
P3 DS

Marina Hernández Bautista
Román Larrosa Lewandowska

Scorelt

mayo de 2020

1. Fase de Diseño

Empezamos la memoria justo después de realizar la fase de análisis, donde concretamos en un documento los requisitos e inquietudes a abordar.

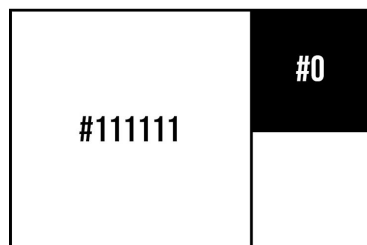
Consideramos que el primer paso en nuestra fase de diseño es establecer unos bocetos de diseño adecuados para hacernos una idea de cómo queremos que luzca la aplicación y poder basar la interfaz en estos bocetos.

En primer lugar, realizamos un Style Guide con los colores primario y secundario de la aplicación y los estilos de letra a utilizar:

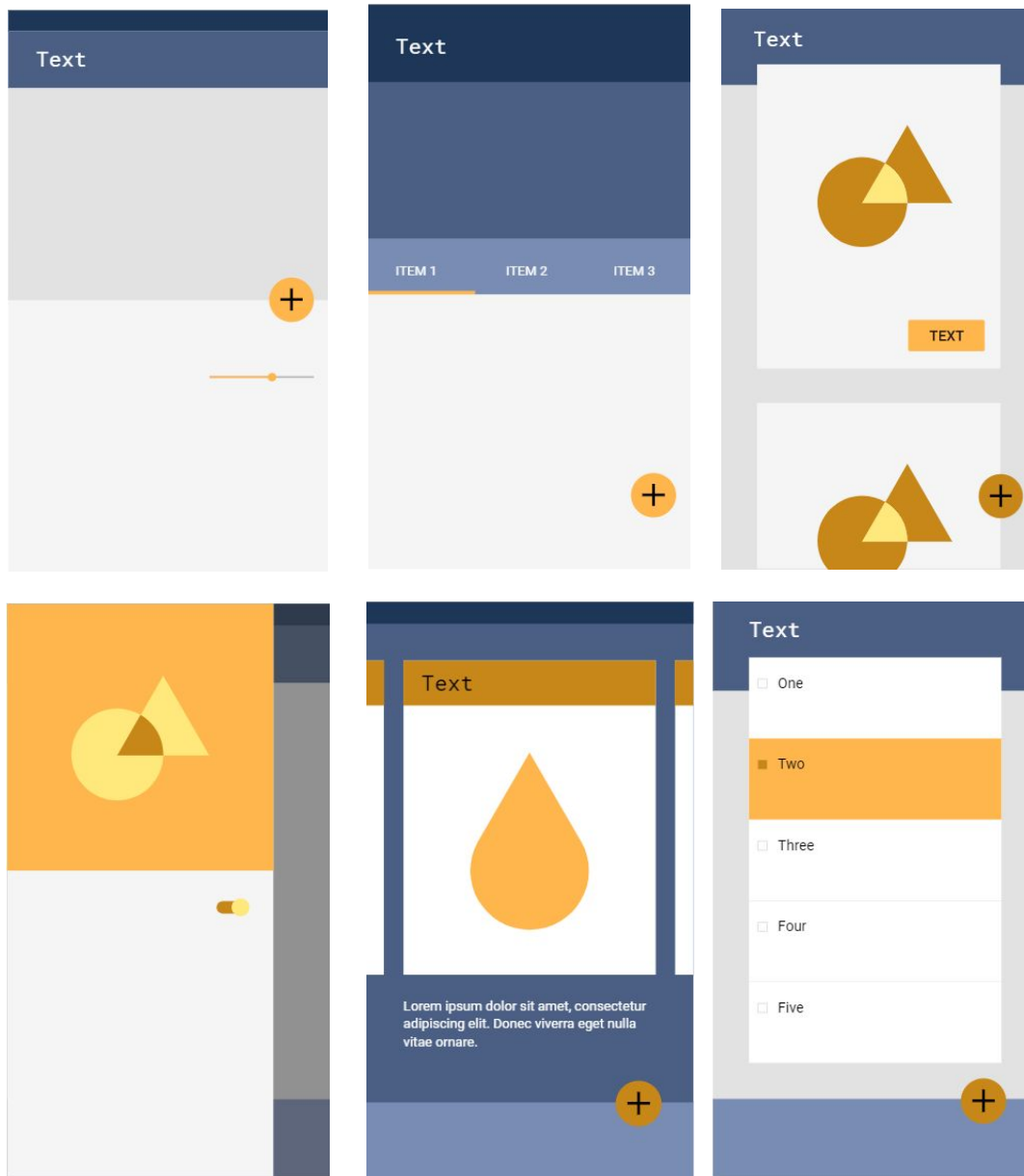
Roboto

(BODY & CAPTIONS)

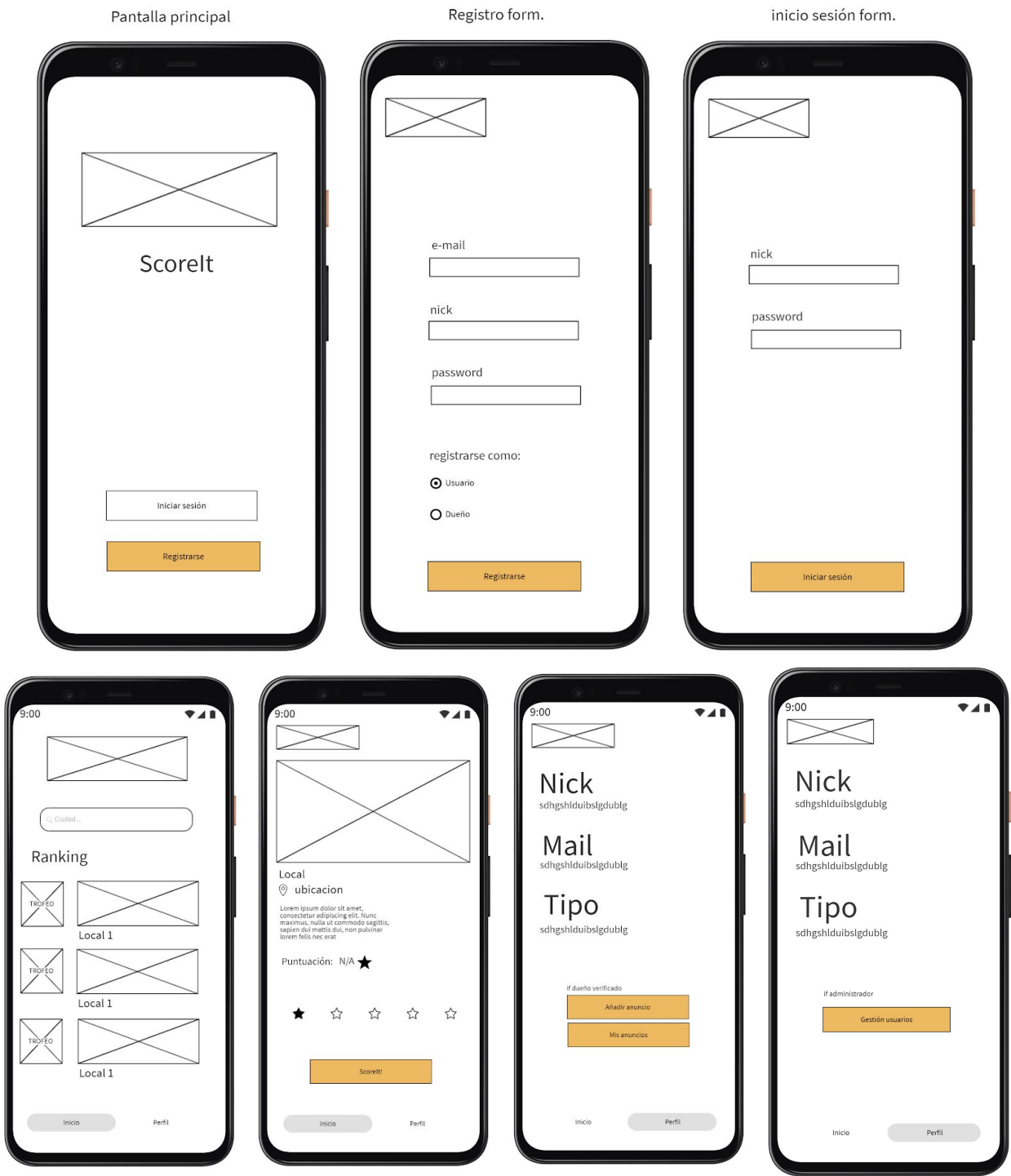
Roboto Normal

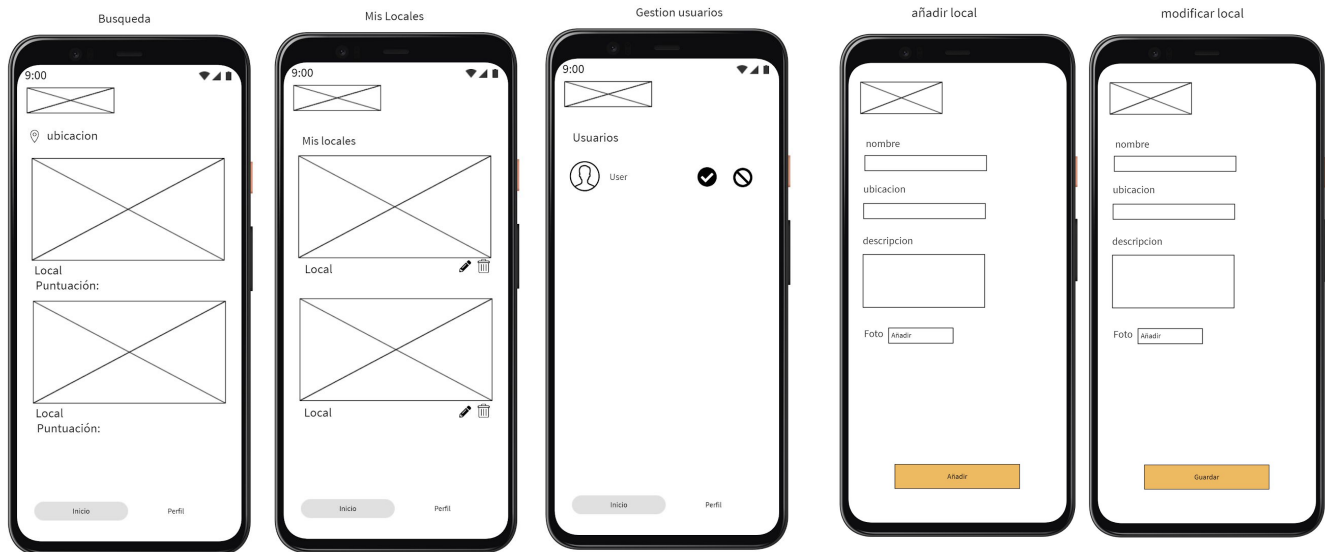


Aquí mostramos un ejemplo de cómo quedarían esos colores sobre una aplicación Android, utilizamos la herramienta ColorTool, de las guías de estilo de Google:



Después, elaboramos los bocetos para las distintas pantallas y procesos de Scorelt:





Por último, determinamos un logo para la marca Scorelt, que aparecerá normalmente en la esquina de la app, y un icono para la aplicación.

Las referencias que utilizamos son: <https://www.flaticon.com/authors/freepik>



Scorelt!

2. Fase de Codificación

Una vez definido el diseño de Scorelt comenzamos con la codificación de la app que servirá de vínculo con los usuarios. Vamos a realizar una aplicación android, y para ello usaremos el **framework Android Studio** y el lenguaje de programación **Java**.

El primer paso a realizar será programar la interfaz de las distintas pantallas, configurando la navegación. Empezamos por la barra de navegación, que será común para la mayor parte de las pantallas. El MainActivity.xml contendrá un elemento BottomNavigationBar, que contendrá en su interior diversas actividades, uno para la página de inicio y otro para el perfil.

Una vez definidas el resto de actividades y sus posibles fragmentos (interfaces fijas), pasamos a programar la funcionalidad. Empezamos realizando las distintas funciones **php** en distintos archivos: login.php, registro.php, conection.php y creamos, a la par, la base de datos que utilizaremos y sus correspondientes tablas de usuarios y locales.

Las conexiones entre Android Studio y el servicio web creado con php las establecemos con la **librería Volley** de Android Studio y también como conexiones **HttpConnection** que implementa Java. Volley es una librería que nos permite hacer peticiones y obtener respuestas de php en formato cadena de caracteres o JSON. Utilizaremos las salidas en forma de cadena de caracteres para manejar los errores y los objetos JSON para luego formatearlos y convertirlos a las clases que manejen los usuarios o los locales de ocio.

Puesto que tenemos funcionalidades diferentes según el tipo de usuario, lo primero que hacemos es poner a funcionar las actividades de registro y login y, una vez conseguido esto, adaptamos la actividad de perfil para mostrar una u otra funcionalidad según se trate de un usuario registrado o no, tal y como mostramos en los bocetos.

En el primer borrador se incluye la funcionalidad que pueden realizar los usuarios de tipo administrador y usuario registrado y la navegación entre las principales pestañas de ScoreIt. Entre ellas, hemos implementado funciones como:

- Registrar un usuario
- Login de un usuario
- Cerrar sesión
- Verificar Usuario
- Eliminar usuario

3. Fase de Codificación (v2)

Continuamos con la fase de codificación, implementando las funcionalidades asociadas a los locales de ocio. La funcionalidad de los locales se basa inserciones, actualizaciones y consultas en dos tablas dentro de nuestra BD.

La tabla LOCALES controla las variables id, nombre, ubicación, descripción y dueño del local, y la tabla PUNTUACIONES controla la puntuación que cada usuario tiene de cada local. La clave primaria de esta tabla es la unión de un id_local y un id_usuario, así prohibimos a un mismo usuario insertar puntuaciones una tras otra en el mismo local.

De esta manera implementamos funcionalidades como:

- Buscar un lugar de ocio
- Puntuar un lugar de ocio
- Ver locales de ocio de un propietario
- Añadir un local de ocio (solo propietarios verificados)
- Ranking de los locales de ocio mejor puntuados
- Recargar Ranking

Algunas dificultades que han ralentizado el desarrollo de Scorelt son:

-Aprender nuevas tecnologías para realizar peticiones tipo REST a un servicio externo. Hemos utilizado dos tecnologías nuevas y diferentes: la librería Volley y la clase HttpURLConnection.

-Familiarizarnos con el entorno de desarrollo Android Studio.

4. Diagrama de Clases Final

