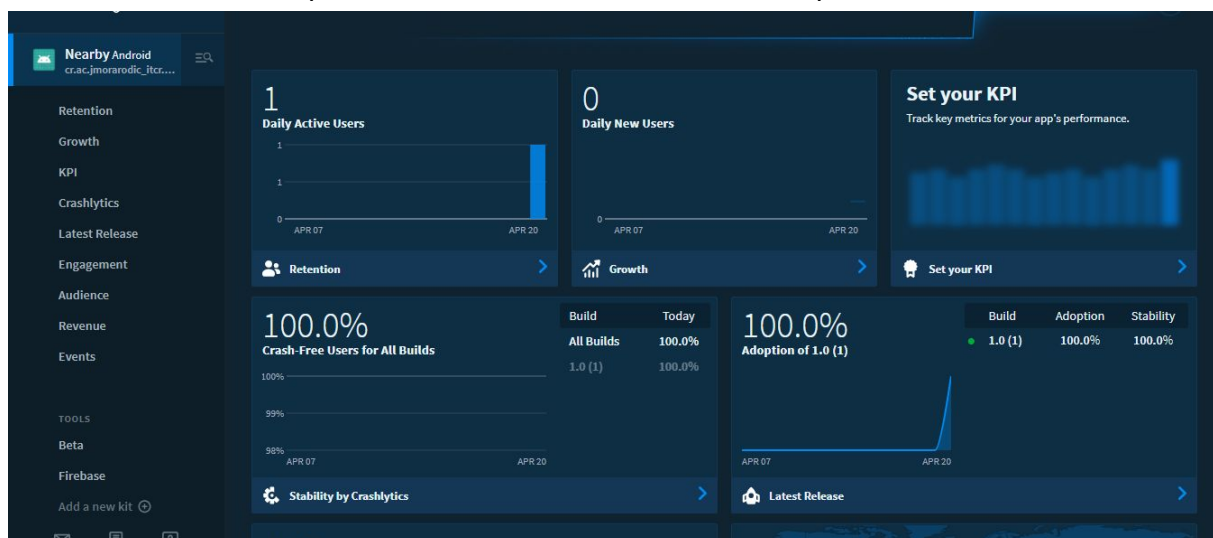
Mixpanel

Se hizo conexión con Mixpanel, por medio de un token es posible dar seguimiento al uso que hace un usuario a la aplicación, por ejemplo, al entrar a alguna sección específica de esta, es posible ver qué acción realizó, desde qué ciudad, provincia y país.

	App Session	3 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo eventos	4 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo el perfil	4 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo notificaciones	4 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo eventos	4 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo eventos	6 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	App Session	7 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo el perfil	7 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo notificaciones	8 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo el perfil	8 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	Viendo notificaciones	8 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—
	App Session	8 min. ago	—	Cartago	Costa Rica	096ba996-b752-43b6-9681-cf0c9693a155	—

Fabric

Se hizo conexión con Fabric, por medio del plugin que tiene android studio para su manejo, de manera que se le pueda dar seguimiento a eventos de la aplicación, por ejemplo, cuando esta falla. También, se pueden ver los usuarios activos de la aplicación.



Facebook

Se realizó conexión con el API de Facebook, de manera que se pueda iniciar sesión en la aplicación por este medio, se obtienen los datos del perfil de Facebook de la persona que inició sesión, para poder asignarlos al modelo que administra los datos de los usuarios del api en django.

Google Places

Se realizó conexión con el API de Google Places, de manera que se pueda obtener la latitud y la longitud de un lugar deseado, al poder digitar su nombre por medio de una entrada de texto, así, esos valores podrán ser asignados al modelo de Eventos en el api de la aplicación.

Google Maps

Se realizó conexión con el API de Google Maps, de manera que se pueda visualizar un mapa y elegir un lugar para asignar al modelo de Eventos en el api de la aplicación.

Cloudinary

Se realizó una conexión con cloudinary, de manera que se pudiese tener un medio en el cual guardar las imágenes de la aplicación, así, para guardar imágenes, estas se guardan en cloudinary, se genera el enlace de la misma y, posteriormente, se asigna al modelo correspondiente en el api de django. Para descargar imágenes, se obtiene la referencia a la misma a través del api, para poder descargarlas y cargarlas en los elementos de interfaz gráfica de android.