

FREEMBE

DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

PREPARADO POR:

JOSÉ MIGUEL MORA RODRÍGUEZ

KARINA ZELEDÓN PINELL

DYLAN RODRÍGUEZ BARBOZA

JUNIO, 2018

Contenido

Propósito.....	3
Requerimientos.....	3
Funcionalidades	4
UX, UI (wireframes)	5
APP MOVIL	5
BACKOFFICE	12
Descripción de diseño de alto nivel.....	19
Diagrama arquitectura	19
Diagrama de base de datos	20
Descripción de los web-services.....	20
Anuncios	20
Autenticación.....	21
Categorías	21
Comentarios.....	22
Emails	22
Favoritos	23
Subcategorías.....	23
SubcategoriasXCategorias	24
Teléfonos	24
Usuarios.....	25
Interacción con sistemas externos	26
Mixpanel	26
Fabric	26
Google.....	26
Google Places.....	26
Google Maps.....	26
Cloudinary	26

Propósito

Freembe es una aplicación desarrollada para dispositivos android. Consiste en un medio de información donde se muestran servicios brindados de forma independiente por los usuarios. Esta aplicación busca que las personas que labora de forma independiente den a conocer su trabajo, además permite que los usuarios creen servicios, así como visualizar servicios según sus necesidades.

La aplicación cuenta con quince ventanas, mediante las cuales se pueden hacer acciones como: crear anuncios, buscar anuncios, guardar anuncios favoritos, ver perfil del usuario, registrarse en la aplicación, ver información de contactos de otros usuarios. En la aplicación se hace uso del GPS y acceso a la memoria interna del dispositivo. Además, también se cuenta con tres conexiones a API, Google, Google Places y Google Maps. La aplicación cuenta un API propio desarrollado en Ruby on Rails, en el cual se encuentra el backend de la aplicación, aquí se encuentra la lógica, así como el consumo de datos de la aplicación.

Requerimientos

Se requería que la aplicación cumpliera con los siguientes requisitos:

- Desarrollo en Android nativo
- Deben usar Material Design para la app de Android
- Hacer un landing page de mercadeo para la app y publicarlo en Internet
- Desarrollo de backend propio (no BaaS), que sea accesible por dominio público. Pueden usar cualquier framework para el Backend, excepto Python/Django.
- El backend debe tener un backoffice web de administración, con su propia autenticación
- Deben usar user login, tanto en backoffice como en aplicación
- 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
- La app debe tener al menos 12 activities
- Incluir al menos: a. 2 patrones de despliegue de datos b. 1 patrón de búsqueda c. 1 patrón de filtro d. 1 patrón de ordenamiento e. 1 patrón de navegación
- Deben manejar métricas de uso con Mixpanel y Fabric
- Uso de AWS (EC2, S3 y Route53) para infraestructura (10% adicional en la nota de la progra)

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

- Hacer un inicio de sesión propio.
- Hacer un inicio de sesión con Google.
- Registrar un usuario a la aplicación.
- Visualizar anuncios.
- Implementación de buscador con filtro de fecha y precio.
- Visualizar categorías.
- Visualizar subcategorías.
- Visualizar anuncios por subcategorías.
- Visualizar en detalle el anuncio.
- Crear un anuncio.
- Visualizar información de contacto de otros usuarios.
- Comentar un anuncio.
- Listar anuncios favoritos y propios.
- Visualizar perfil del usuario.
- Editar perfil del usuario.
- Ver ubicación de un anuncio, utilizando Google Maps.

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.
- Inicio de sesión con Google.
- Listar categorías.
- Mostrar anuncio más popular.
- Buscador de anuncios con filtro de fecha y precio.
- Listar subcategorías dada a una categoría.
- Mostrar anuncios dada una subcategoría.
- Ver detalle de anuncio.
- Mostrar ubicación de un anuncio con Google Maps.
- Mostrar comentarios o reseñas de un anuncio
- Comentar un anuncio.
- Visualizar información de contacto de otros usuarios.
- Seleccionar como favorito un anuncio.
- Listar anuncios propios y favoritos.
- Crear anuncio, se podría utilizar la ubicación para colocar en el anuncio.
- Visualizar perfil del usuario.
- Editar información de contacto del usuario.

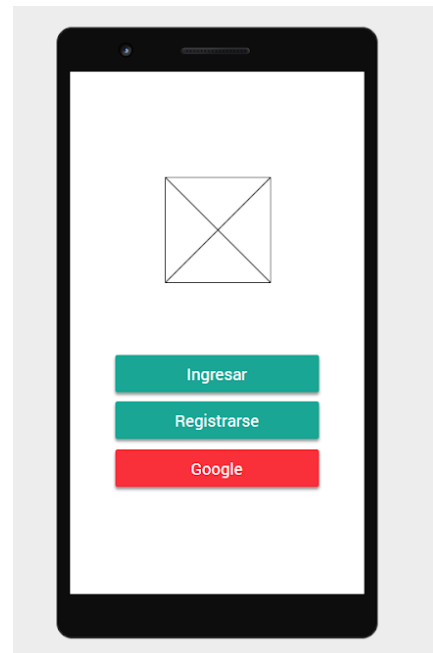
UX, UI (wireframes)

APP MOVIL

A continuación, se muestran los wireframes que se propusieron originalmente para la aplicación, estos corresponden a la app móvil:

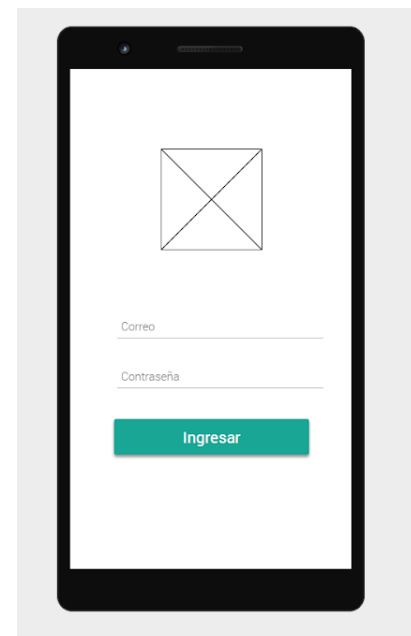
Pantalla inicio

En esta pantalla se tienen las opciones de ingresar, registrar o autenticarse con Google.



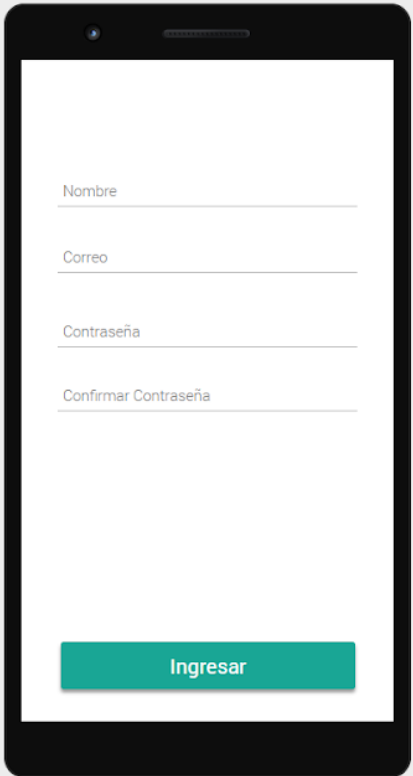
Pantalla ingresar

En esta pantalla se ingresan los datos para autenticarse e ingresar a la aplicación



Pantalla de registro

En esta pantalla se ingresan los datos para crear un nuevo usuario

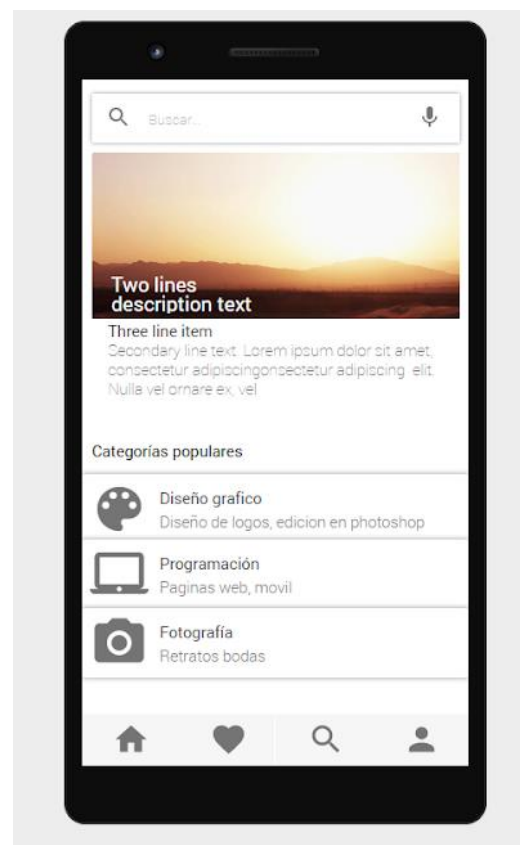


A mobile app registration screen with a white background and a black border. It features four input fields with labels: 'Nombre', 'Correo', 'Contraseña', and 'Confirmar Contraseña'. At the bottom, there is a green button labeled 'Ingresar'.

Pantalla principal

En esta pantalla se encuentra un buscador en el área superior, para buscar anuncios. Luego se encuentra una área que muestra el anuncio más reciente/popular y por último se encuentra las categorías predefinidas.

Además, cuenta con una barra de navegación.

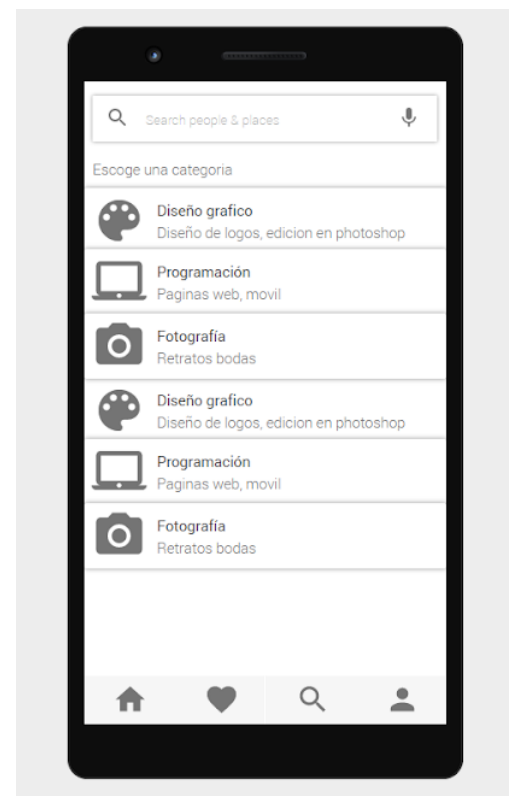


Pantalla buscar principal

Esta pantalla se accede haciendo clic a la lupa de la barra de navegación.

En esta pantalla se encuentra un buscador para buscar categorías mas fácilmente.

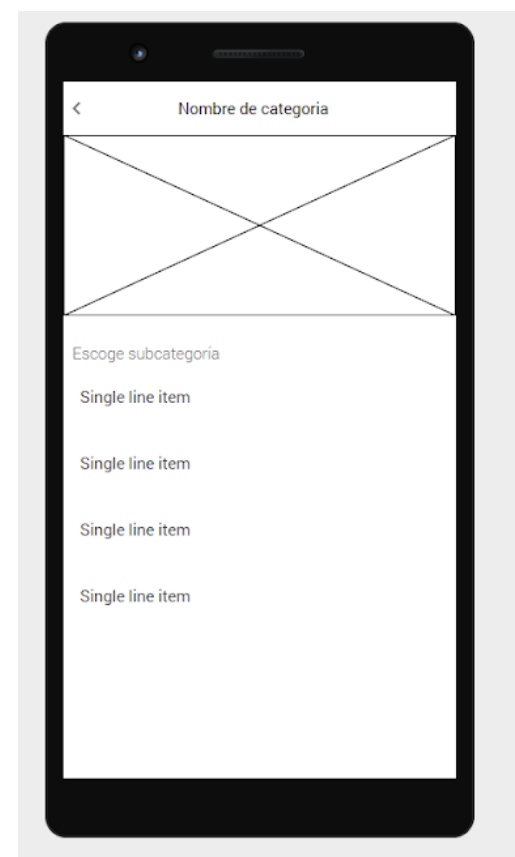
Además, debe seleccionar una categoría para continuar con la siguiente pantalla.



Pantalla buscar secundario

Para acceder esta pantalla se tuvo que haber seleccionado una categoría en la pantalla mencionado anteriormente.

En esta pantalla se muestra el nombre y una imagen general de la categoría y luego se enlistan las subcategorías.

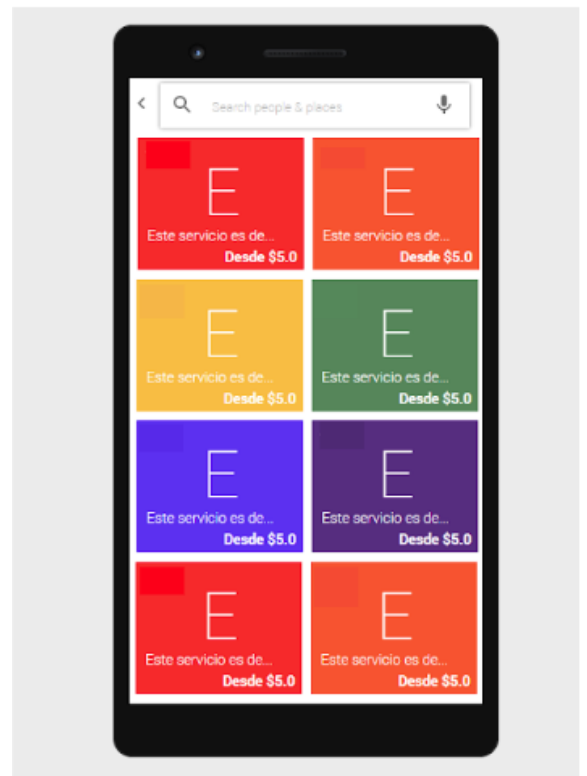


Pantalla buscador anuncios

Acá se muestran los anuncios según los datos seleccionados con anterioridad en las últimas dos secciones.

Esta sección cuenta con un buscador para filtrar los anuncios.

Los anuncios cuentan con una pequeña descripción y cuál es el precio que tiene.



Pantalla detalle del anuncio

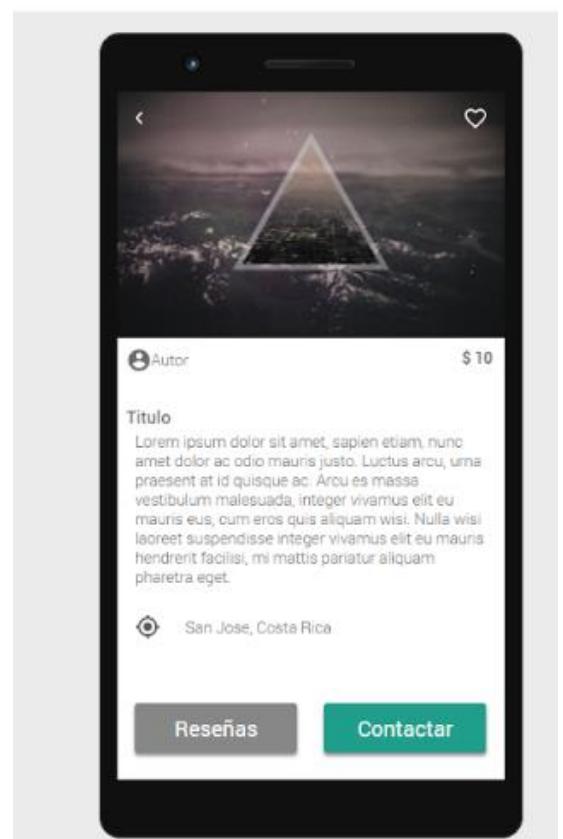
A esta pantalla se accede al seleccionar un anuncio.

Acá se puede visualizar una imagen y un corazón que funciona para seleccionar el servicio como favorito.

En el contenido se muestra el autor, el costo, el título, una descripción detallada y opcionalmente se puede tener una ubicación.

Por último, cuenta con dos botones, una para ver reseñas y otro para contactar al autor.

NOTA: El botón reseñas se quitó, y la pantalla reseñas se maneja en este activity.

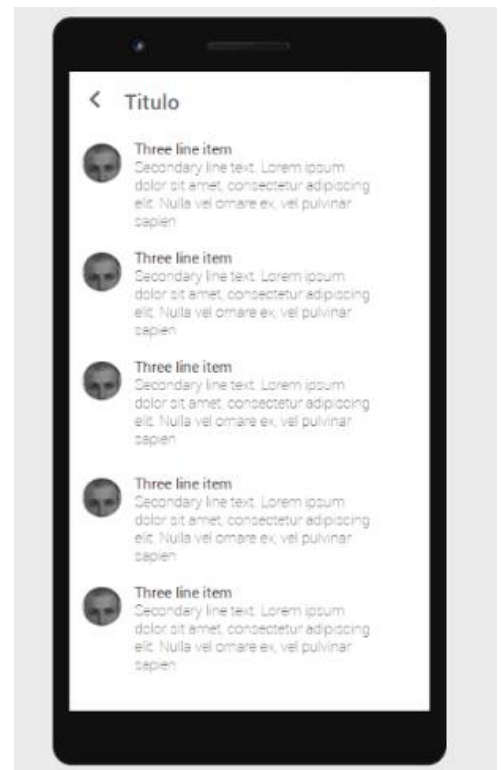


Pantalla reseñas de servicio

La funcionalidad de esta pantalla es la de ver comentarios de un servicio y se accede al seleccionar el botón de reseñas en el detalle del servicio.

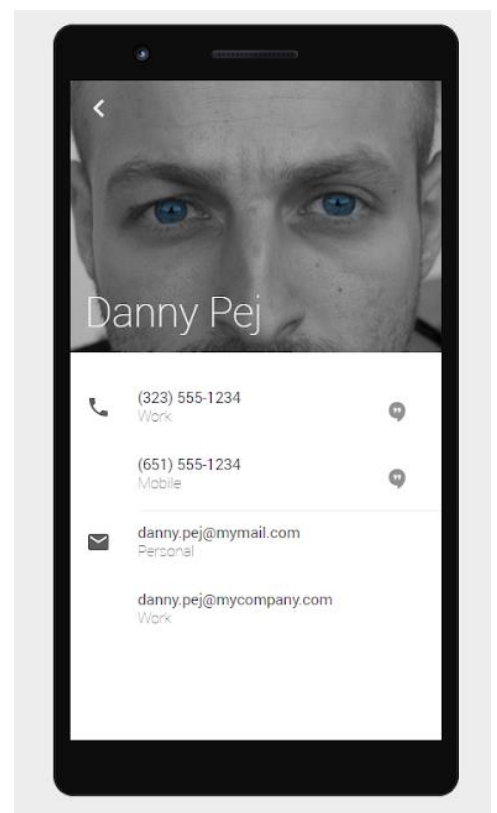
Acá se muestra el título del servicio y luego se muestra las reseñas. En esta pantalla no se visualiza, pero también se encuentra un área para dejar una nueva reseña.

Esta pantalla se implementa en la pantalla detalle del anuncio



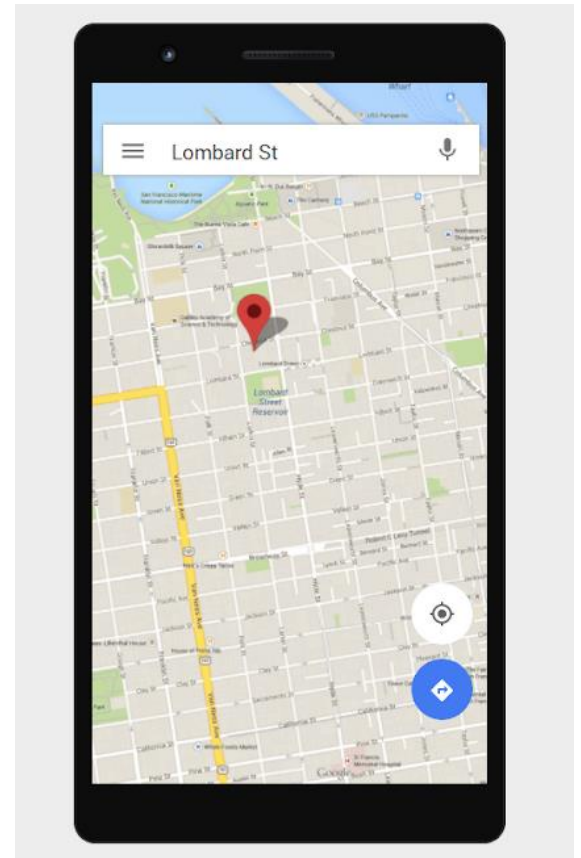
Pantalla para contactar a creador servicio

En esta pantalla se puede visualizar la información de contacto del creador del servicio, como numero de teléfono y email.



Pantalla de Maps

En esta pantalla se visualiza la ubicación de un servicio, se puede acceder al seleccionar la ubicación en el área de detalle del servicio.

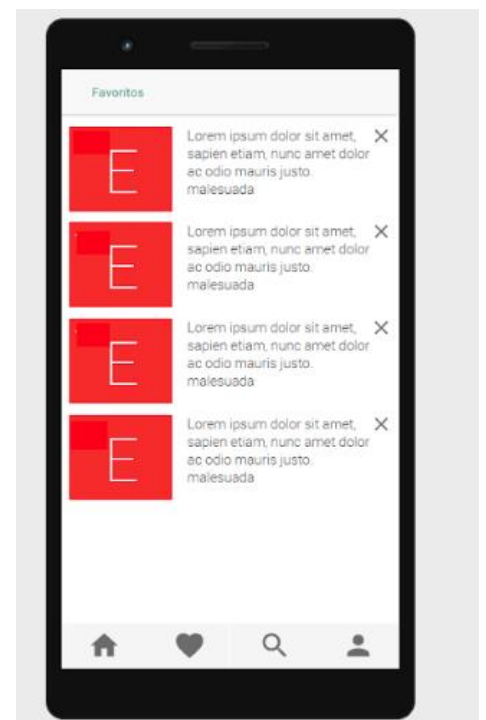


Pantalla favoritos

En esta pantalla se muestran los servicios seleccionados como favoritos y se accede al darle clic al símbolo de corazón.

La información que se muestra es la puntuación, una imagen, y una descripción y también la opción para eliminar el servicio de los favoritos.

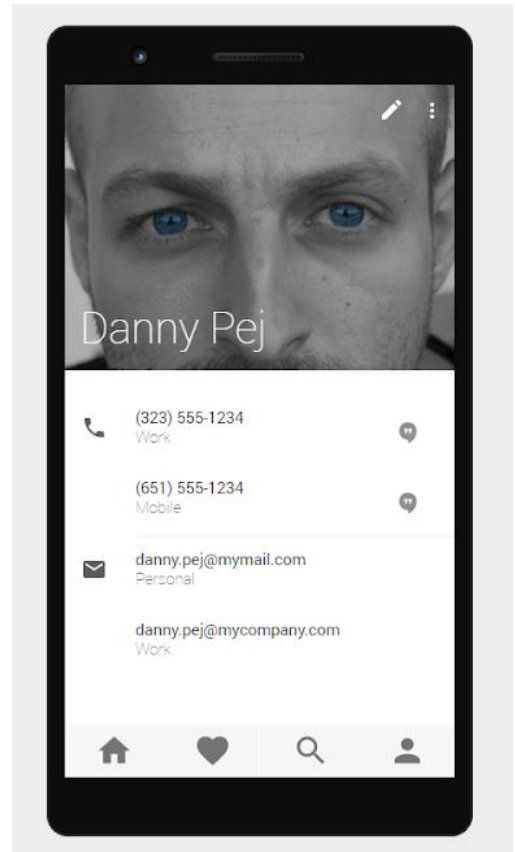
NOTA: Esta pantalla se uno a mis servicios, se maneja con tab superior para pasar de una pantalla a otra. Ambas pantallas contienen la misma información



Pantalla perfil

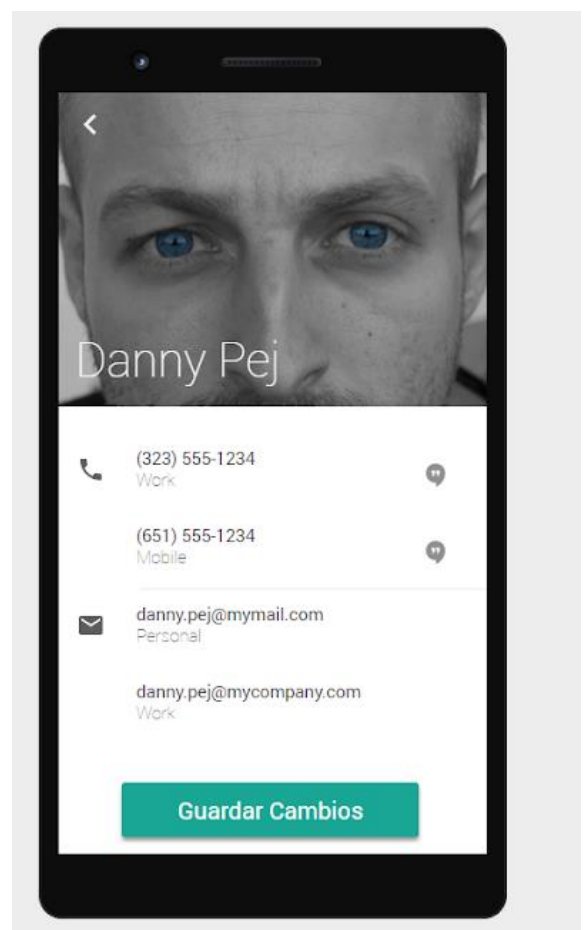
Acá se muestra una imagen, y la información de contacto, y se accede al darle clic al último icono de la barra de navegación.

Además, muestra un icono de lápiz para editar para editar el perfil.



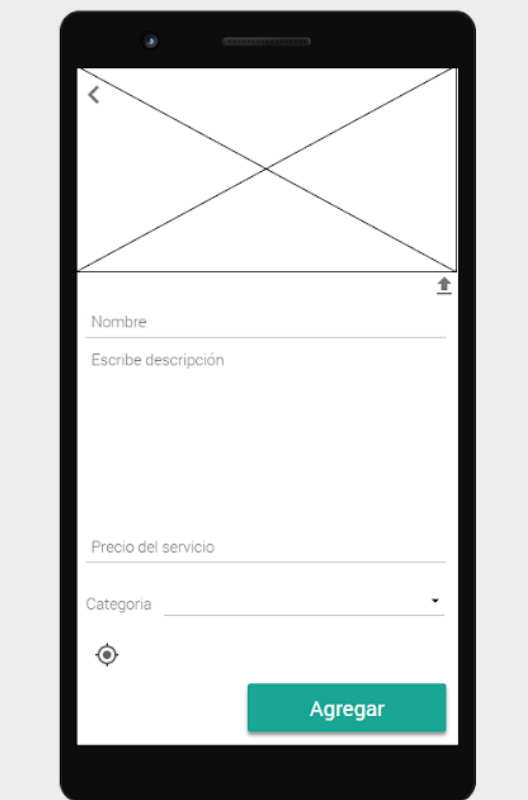
Pantalla editar perfil

Acá se editan los datos de contacto del usuario y se muestra un botón para confirmar cambios.



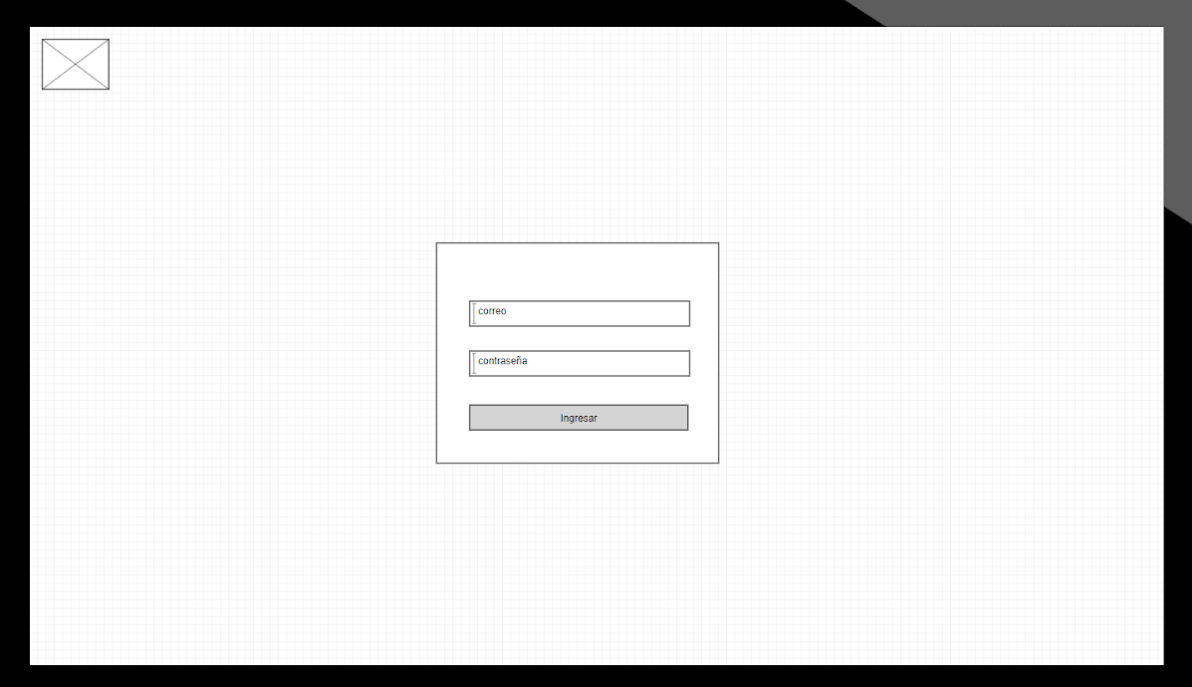
Pantalla nuevo servicio

En esta pantalla se agrega un nuevo servicio. Se agrega una imagen, nombre, descripción, precio, una categoría y opcionalmente una ubicación. Para confirmar la creación se utiliza el botón agregar.



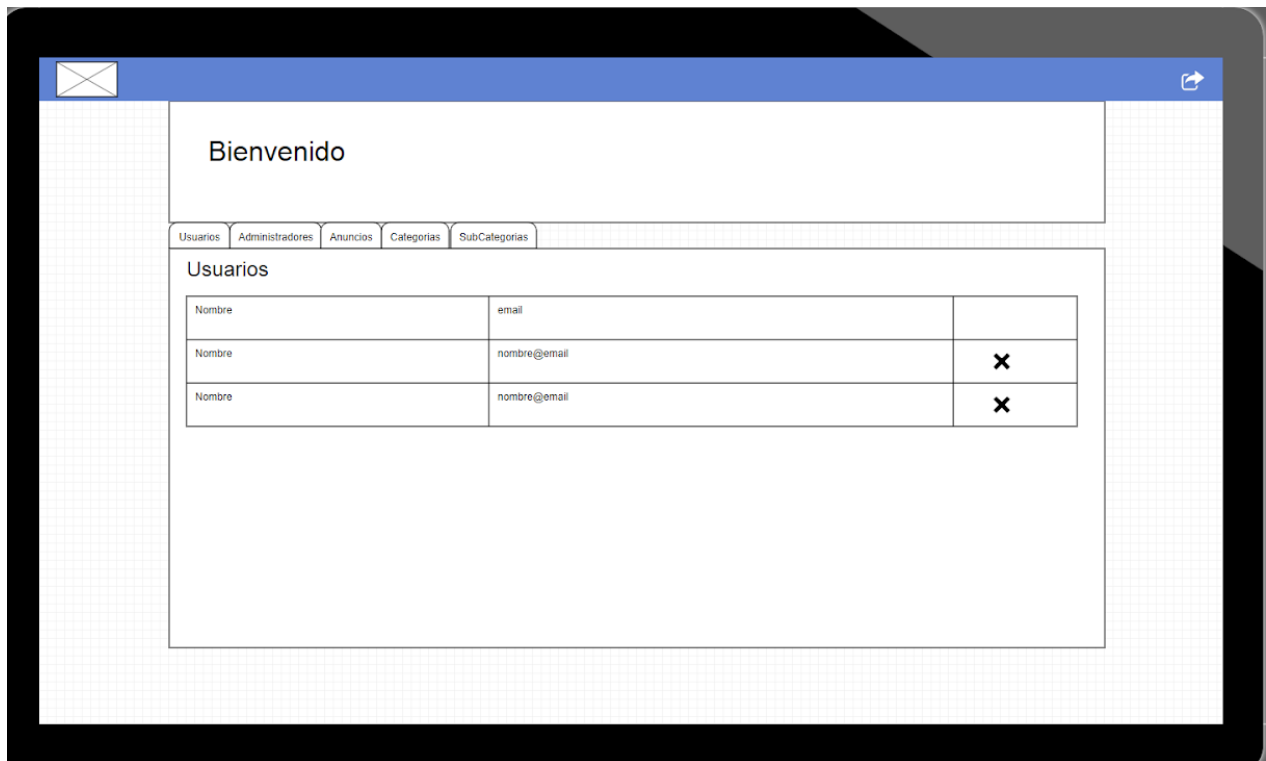
A mobile application screen for adding a new service. The screen features a back arrow in the top left corner and a placeholder for an image at the top. Below the image placeholder is an upload icon. The form includes input fields for 'Nombre', 'Escribe descripción', and 'Precio del servicio'. There is a dropdown menu for 'Categoría' and a location icon. A green 'Agregar' button is located at the bottom right.

BACKOFFICE

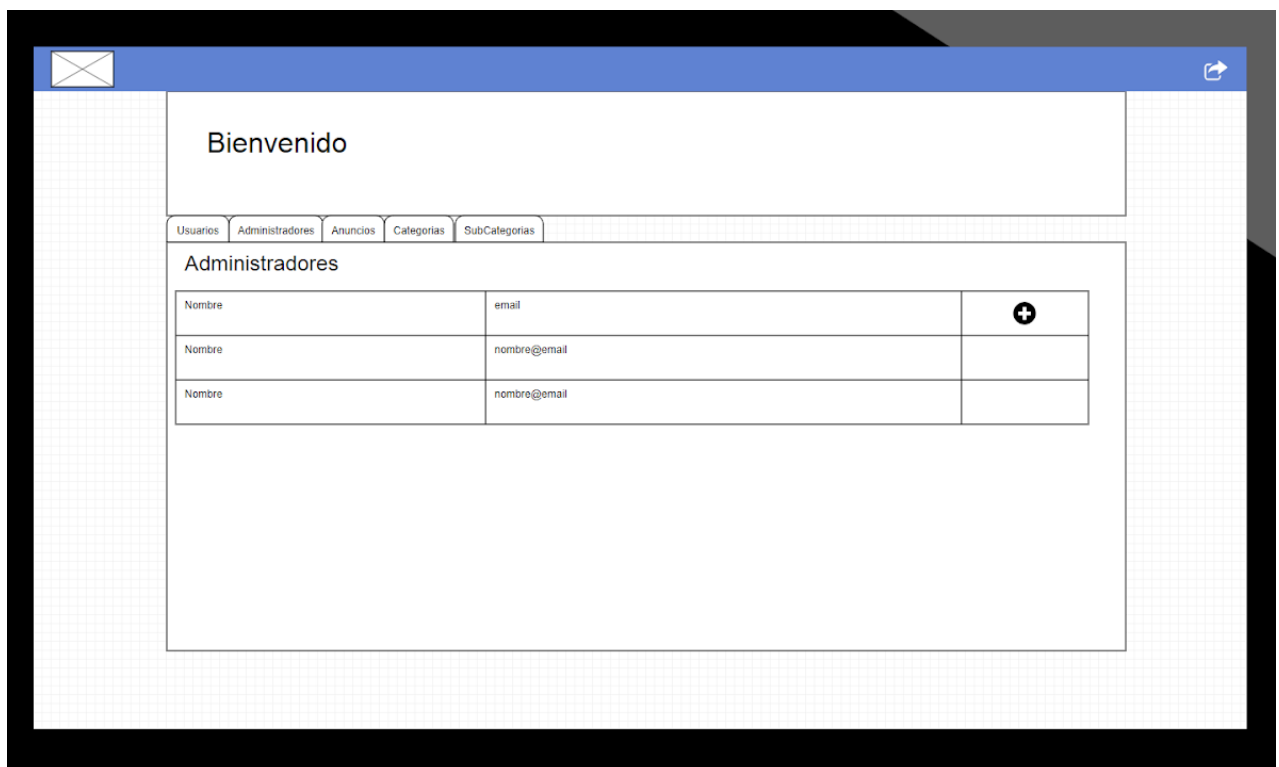


A backend login screen with a grid background. In the top left corner, there is a placeholder for a logo. Centered on the screen is a login form with two input fields labeled 'correo' and 'contraseña', and a button labeled 'Ingresar'.

Login



Principal – Usuarios



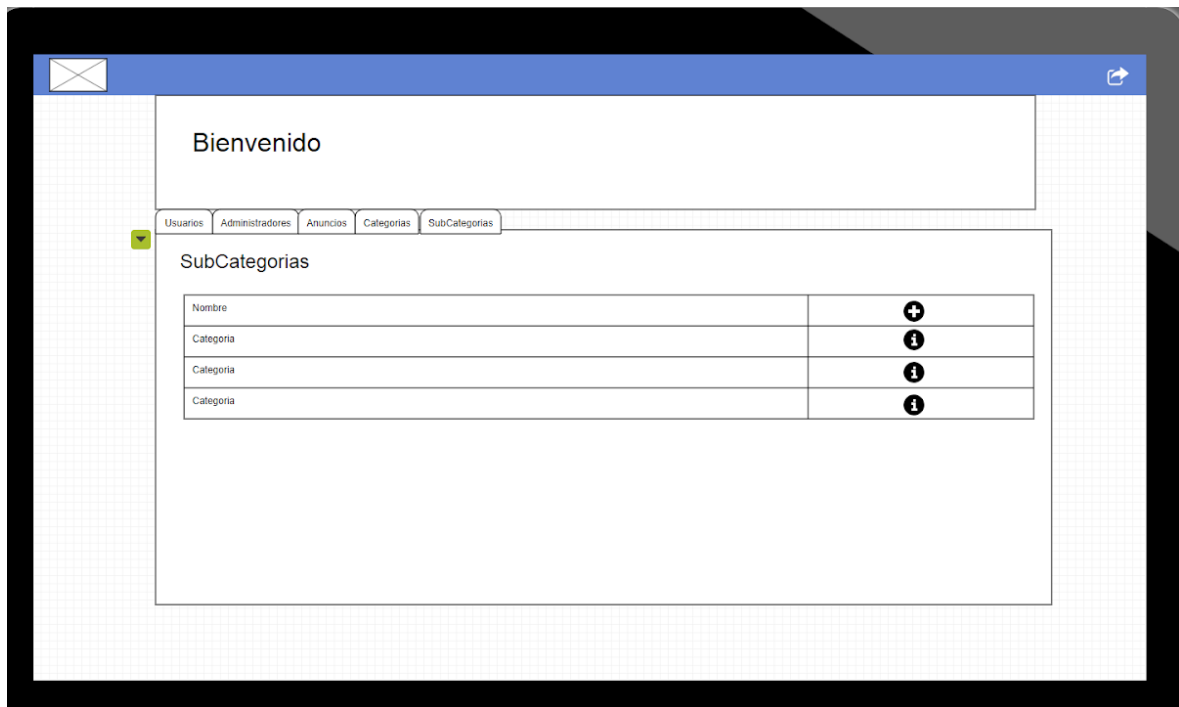
Principal – Administradores



Principal – Anuncios



Principal – Categorías



Principal – Subcategorías

The modal form is titled 'Nuevo Administrador' and features a close button (X) in the top right corner. It contains three input fields: 'Nombre', 'Email', and 'Contraseña'. At the bottom right, there are two buttons: 'Agregar' (highlighted in blue) and 'Cancelar' (gray).

Nuevo Administrador X

Nombre

Email

Contraseña

Agregar Cancelar

Agregar Administrador (Modal)

×

Titulo

Precio

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est.

Eliminar

Cancelar

Detalle Anuncio (Modal)

×

Nueva Categoría

Nombre

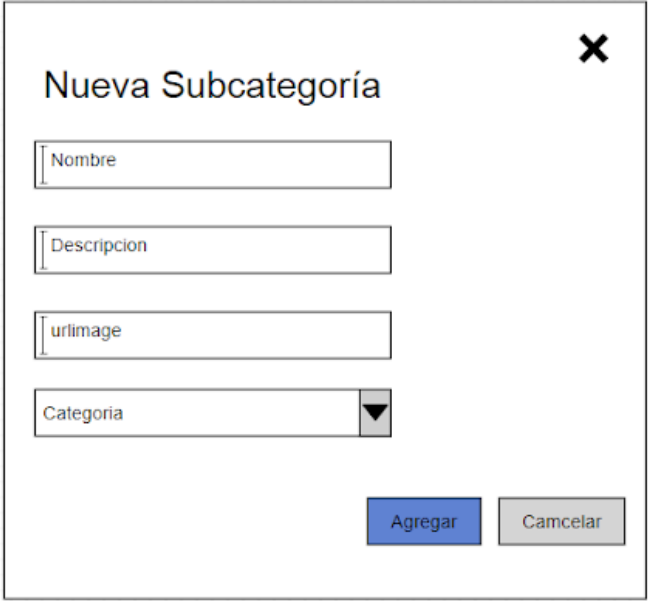
Descripcion

urlimage

Agregar

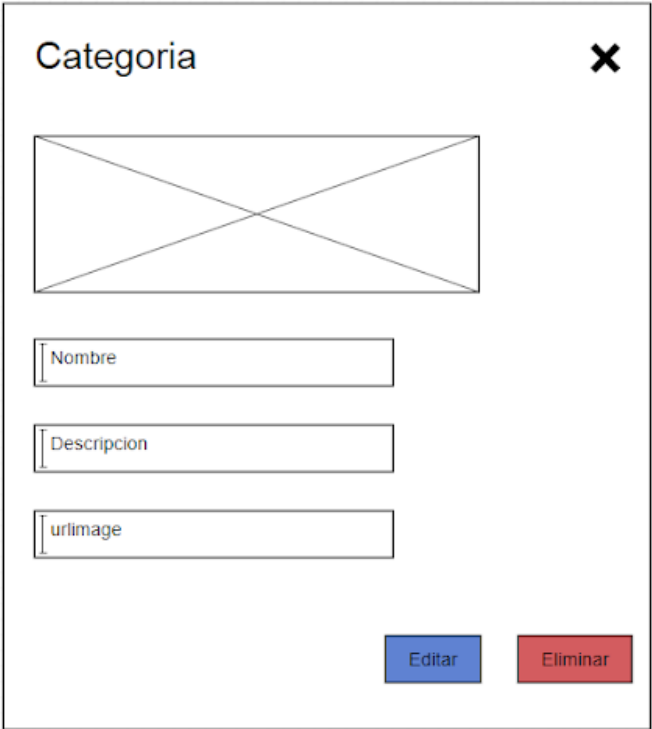
Cancelar

Nueva Categoría (Modal)



A modal window titled "Nueva Subcategoría" with a close button (X) in the top right corner. The form contains four input fields: "Nombre", "Descripción", "urlimage", and "Categoría" (a dropdown menu). At the bottom right, there are two buttons: "Agregar" (blue) and "Cancelar" (gray).

Nueva Subcategoría (Modal)

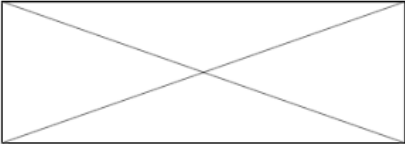


A modal window titled "Categoría" with a close button (X) in the top right corner. The form features a large rectangular placeholder with a diagonal cross, likely for a category image. Below this are three input fields: "Nombre", "Descripción", and "urlimage". At the bottom right, there are two buttons: "Editar" (blue) and "Eliminar" (red).

Detalle Categoría (Modal)

SubCategoria

✕



Nombre

Descripcion

urlimage

Categoria

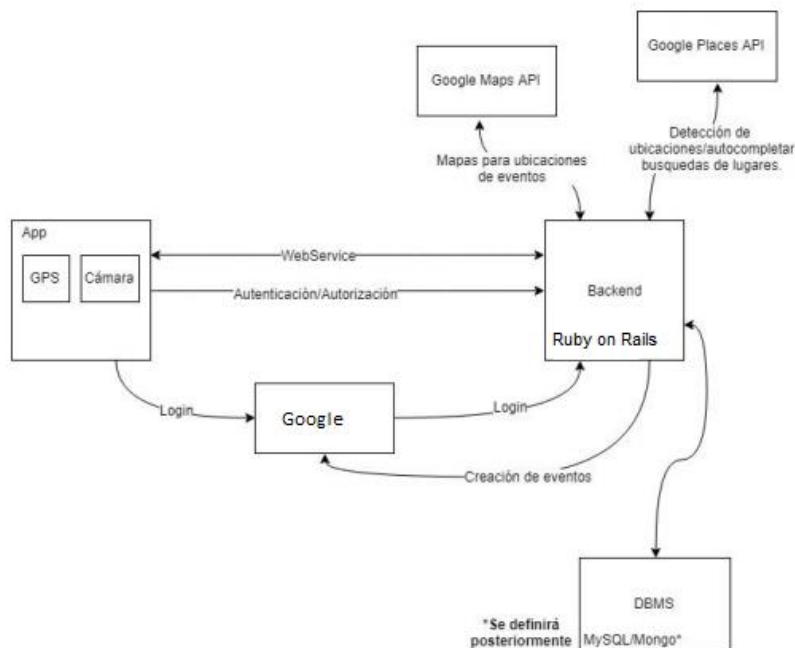
Editar

Eliminar

Detalle Subcategoría (Modal)

Descripción de diseño de alto nivel

Diagrama arquitectura



Como se

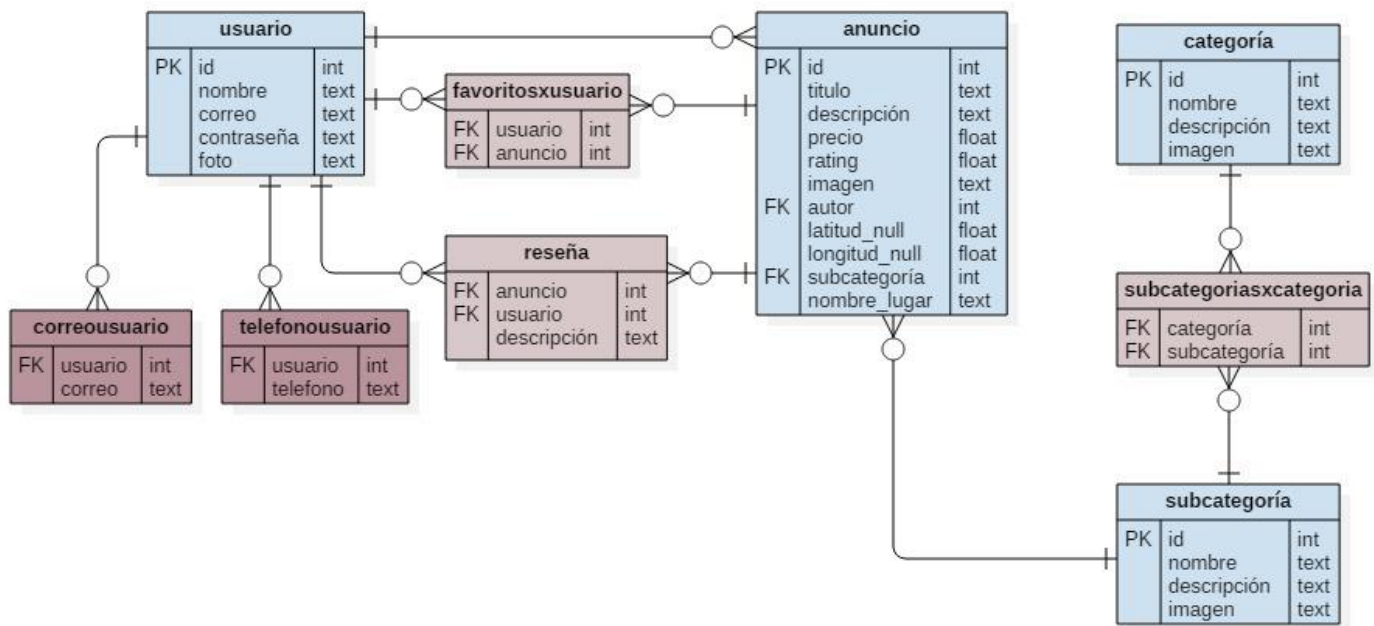
aprecia en la imagen, la aplicación consiste de un backend, el cual posee sus propias rutinas para el manejo de los datos, en la implementación final, el backend se encarga de guardar los valores de la latitud y la longitud de los lugares en donde se llevará a cabo el evento, mientras que la aplicación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

El backend fue desarrollado en Ruby on Rails, en el cual se hace el uso de una base de datos PostgreSQL, 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 Google para hacer el inicio de sesión. Para poder hacer solicitudes al backend, 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



Descripción de los web-services

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

Anuncios

get

obtiene todos los anuncios de la base de datos

/api/announcements/

get

obtiene un anuncio, dado su id

/api/announcements/id donde id es el identificador del anuncio

post

crea un nuevo anuncio en la base de datos

#

/api/announcements?title=&description=&price=&photo=&user_id=&latitude=&longitud=&subcategory_id=&place= donde id es el identificador del anuncio

delete

*# elimina un anuncio de la base de datos, dado su id
/api/announcements/id donde id es el identificador del anuncio*

*# put
edita los datos de un anuncio en la base de datos, dado su id

/api/announcements/id?title=&description=&price=&photo=&latitude=&longitude=&subcategory_id=&place= donde id es el identificador del anuncio*

*# get
obtiene una publicación random
/api/randomannouncement*

*# get
obtiene los anuncios de una subcategoría, estos son ordenados por la fecha de creación, descendentemente
/api/announcementsubcategory?subcategory_id=*

*# get
obtiene los anuncios de una subcategoría, estos son ordenados por el precio, descendentemente
/api/announcementsubcategoryP?subcategory_id=*

Autenticación

*# post
autenticación de usuario mediante su email y contraseña
/api/authenticate?email=&password=*

Categorías

*# get
obtiene todas las categorías en la base de datos
/api/categories/*

*# get
obtiene la categoría especificada por su id
/api/categories/id donde id = un id de categoría existente en la base de datos*

*# post
crea una nueva categoría, recibe como parámetro el nombre, una descripción y un enlace a una imagen
/api/categories?name=&description=&photo=*

delete

*# borra una categoría, dado su id
/api/categories/id donde id = un id de categoría existente en la base de datos*

*# put
edita una categoría, recibe como parámetro el id de la categoría y los nuevos nombre, descripción y enlace a una imagen
/api/categories/id?name=&description=&photo=*

*# get
obtiene una categoría por el nombre
/api/categorybyname?name=*

Comentarios

*# get
obtiene todos los comentarios de la base de datos
api/comments*

*# get
obtiene un comentario por medio de su identificador
api/comments/id donde id es el identificador del comentario*

*# post
crea un nuevo comentario en la base de datos
/api/comments?user_id=&announcement_id=&description=*

*# delete
borra un comentario de la base de datos
/api/comments/id donde id es el identificador del comentario*

*# update
modifica un comentario de la base de datos, dado su id
/api/comments/id?user_id=&announcement_id=&description= donde id es el identificador del comentario*

Emails

*# get
obtiene todos los emails
/api/emails/*

*# get
obtiene el email por su id
/api/emails/id donde id es el identificador del email*

*# post
crea un nuevo email para un id de usuario dado
/api/emails?user_id=&email=*

*# delete
borra un email, dado su id
/api/emails/id donde id es el identificador del email*

*# put
edita un email, dado su id
/api/emails/id?telephone= donde id es el identificador para el email*

Favoritos

*# get
obtiene todos los favoritos de la base de datos
api/favorites*

*# get
obtiene un favorito por medio de su identificador
api/favorites/id donde id es el identificador del favorito*

*# post
crea un nuevo favorito en la base de datos
/api/favorites?user_id=&announcement_id=*

*# delete
borra un favorito de la base de datos
/api/favorites/id donde id es el identificador del favorito*

*# update
modifica un favorito de la base de datos, dado su id
/api/favorites/id?user_id=&announcement_id= donde id es el identificador del favorito*

Subcategorías

*# get
obtiene todas las subcategorías en la base de datos
/api/subcategories/*

*# get
obtiene la subcategoría especificada por su id
/api/subcategories/id donde id = un id de subcategoría existente en la base de datos*

post

crea una nueva subcategoría, recibe como parámetro el nombre, una descripción y un enlace a una imagen

/api/subcategories?name=&description=&photo=

delete

borra una subcategoría, dado su id

/api/subcategories/id donde id = un id de subcategoría existente en la base de datos

put

edita una subcategoría, recibe como parámetro el id de la subcategoría y los nuevos nombre, descripción y enlace a una imagen

/api/subcategories/id?name=&description=&photo=

SubcategoriasXCategorias

get

obtiene todos los registros que guardan una subcategoría y categoría

/api/subcategoriesxcategories/

get

obtiene el registro que tiene una subcategoría y categoría, especificado por su id

/api/subcategoriesxcategories/id donde id = un id del registro existente en la base de datos

post

crea un nuevo registro que contenga una subcategoría y una categoría, recibe como parámetro el id de la subcategoría y de la categoría

api/subcategoriesxcategories?category_id=&subcategory_id=

delete

borra un registro que posea una categoría y una subcategoría, dado su id

/api/subcategoriesxcategories/id donde id = un id del registro existente en la base de datos

get

obtiene todos los registros que guardan una subcategoría y categoría

/api/subcategoriesxcategoriess/

Teléfonos

get

obtiene todos los números de teléfono

/api/telephones/

get

obtiene el número de teléfono por su id

/api/telephones/id donde id es el identificador del teléfono

post

crea un nuevo número de teléfono para un id de usuario dado

/api/telephones?user_id=&telephone=

delete

borra un número de teléfono, dado su id

/api/telephones/id donde id es el identificador del teléfono

put

edita un número telefónico, dado su id

/api/telephones/id?telephone= donde id es el identificador para el teléfono

Usuarios

get

obtiene los usuarios de la base de datos

/api/users

get

obtiene un usuario de la base de datos, dado su id

/api/users/id donde id es el identificador del usuario

post

crea un nuevo usuario en la base de datos

/api/users?name=&email=&password=&photo=&type=

delete

elimina un usuario de la base de datos, dado su id

/api/users/id donde id es el identificador del usuario

put

edita un usuario de la base de datos, dado su id

/api/users/id?name=&email=&password=&photo=&type= donde id es el identificador del usuario

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

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.

Google

Se realizó conexión con el API de Google, 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 en el backend.

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 backend. 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.