



## TAREA 1 – API Instagram

### 1. Identificar colores

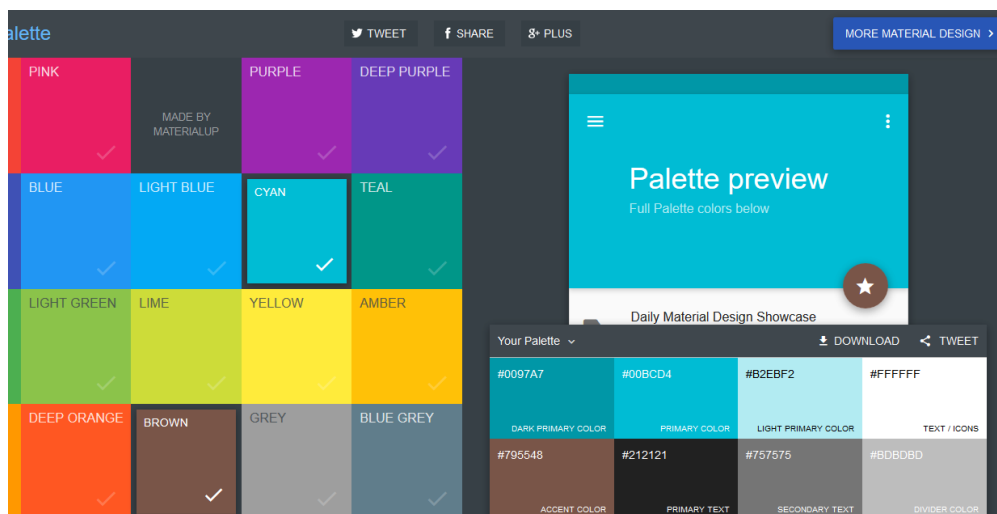
Primary Color: #00BCD4 

Dark Primary Color: #0097A7 

Accent Color: #795548 

### 2. Aplicación en Proyecto

[www.materialpalette.com](http://www.materialpalette.com)



colors.xml

```
p:\src\main\res\values\colors.xml - Android Studio 2.1.2
ild Run Tools VCS Window Help
es colors.xml
activity_main.xml x styles.xml x colors.xml x MainActivity.java x
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#00BCD4</color>
  <color name="colorPrimaryDark">#0097A7</color>
  <color name="colorAccent">#795548</color>
</resources>
```

## Descripción

El programa muestra una lista de 5 perros (recycler view), cada ítem de la lista incluye el nombre del perro, el número de likes que tiene, una foto del perro y un botón para darle like.

Una vez que le das like a un perro, el número de likes se incrementa y aparece un mensaje donde indica el nombre del perro al que le diste like.

En la parte superior hay un botón con una estrella, el cual al dar clic, te manda a la siguiente pantalla donde se ve nuevamente la lista de los perros, con la diferencia de que esta se ordena de acuerdo al número de likes de cada perro.

Para regresar a la pantalla principal, solo se presiona la flecha que aparece en la parte superior de la pantalla actual.

Se agregó un menú de opciones, con las opciones “Contacto” y “Acerca de”. La opción Contacto envía a una pantalla donde se solicitan 3 datos: nombre, correo y mensaje; se puede enviar un correo de dos formas: envío directo al correo escrito en el campo correo electrónico o mediante la aplicación de correo del teléfono.

La opción Acerca de, muestra una pantalla con la información de la persona que desarrolló la aplicación.

En la pantalla principal se agregaron dos tabs, el primer tab muestra la lista mencionada arriba y el segundo tab, muestra el perfil del perro favorito con algunas fotos así como su foto principal (Circle Image).

## Nota:

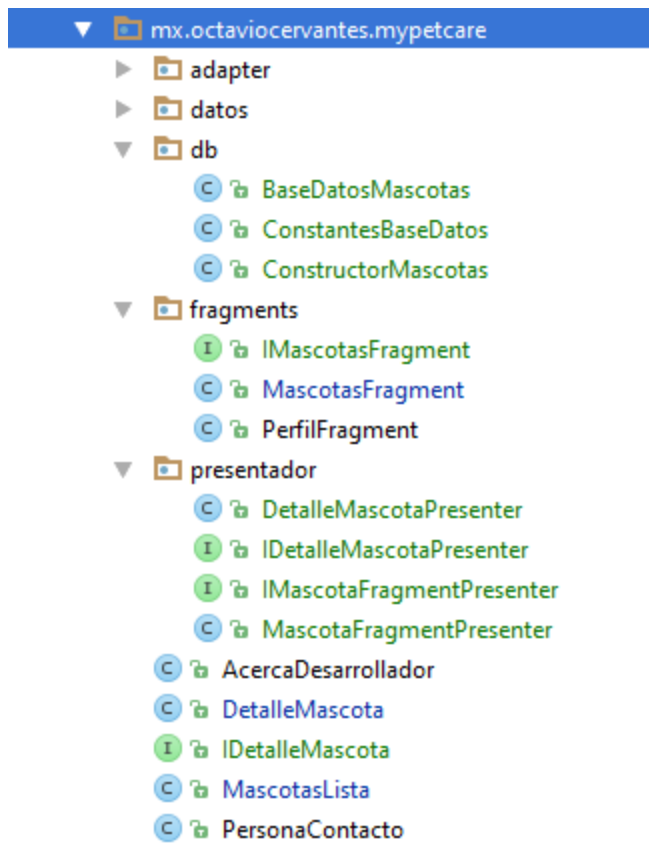
**Para el funcionamiento del envío de correo, hice la prueba en mi celular utilizando los datos, sin problema pude enviar el correo, al querer utilizar el internet de la casa, no se permitió el envío, esto fue porque estaba bloqueado el acceso al puerto 587.**

**En mi negocio también hice una prueba con el internet de ahí y sin problema pude enviar el correo.**

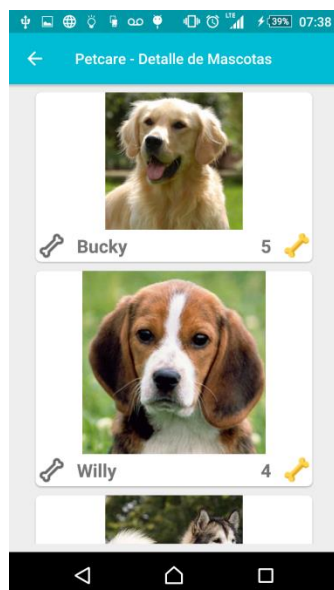
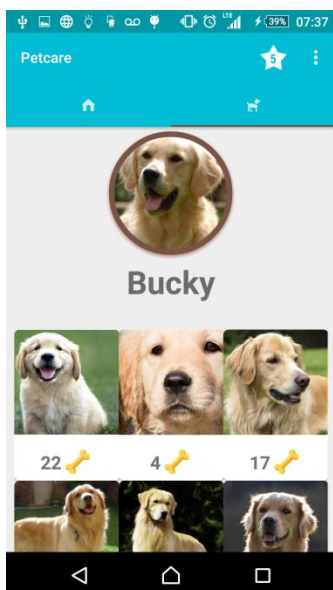
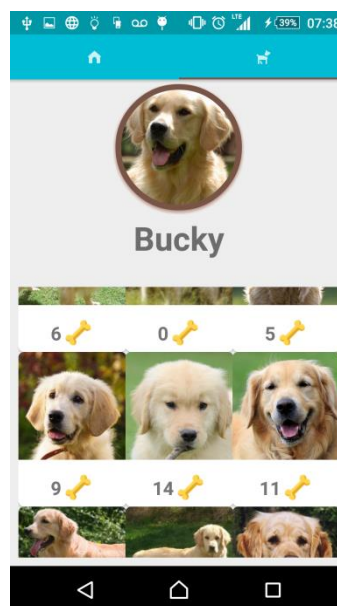
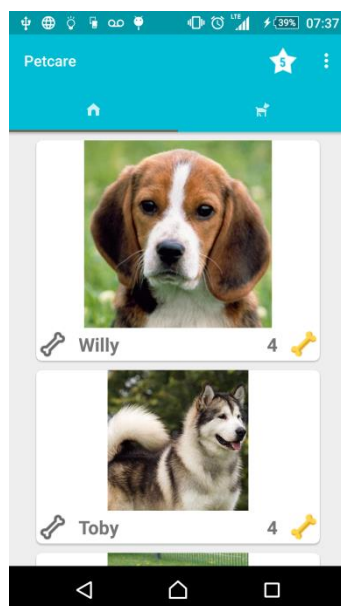
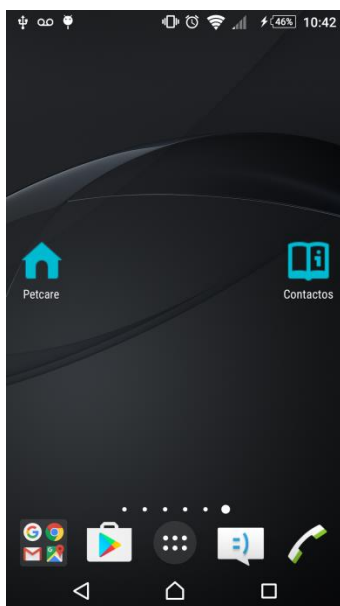
Se modificó el proyecto para generar una base de datos llamada mascotas. Esta base de datos contiene una tabla llamada DatosMascota, la cual contiene los siguientes datos:

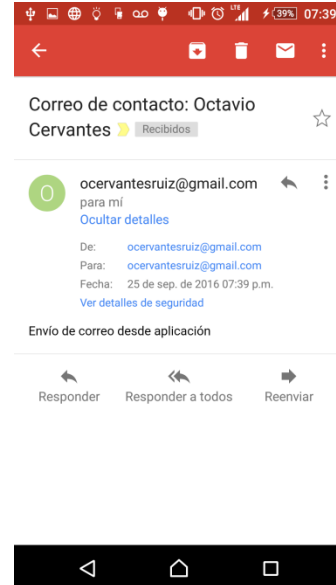
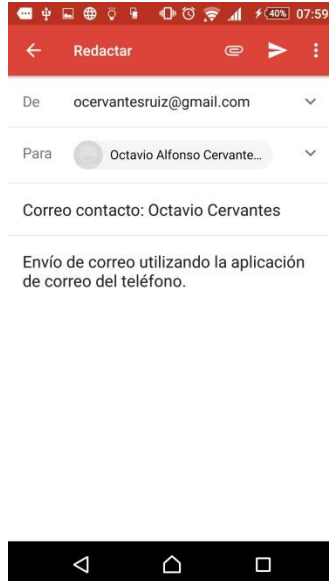
Name	Type	Schema
✓ Tables (3)		
✓ DatosMascota		CREATE TABLE DatosMascota(idMascota INTEGER PRIMARY KEY ,
idMascota	INTEGER	`idMascota` INTEGER PRIMARY KEY AUTOINCREMENT
sMascota	TEXT	`sMascota` TEXT
iFotoMasc...	INTEGER	`iFotoMascota` INTEGER
iRatingMa...	INTEGER	`iRatingMascota` INTEGER
> android_meta...		CREATE TABLE android_metadata (locale TEXT)
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
Indices (0)		
Views (0)		
Triggers (0)		

Y el proyecto quedo definido de la siguiente forma:

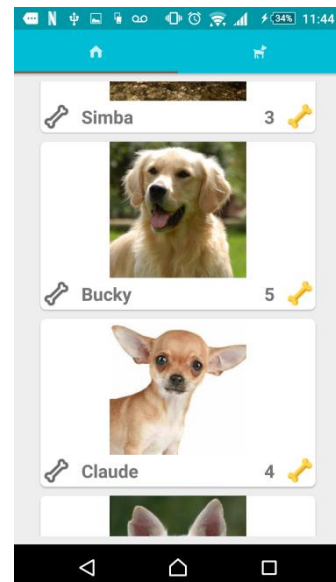
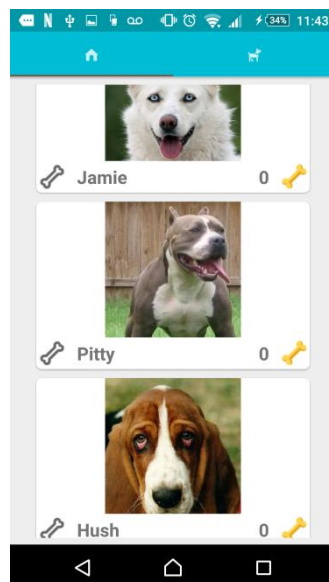
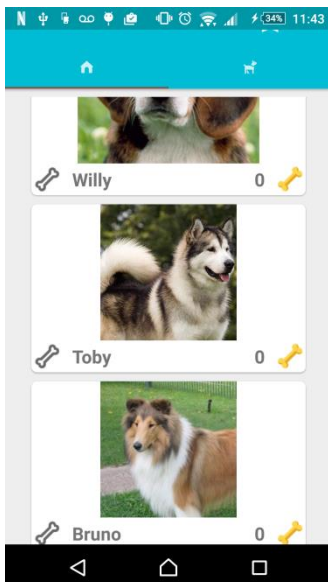


## Aplicación ejecutada en celular con Android 5.1 (API 22)

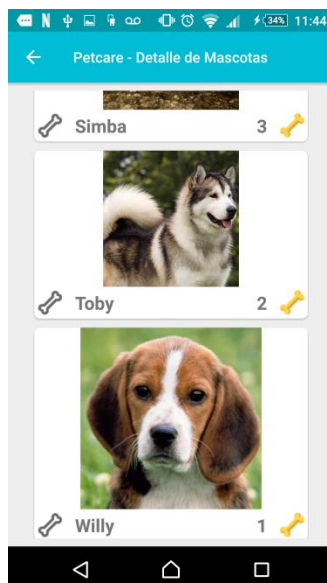
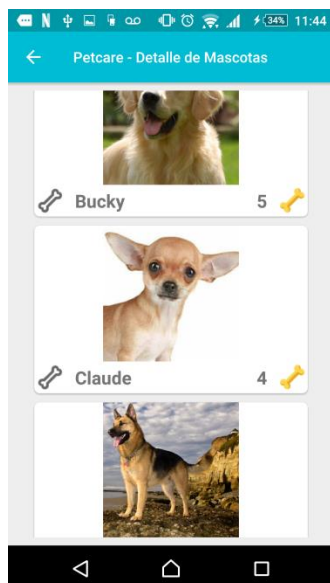




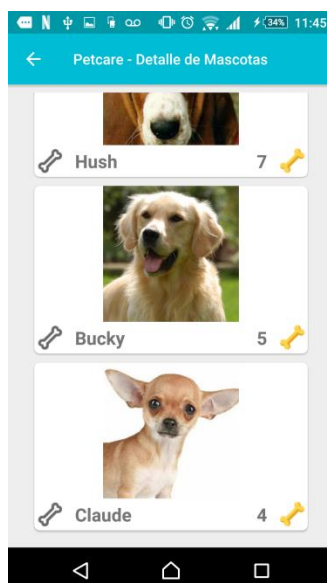
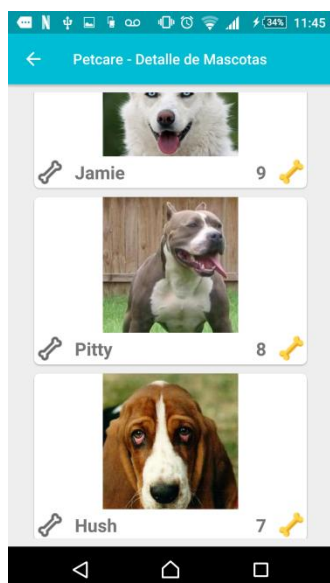
Se anexan pantallas del funcionamiento de la aplicación con base de datos:



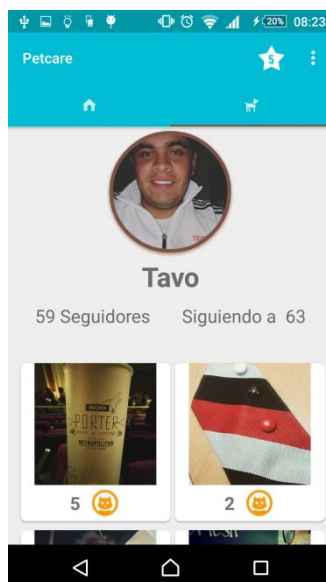
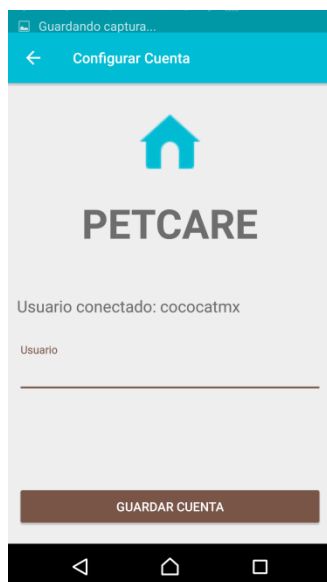
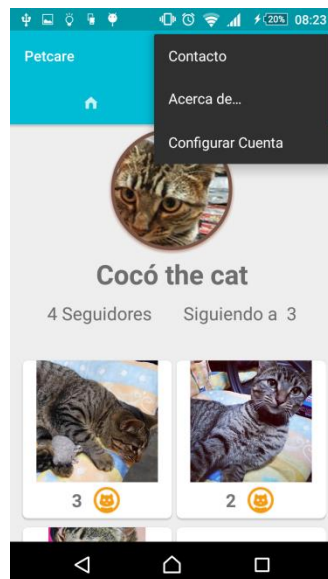
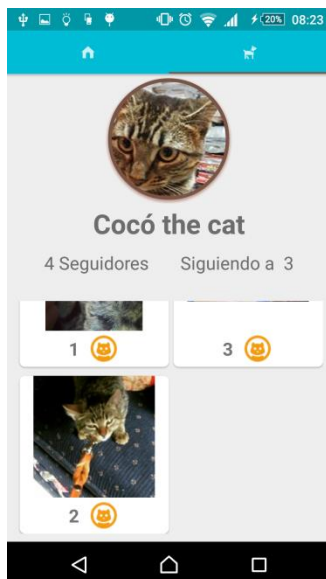
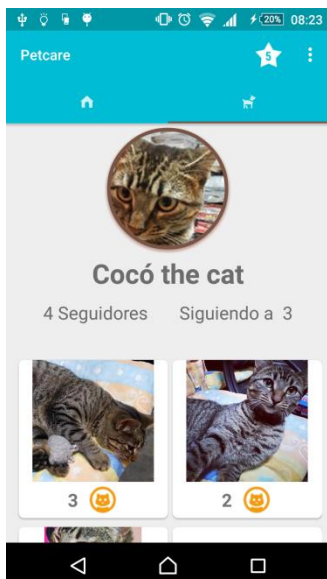
Primera lista ordenada por likes:



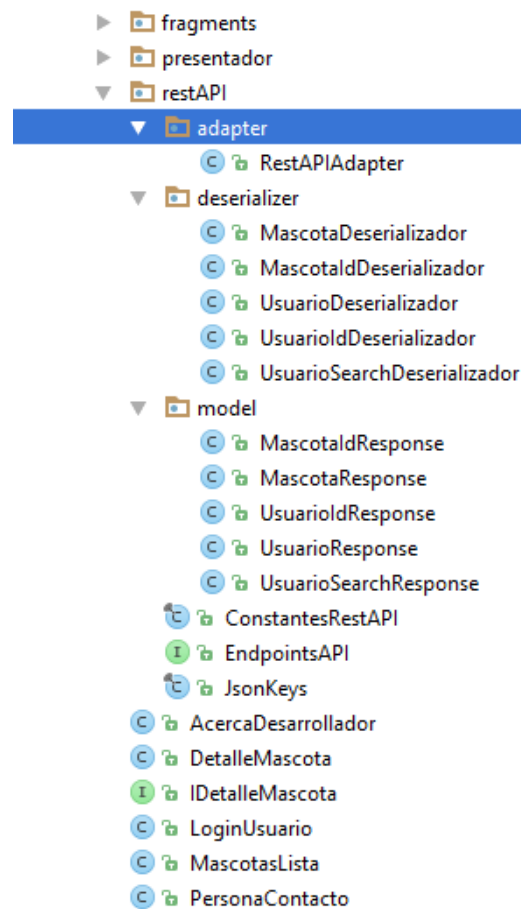
Lista actualizada por los likes:



## API Instagram



## Estructura del proyecto





## Endpoints utilizados

**GET** /users/self

https://api.instagram.com/v1/users/self/?access\_token=ACCESS-TOKEN

RESPONSE ▾

Get information about the owner of the access\_token.

PARAMETERS

**ACCESS\_TOKEN** A valid access token.

**GET** /users/**user-id**

https://api.instagram.com/v1/users/{user-id}/?access\_token=ACCESS-TOKEN

RESPONSE ▾

Get information about a user.  
The public\_content scope is required if the user is not the owner of the access\_token.

REQUIREMENTS

Scope: public\_content

PARAMETERS

**ACCESS\_TOKEN** A valid access token.

**GET** /users/self/media/recent

https://api.instagram.com/v1/users/self/media/recent/?access\_token=ACCESS-TOKEN

RESPONSE ▾

Get the most recent media published by the owner of the access\_token.

PARAMETERS

**ACCESS\_TOKEN** A valid access token.

**COUNT** Count of media to return.

**MIN\_ID** Return media later than this min\_id.

**MAX\_ID** Return media earlier than this max\_id.

GET

/users/**user-id**/media/recent

https://api.instagram.com/v1/users/{user-id}/media/recent/?access\_token=ACCESS-TOKEN

RESPONSE ▾

Get the most recent media published by a user.

The public\_content scope is required if the user is not the owner of the access\_token.

REQUIREMENTS

Scope: public\_content

PARAMETERS

ACCESS_TOKEN	A valid access token.
COUNT	Count of media to return.
MIN_ID	Return media later than this min_id.
MAX_ID	Return media earlier than this max_id.

GET

/users/search

https://api.instagram.com/v1/users/search?q=jack&access\_token=ACCESS-TOKEN

RESPONSE ▾

Get a list of users matching the query.

REQUIREMENTS

Scope: public\_content

PARAMETERS

ACCESS_TOKEN	A valid access token.
Q	A query string.
COUNT	Number of users to return.

## Iconos e imágenes usadas en la aplicación.

