

Cheer App!

A series of five parallel white lines of varying lengths, slanted diagonally from the top right towards the bottom left, creating a sense of motion or speed.

TFG DAM - MEMORIA

Por Kevin Castillo Escudero

ÍNDICE

1. Agradecimientos	3
2. Resumen.....	4
3. Abstract (resumen en inglés)	5
4. Introducción	6
5. Alcance funcional del sistema	7
a. Sistema de autenticación de usuarios:.....	7
b. Sistema de almacenamiento para elementos variados:	8
c. Sistema de ubicación de cervezas:	9
6. Diseño técnico	10
a. Estructura general:	10
b. Android Manifest:	11
c. Tema de la app:	11
d. Estructura del directorio Java :	12
e. Estructura del directorio Res :.....	23
f. Diagramas de clases	24
g. Diagrama de BDD	24
7. Tecnologías utilizadas.....	25
a. Android Studio.....	25
b. Google Firebase	25
c. Google Maps	25
d. GitHub	25
8. Código documentado	26
9. Presupuesto.....	27
10. Planificación	28
11. Manual de instalación	29
a. Requisitos de hardware y software.....	29
b. Instalación	29
12. Manual de usuario.....	31
13. Análisis de competencia.....	32
14. Conclusiones.....	33
a. Mejora de Firebase:	33
b. Mejora de la app de Android:	34
15. Bibliografía:	35

1. Agradecimientos

- [Material Design Icons](#) por ofrecer una amplia gama de iconos útiles para el proyecto.
- [Flaticon](#) por ofrecer imágenes que pueden ser usadas como icono de launcher y otros recursos.
- [Miguel Catalán Bañúls](#) por su [librería MaterialSearchView](#).
- **Al equipo de profesores del C.F.P. Juan XXIII de Alcorcón** por ayudarme a llevar a cabo este proyecto, y en especial al profesor **Fernando Paniagua Martín** por su ayuda e ideas.

2. Resumen

El proyecto trata sobre una **aplicación para smartphones creada con Android Studio, orientada a dispositivos con API igual al 21 o superior** (Android Lollipop, versión 5.0). Se trabajará, además, **con otras tecnologías pertenecientes al API de Google, las cuales son Firebase (manejo de base de datos y autenticación de usuarios) y Google Maps (servicios de ubicación en mapa digital).**

Dicha aplicación **permitirá a sus usuarios realizar numerosas acciones**, entre las cuales, **darse de alta en la aplicación**, mantener un **registro personalizado y único de sus cervezas favoritas**, las cuales pueden encontrarse **almacenadas en la BDD de Firebase** y son **localizables gracias al servicio de búsqueda**, **ver los detalles** tanto de las cervezas favoritas como las que no, **guardar o borrar favoritos**, **compartir las cervezas**, e incluso **ubicarlas**, y más acciones.

La idea no está totalmente desarrollada, puesto **que faltan muchos componentes por añadir al proyecto**, pero **dichos componentes serán desarrollados posteriormente** puesto que **yo, el alumno, preveo continuar con este proyecto de manera personal y publicarlo comercialmente en Google Play en un futuro**, y el alcance que tendría la aplicación iría mucho más allá de lo necesario. **Dicho alcance se explicará más adelante en este documento.**

Anotación 1. En la estructura del proyecto se puede encontrar una carpeta llamada “Cheer App”. La explicación a ésta es simple: en un principio el proyecto se llamaba **“The Beer App”**, pero es un nombre que se declinó y fue sustituido por **“Cheer App”**. El problema vino al **renombrar los paquetes y otros componentes**, ya que Android Studio no lo ponía fácil, así que **decidí cambiar los nombres de esos por los originales y sólo cambiar el nombre de la app a la vista del usuario.**

Anotación 2. El nombre proviene de un juego de palabras: en inglés, **cuando se brinda se dice “Cheers!”**, que significa lo mismo que **“Salud!”**, y ese significado se mezcla con el del verbo preposicional **“cheer up”**, que significa **animar o alegrarse.**

3. Abstract (resumen en inglés)

The project is about **an app for Android smartphones, developed with Android Studio, focused on devices with and API equal to version 21 or higher** (Android Lollipop, version 5.0). Other **Google API technologies will be used**, which are **Firestore (database management and user authentication)** and **Google Maps (digital map location services)**.

Such application **will allow users to do several actions**, such as **signing up in the services of the app**, keep a **personal and unique record of their favourite ales** (which can be found stored in **Firestore DB** and are **reachable thanks to the search engine**), **view details** of both favourite and not favourite ales, **save or delete their personal favourites**, **share the ales**, and even **locate them**, and so on.

The idea is not fully developed, since **there are lots of components yet to be added to the project**, but **those components will be developed later**, as me, the student, **I intend to continue with this project on my own and commercially publish it on Google Play in the future**, and the scope of this app would go way further than necessary for now. **The said scope will be explained further in this document.**

4. Introducción

Este proyecto ha sido **realizado por Kevin Castillo Escudero** para el **módulo del Trabajo de Fin de Grado** perteneciente al **Ciclo de Grado Superior de Desarrollo de Aplicaciones Multiplataforma**.

El desarrollo del proyecto ha sido **basado en una idea surgida entre el alumno y el profesor Fernando Paniagua**.

Como todo proyecto de Android, las distintas ventanas de la app se llaman Activities, y en cada una de ellas se trabaja una funcionalidad distinta. Los apartados que contiene esta aplicación son los siguientes:

- **Pantalla de saludo de la app (SplashScreenActivity):** al abrirse la app muestra una pantalla de saludo durante 2 segundos, para luego redirigir a la Activity correspondiente en función de si ya está registrado o no.
- **Inicio de Sesión/Registro de Usuarios (MainActivity):** los usuarios deben registrarse por primera vez o iniciar sesión con sus credenciales en la app para poder acceder al resto de funcionalidades de la app. Este sistema de usuarios se realiza gracias a Firebase y sus métodos de autenticación de usuarios.
- **Inicio de la app (HomeActivity):** se muestran distintos elementos para distintas funcionalidades.
 - **Lista de favoritos:** la lista de las cervezas favoritas del usuario. Cada ítem de la lista contará con un menú contextual que se mostrará al ejecutar una pulsación larga sobre ellos.
 - **Menú superior:** menú de la barra superior, que contiene las acciones de generación de nuevas cervezas y búsqueda de las ya existentes.
 - **Menú lateral desplegable de navegación:** similar al de otras apps, como Gmail, un menú que contiene distintas opciones de administración de cuenta y miscelánea.
- **Generar nuevas cervezas (NuevaCervezaActivity):** aquí el usuario podrá sugerir nuevas cervezas que se añadirán a la BDD para que estén accesibles a otros usuarios y a sí mismos.
- **Detalle de la cerveza (DetalleCervezaActivity):** se muestran distintos campos pertenecientes a la cerveza sobre la que desea conocer más información, y controles para gestionar favoritos y ubicar la cerveza.
- **Ubicación de la cerveza (MapsActivity):** el usuario podrá ubicar los distintos puntos dónde disfrutar de la cerveza.

Todos estos apartados se verán en detalle más adelante con imágenes ilustrativas.

5. Alcance funcional del sistema

La aplicación cubre distintos ámbitos con los que el usuario interactuará, que son los siguientes:

a. Sistema de autenticación de usuarios:

Para poder acceder a los distintos aspectos de la aplicación es necesario que el usuario se autentique en el Sistema de Autenticación de Firebase. Para ello existen 2 opciones:

- i. *Servicio de registro*: el usuario **se dará de alta** en el servicio en caso de no haberlo hecho previamente, **rellenando los distintos campos que figuran en la pantalla de Registro**. Todos ellos son **campos obligatorios de cara a la inscripción**, pero **debido a distintos motivos**. El Sistema de Autenticación de Firebase registrará a los usuarios con su e-mail y contraseña, y el resto de campos se almacenarán en la BDD de Firebase para el resto de acciones y otros aspectos. **Al registrarse en la aplicación, será redirigido al menú principal de la aplicación** sin necesidad de iniciar sesión manualmente. El Sistema de Autenticación de Firebase **distingue intrínsecamente si el usuario ya se ha registrado previamente** con el mismo e-mail para así **evitar duplicaciones en cuanto a credenciales de usuario** (sólo el e-mail, por supuesto, ya que la misma contraseña puede ser usada por distintos usuarios como en cualquier otro sistema de autenticación).
- ii. *Servicio de inicio de sesión*: **el usuario iniciará sesión con sus credenciales**, en caso de que use el servicio en otro dispositivo o de que haya cerrado sesión manualmente, por ejemplo. Para ello, **usará su e-mail y contraseña en los campos de la pantalla de Login**, ya que son las credenciales que se almacenaron en el **Sistema de Autenticación de Firebase**. Una vez autenticado, **será redirigido al menú principal de la aplicación**.

b. Sistema de almacenamiento para elementos variados:

La aplicación almacenará datos de variada naturaleza concernientes al usuario mientras éste use la aplicación. La estructura de estos datos se puede analizar en la captura del modelo de datos adjuntada posteriormente en este documento en el apartado de Diseño Técnico. Dichos datos serán los siguientes:

- i. *Usuarios registrados en la aplicación*: los usuarios que sean autenticados en el servicio de la aplicación también serán registrados en la BDD para almacenar otros datos además de sus credenciales. Estos datos serán el nick de usuario y la edad, y se almacenan para ser usados en la aplicación como, por ejemplo, en la cabecera del Navigation Drawer del menú principal de la aplicación.
- ii. *Cervezas almacenadas (previo uso de los usuarios y creadas por los mismos)*: existen cervezas en la BDD creadas por mí, el desarrollador, para que el usuario pueda buscar cervezas sin tener que crear las suyas propias, de tal manera que resulte fácil e intuitivo usar la app desde el primer uso. Por supuesto, si otros usuarios han registrado alguna cerveza nueva, los usuarios que usen la app posteriormente podrán acceder a las cervezas por defecto y a las nuevas, las creadas por los usuarios. Funcionará así para esta versión de la aplicación, pero en versiones futuras las cervezas añadidas por los usuarios constarán como “sugerencias de cervezas” hasta que sean moderadas y acreditadas.
- iii. *Ubicaciones de las cervezas*: en la BDD de Firebase están también almacenadas las ubicaciones de los puntos de venta de las cervezas que los usuarios usarán para saber dónde localizarlas y adquirirlas en los puestos de venta. Cabe destacar que, para esta versión del proyecto, y tal y como se menciona en otros puntos de este documento, todas las cervezas comparten por defecto las mismas ubicaciones, ya que las ubicaciones figuran como un nodo (una tabla) independiente de las cervezas, pero en versiones futuras cada cerveza tendrá registradas sus propias ubicaciones. Se ha trabajado así porque se ha considerado que para la versión de prueba de la app es más útil, pero será modificado posteriormente.
- iv. *Cervezas favoritas de los usuarios*: cada usuario tendrá almacenadas sus cervezas favoritas personales para poder acceder a ellas en cualquier momento. Las favoritas de un usuario solo pueden ser modificadas por ese mismo usuario, ya que no son accesibles por ningún otro usuario de la aplicación.

- v. *Sugerencias y reportes*: los usuarios pueden enviar reportes de incidencias y sugerencias de desarrollo que serán almacenados en la BDD de Firebase para poder ser tenidos en cuenta posteriormente. Ambas funcionalidades tienen una estructura similar pero no la comparten, ya que son almacenados en distintos nodos, pero ambos se almacenan con un id, y sus datos internos son el texto (de sugerencia o reporte, según corresponda), y el ID del usuario que lo agregó a la BDD.

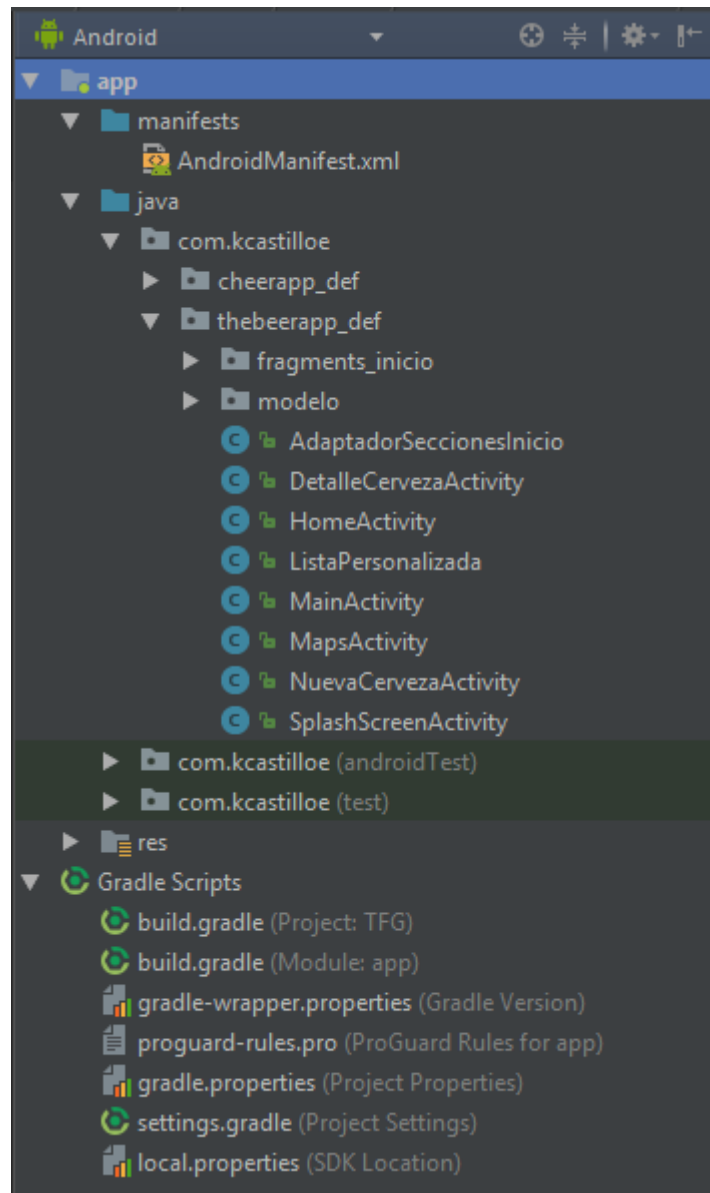
c. Sistema de ubicación de cervezas:

La aplicación usa un sistema de localización de cervezas en tiempo real implementado en la aplicación. Para ello se usa el API de Google Maps, y funciona de tal manera que el usuario selecciona una cerveza a localizar, se recogen las ubicaciones para esa cerveza y las carga en la aplicación para que el usuario pueda situarlas visualmente en un mapa virtual, y si se clica en un marcador, se muestra el título de la ubicación y la posibilidad de navegar hasta la ubicación (esta opción redirige a la aplicación Google Maps instalada en el dispositivo).

6. Diseño técnico

El proyecto de Android Studio se ha estructurado de la siguiente manera:

a. Estructura general:



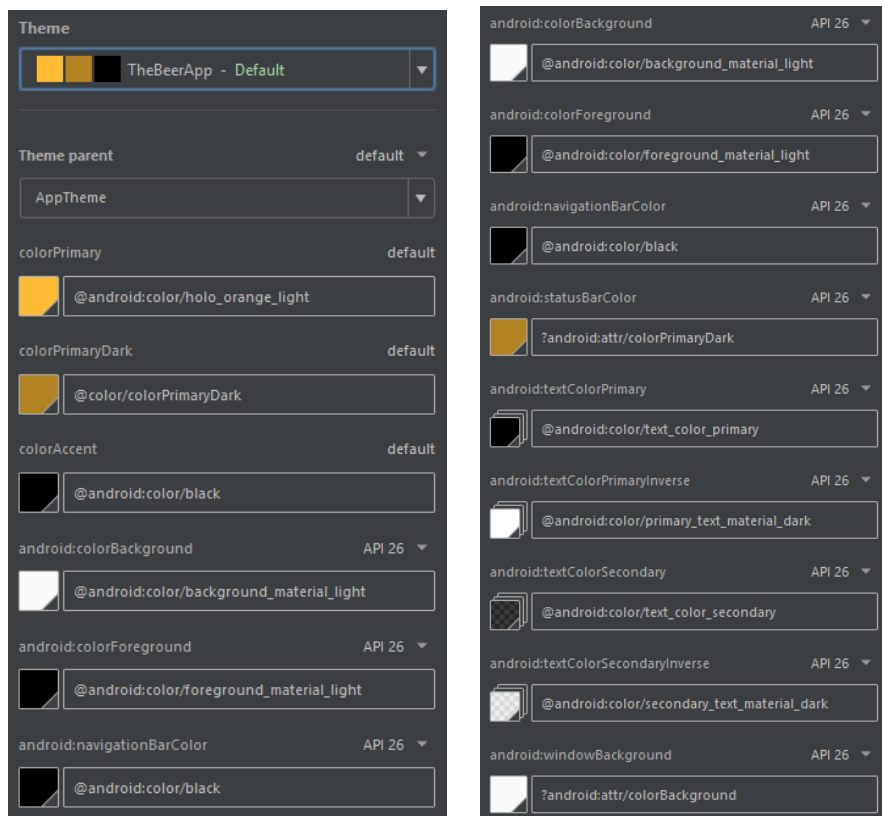
b. Android Manifest:

En este archivo se definen distintos parámetros, a saber:

- **Permisos:** los permisos que se solicitarán al usuario para poder usar la aplicación. Para este proyecto figuran los permisos de **conexión a redes móviles y Wi-Fi, conexión a Internet y Maps (navegación y ubicación)**.
- **Parámetros de aplicación:** permite **backup (copia de seguridad)**, tiene **icono de aplicación y título, tema de la app (explicado más adelante) y metadatos**, tales como **claves de API** y demás.
- **Definición de las distintas Activities que componen la aplicación.**

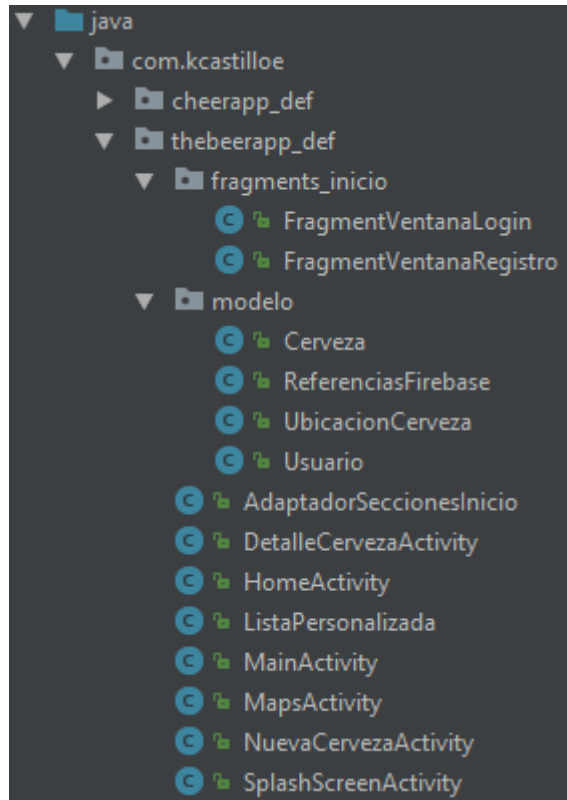
c. Tema de la app:

A continuación, se puede observar la configuración del tema de la app.



d. Estructura del directorio Java:

Aquí se almacenarán **las distintas clases creadas para la aplicación** y el propósito y uso de cada uno. Debido a que sería un **contenido demasiado difícil de explicar en este documento**, solo se verá brevemente cada uno de ellos, **y se pone en disposición la documentación generada para el proyecto, así como el código, para poder entender su funcionamiento.**



i. Directorio “fragments_inicio”:

Aquí se almacena el código Java para los **Fragment** que se mostrarán como **ventanas parejas en la MainActivity**, y que como se puede apreciar por su nombre, **contienen los formularios de login y registro**.

ii. Directorio “modelo”:

Contiene las clases cuyo código **sirve de modelo para distintos aspectos de la aplicación**. **Cerveza, UbicacionCerveza y Usuario** se corresponden con **objetos de la BDD de Firebase** y por tanto contienen un **modelo de datos igual al de Firebase**, mientras que **ReferenciasFirebase** guarda **referencias a distintos nodos de la BDD** como constantes propias **para poder acceder a los mismos fácilmente**.

iii. AdaptadorSeccionesInicio:

En esta clase se define un **adaptador personalizado para los Fragment de la MainActivity** previamente mencionados (**login y registro**), y sus métodos para manejarlos.

iv. ListaPersonalizada:

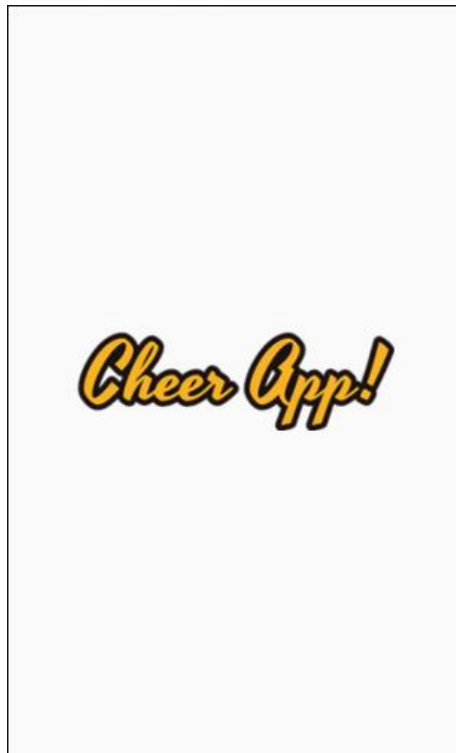
Contiene el código para generar una **lista con modelo personalizado que servirá para mostrar las cervezas favoritas del usuario**, con una **disposición personalizada para los ítems**.

A continuación, en orden de acceso a las mismas en la aplicación, se muestran las distintas Activities que componen la aplicación.

v. *SplashScreenActivity*:

Activity que se muestra siempre al abrir la aplicación, y que presenta durante 2 segundos logo de la app a modo de saludo antes de pasar a la siguiente Activity. Dependiendo de si figura como que ha iniciado sesión o no pasará a una Activity o a otra:

1. Si no ha iniciado sesión, muestra la MainActivity para que el usuario se registre o inicie sesión.
2. Si ha iniciado sesión, pasará a HomeActivity directamente y cargará los datos del usuario.



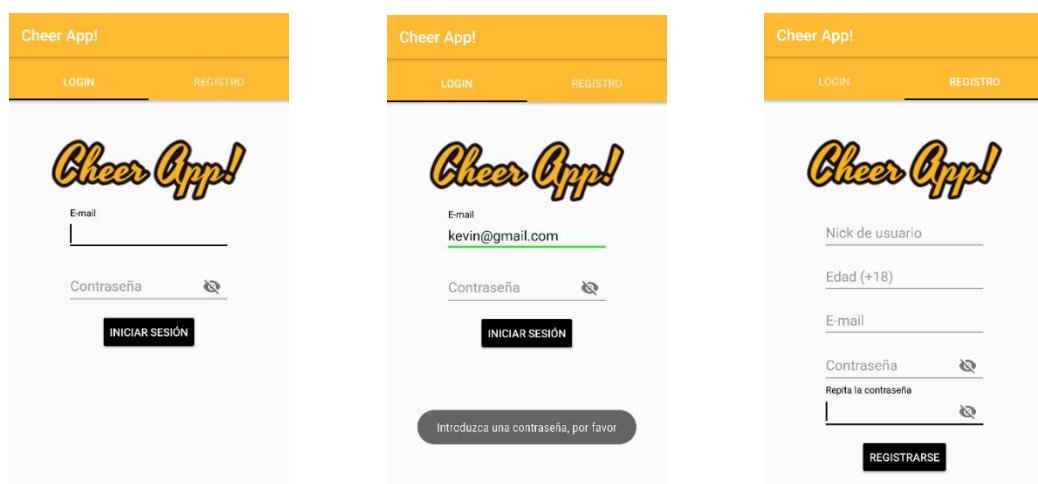
vi. *MainActivity*:

Contiene los **formularios de inicio de sesión y registro de usuarios**. Los **campos de ambos formularios cuentan con controles de validación** dependiendo del contenido que les corresponda (en esta guía se encontrarán entre paréntesis a continuación de cada campo), **que de no cumplirse se notificarán al usuario con un mensaje por pantalla**.

1. **Inicio de sesión (Login)**: cuenta con 2 campos, para introducir las credenciales, que son el **correo electrónico (formato de correo electrónico)** y la **contraseña (8 caracteres o más; mínimo 1 minúscula, 1 mayúscula y 1 número; sólo caracteres alfanuméricos)**. Si dichas credenciales no coincidiesen con ninguna de las almacenadas en el sistema de autenticación de Firebase, se notificará al usuario.
2. **Registro**: cuenta con 5 campos, que son el **nick de usuario (de 5 a 15 caracteres, solo alfanuméricos y barra baja ["_"])**, la **edad (mayor de 18 y menor de 130, por poner un límite superior)**, el **correo electrónico (formato de correo electrónico)**, y la **contraseña y verificación de contraseña (8 caracteres o más; mínimo 1 minúscula, 1 mayúscula y 1 número; sólo caracteres alfanuméricos)**. Una vez se complete con éxito, dará de alta al usuario en el sistema de Autenticación de Firebase para este proyecto con su correo electrónico y contraseña, y lo almacenará en la BDD con su ID y los otros campos (nick, edad).

Una vez que se haga la **operación pertinente de manera correcta**, **enviará al usuario a la HomeActivity**.

Izquierda a derecha: ejemplos de ventana de login, error en campo, y ventana de registro.



Ejemplo de usuario registrado y almacenado.

```
usuarios
├── gVYtmjWti7MQ7nFRUx4rAkRgYDz1
└── noVgIY38ECMPgOotVRVnRw3hpCO2
    ├── edad: 18
    ├── email: "pepito@gmail.com"
    └── nick: "Pepito"
```

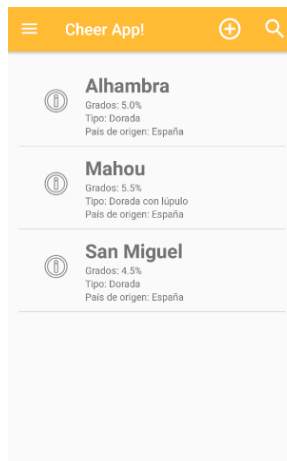

vii. *HomeActivity:*

En esta Activity el usuario contará con varios elementos que cuentan con sus propias funcionalidades. La lista de cervezas favoritas se mostrará en la pantalla una vez que la Activity haya cargado, y puede contar con 2 estados: vacía o con elementos. Para cada una de ellas contará con un modelo distinto.

1. **Vacía:** mostrará un mensaje al usuario indicando que está vacía.

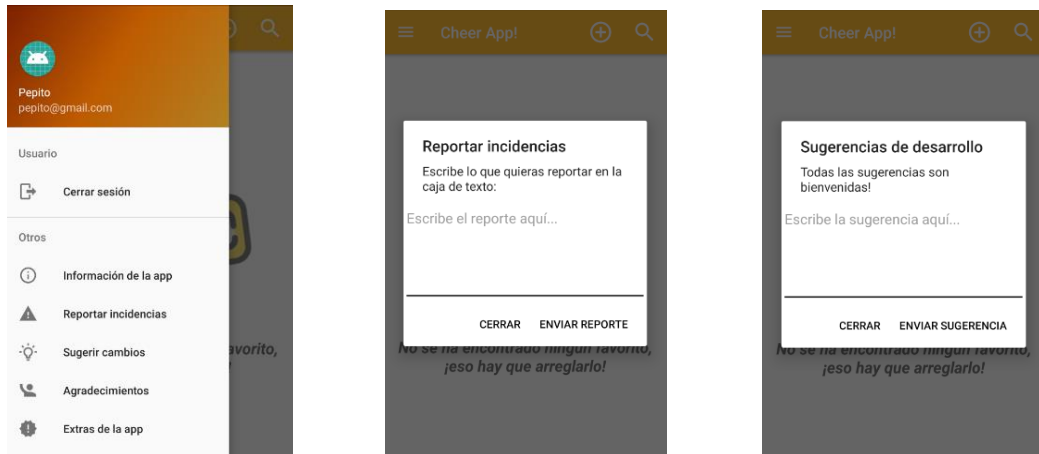


2. **Con elementos:** mostrará cada cerveza favorita del usuario como un ítem único en la lista, ordenados alfabéticamente por su nombre.



Como se puede apreciar, **existe un control en la parte superior izquierda. Al clicarlo o al arrastrar desde el margen izquierdo, se acciona el cajón lateral de navegación o "Navigation Drawer",** que cuenta con acciones de administración del usuario y otros, tales como reportes y sugerencias, que muestran un diálogo emergente en el que el usuario podrá escribir una sugerencia o reporte. Dichos textos se almacenarán en la BDD, teniendo como campos el texto enviado y el usuario que lo envió.

De izquierda a derecha: muestra de Navigation Drawer y ejemplos de reporte y sugerencia.



Ejemplo de reporte y sugerencia en la BDD.

```

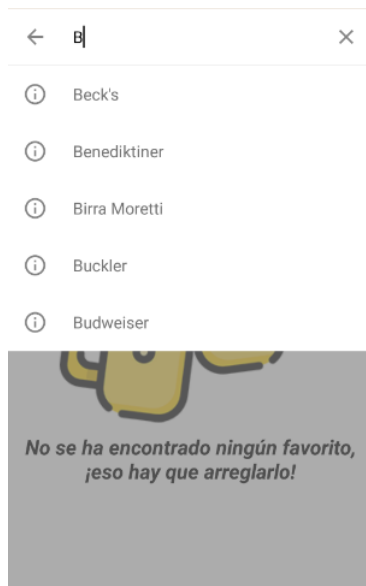
reportes
  -L01hxD7YnRCdIH7Kp5
    reporte: "Podría mejorarse.."
    usuario: "noVgIY38ECMPg00tVRVnRw3hpC0"
sugerencias
  -L01heaN2D4RLB8yXJEe
    sugerencia: "Todo va fenomenal"
    usuario: "noVgIY38ECMPg00tVRVnRw3hpC0"
  
```

También existen 2 controles en la barra superior: añadir cerveza (**abre la NuevaCervezaActivity**) y buscar cervezas (**abre una búsqueda para que el usuario pueda acceder a todo el catálogo de cervezas que hay en la BDD**). Veamos primero la búsqueda.

Dicho sistema de búsqueda se basa en una librería externa llamada [MaterialSearchView](#), creada por [Miguel Catalán Bañuls](#). He decidido usar esta librería ya que **ofrece facilidades a la hora de poblar la lista de sugerencias y trabajar con sus eventos**. Las sugerencias también se encuentran ordenadas alfabéticamente.

La lista se rellena con los elementos del catálogo de la BDD, y **según se vaya escribiendo va cambiando las sugerencias que se muestren al usuario**. Una vez que el usuario seleccione una de las sugerencias, **mostrará la DetalleCervezaActivity para dicha sugerencia**.

Ejemplo de búsqueda

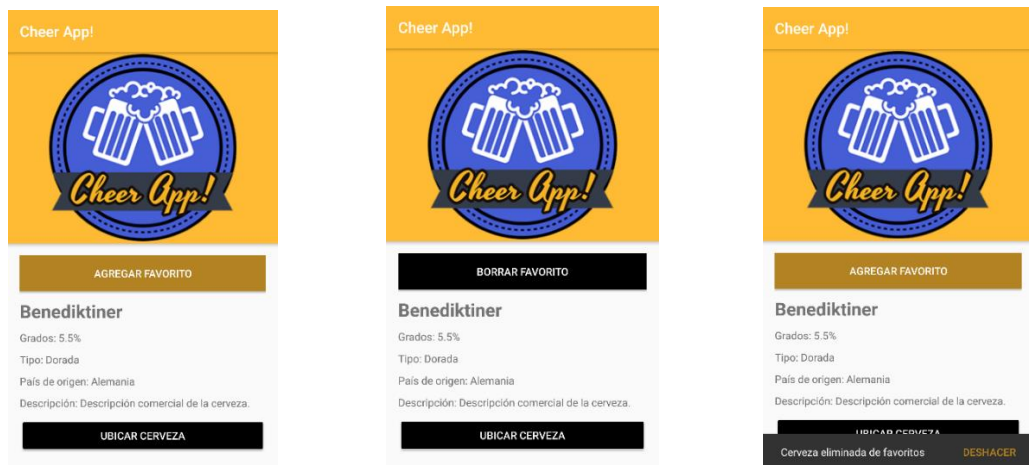


viii. *DetalleCervezaActivity*:

Activity que muestra los detalles de cada una de las cervezas almacenadas en la BDD, según cuál se seleccione. Dichos detalles son el nombre, la gradación, el tipo, el país de origen, la imagen y la descripción. Se puede acceder a esta Activity a través del menú contextual de los ítems de la lista de favoritos del usuario (al mantener presionado un ítem) y desde la búsqueda. Contiene controles para elegir si se quiere o no marcar como favorito y para ubicar la cerveza de la que se está viendo el detalle en ese momento.

Al añadir la cerveza a favoritos, el estilo del botón cambia, y al eliminarla ocurre lo mismo, y sirve como distintivo para dar a entender al usuario que la operación se ha realizado. Antes de borrar, pregunta al usuario con una ventana de diálogo si está seguro, e incluso después de eliminar de favoritos se puede deshacer la acción clicando en el Snackbar inferior.

De izquierda a derecha: ejemplos de *DetalleCervezaActivity* sin cerveza como favorito, con cerveza como favorito y Snackbar inferior.



ix. *MapsActivity*:

Si se **clica el botón inferior de DetalleActivity** o se **accede a través del menú contextual de los ítems de la lista de favoritos del usuario (al mantener presionado un ítem)**, se **accede a la MapsActivity**, en la cual el usuario podrá **localizar una cerveza**. Se hace de esta manera en lugar de seleccionar qué cerveza dentro de la propia **MapsActivity** para **evitar demasiadas llamadas de actualización al Fragment de Maps**, ya que conllevaría un mayor gasto de batería y consumo de datos móviles.

La Activity, al iniciarse, **recoge las ubicaciones almacenadas en la BDD** para esa cerveza y **coloca un marcador en el mapa por cada una de ellas**, y aplica un **zoom al mapa inicial de valor 1**, y anima durante **2 segundos el zoom hasta llegar al valor 15**. Estos valores están definidos por Google y son [los siguientes](#):

- **1: World**
- **5: Masa de tierra/continente**
- **10: Ciudad**
- **15: Calles**
- **20: Edificios**

Al clicar en uno de estos marcadores, muestra el título sobre él, y en la parte inferior derecha **se pueden apreciar 2 controles o acciones**, que son **propias del Fragment de Maps: navegación y localización en la app Google Maps**.

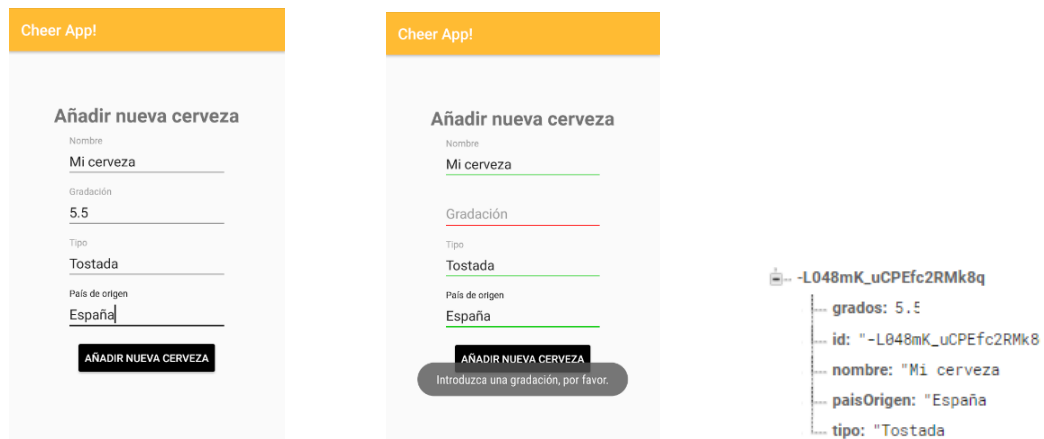
De izquierda a derecha: ejemplo de marcadores, controles del mapa y ubicaciones almacenadas en la BDD.



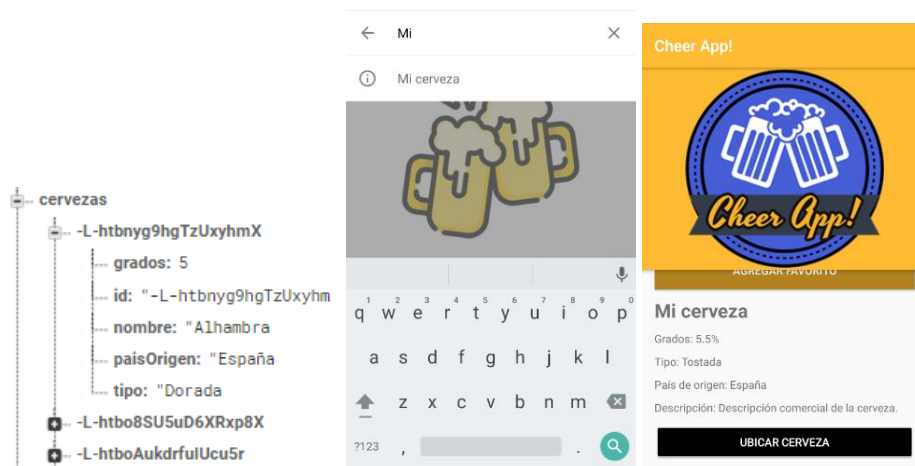
x. *NuevaCervezaActivity:*

Tal y como se había mencionado previamente, también existe un apartado de la aplicación en el que el usuario puede agregar sus propias cervezas. Para acceder a él, el usuario debe pulsar el botón de la barra superior que se encuentra en *HomeActivity*, lo que le llevará a *NuevaCervezaActivity* para que pueda insertar valores en los campos y generar las cervezas. Estos campos sólo enviarán datos si cada uno de sus textos es distinto de nulo, es decir, si tienen algún texto insertado. Una vez que el usuario haya insertado los datos, puede clicar en “Añadir nueva cerveza”

De izquierda a derecha: ejemplo de cerveza a insertar, muestra de error y muestra de cerveza creada por el usuario.



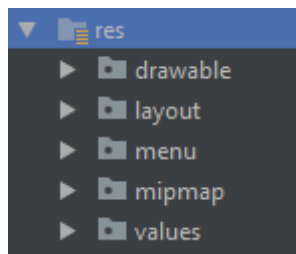
De izquierda a derecha: muestra de más cervezas insertadas en la BDD, búsqueda y detalle de la cerveza creada por el usuario.



e. Estructura del directorio Res:

Aquí se almacenan **distintos recursos**. Gran parte de estos archivos es **autogenerada por Android Studio cuando se crea el proyecto** (string.xml, colors.xml, styles.xml) o cuando se crea una nueva Activity (su “layout” o disposición de elementos, entre otros), pero **también en este directorio se almacenan otros recursos**:

- i. *Drawable*: guarda las imágenes y otros recursos gráficos que serán usados en la aplicación.
- ii. *Layout*: contiene las disposiciones visuales de las Activities, ítems de la lista, Navigation Drawer, etc.
- iii. *Menu*: recoge los layouts de los elementos de los distintos menús: el del Navigation Drawer, el de la barra superior, el contextual de los ítems de la lista de favoritos, etc.
- iv. *Mipmap*: almacena los iconos del launcher de la app (los que el usuario ve en su dispositivo y clic para abrir la aplicación).
- v. *Values*: guarda distintos archivos para distintos parámetros, los cuales son **colors.xml** (colores del tema y otros recursos de color), **dimens.xml** (las dimensiones de los elementos [predefinido y actualizado por Android Studio]), **las claves de API** (como puede ser la de Maps), **strings.xml** (distintos recursos en forma de cadenas de caracteres, para títulos de Activities, arrays de strings, etc), y **styles.xml** (define parámetros visuales de elementos, como por ejemplo si mostrar la barra superior en una Activity o no, que el menú de la barra superior se muestre bajo la barra superior y no sobre ésta, etc.).

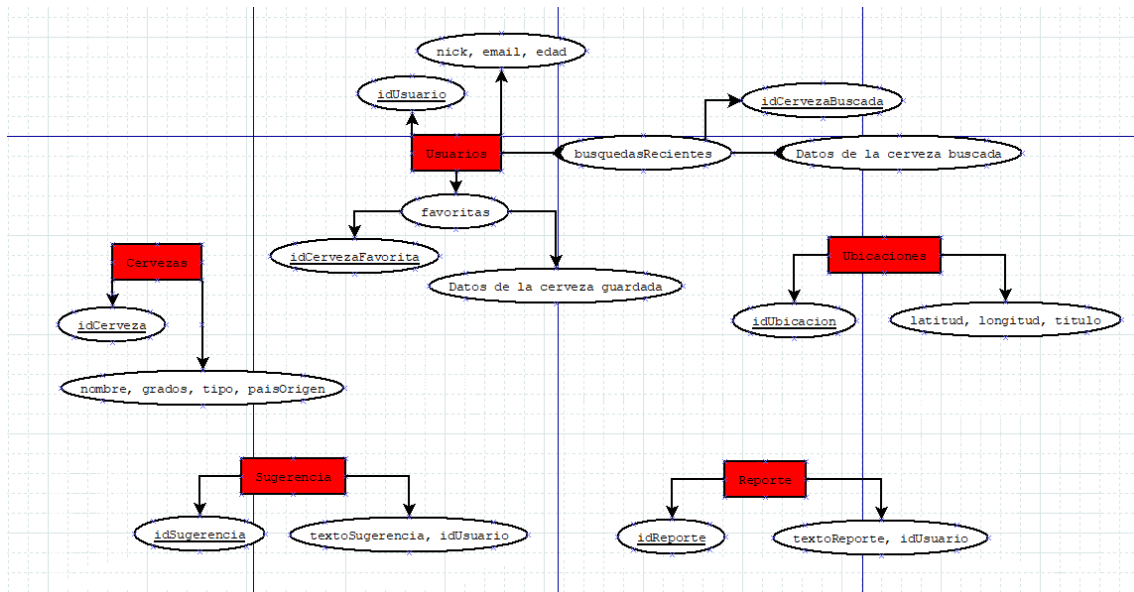


A continuación, se nombrarán los distintos diagramas que constan del proyecto.

f. Diagramas de clases

Se adjuntan a la entrega del proyecto. Se recomienda el uso de [Software Ideas Modeler](#) para poder abrirlos correctamente.

g. Diagrama de BDD



7. Tecnologías utilizadas

Las tecnologías utilizadas, como ya se ha mencionado previamente en otros puntos de este documento, han sido las siguientes:

a. Android Studio

Al haber trabajado con ella previamente en el Ciclo de Grado Superior de DAM, fue fácil de usar por familiaridad. Ofrece una amplia gama de opciones para desarrollo de todo tipo, como pueden ser clases Java básicas, Activities (ambos Java y disposición de elementos) y conexión rápida y fácil con otras tecnologías como Firebase y el API de Google Maps gracias a la sincronización con una cuenta de correo electrónico de Google Mail que puede ser usada para autenticarse en ambos sistemas.

b. Google Firebase

La decisión de usar esta tecnología se basa en varios aspectos. El primero de ellos sería la facilidad de conexión con Android Studio, seguido por la facilidad para usar sus servicios, habiéndose usado en este proyecto los servicios de Realtime Database (BDD en tiempo real) y Authentication (autenticación de usuarios). Otro aspecto a tener en cuenta es el hecho de que los cambios realizados por el usuario tienen un efecto real inmediato para con la aplicación, como en el caso de la lista de favoritos, que es actualizada siempre que el usuario realice algún cambio en la misma. También cabe destacar que, teniendo en cuenta que mi propósito es implementar los servicios de la aplicación de Android en una versión para navegadores web, un punto a favor de esta tecnología es la comodidad para poder relacionar los mismos sistemas de autenticación y almacenamiento que figuran en la aplicación con dicha versión web.

c. Google Maps

Esta tecnología fue seleccionada por los mismos motivos que Firebase en cuanto a comodidad de sincronización con proyectos de Android, y además debido a que está implementado en dispositivos Android de manera nativa. Su API es fácil de usar en cuanto a metodología, y además ofrece soporte visual para el usuario gracias a su mapa digital.

d. GitHub

Se ha usado GitHub por la facilidad para mantener un registro de los cambios realizados, y por la comodidad de relacionar el mismo repositorio en distintos equipos.

Todas las tecnologías anteriores tienen, además, otro punto en común que las favorece, y es que son tecnologías de uso gratuito. Existen planes de pago para acceder a funcionalidades mejoradas, pero para este proyecto no se necesita más que la versión básica y gratuita de las mismas.

8. Código documentado

El código fuente del proyecto y su documentación se pueden encontrar disponibles junto a la entrega de esta memoria.

9. Presupuesto

El presupuesto de la aplicación es el siguiente:

MATERIAL	COSTE (Euros)
Android Studio	0€
Google Firebase	0€
Google Maps API	0€
GitHub	0€
MaterialSearchView (plugin)	0€
Ordenador de sobremesa	0€ (previamente adquirido)
Dispositivo Android	0€ (previamente adquirido)

10. Planificación

A continuación, figura la **planificación para el desarrollo del proyecto**.

TAREA	Nº DIAS	INICIO	FIN
ANÁLISIS DE PROYECTO	35 DÍAS (CONTINUAMENTE)	5/11/2017	10/12/2017
Estructura de las clases	2 días	05/11/2017	06/11/2017
Estructura de la BDD	2 días	07/11/2017	08/11/2017
Búsqueda de recursos visuales	2 días	09/11/2017	11/11/2017
Documentación (continuamente)	30 días	11/11/2017	10/12/2017
DISEÑO Y DESARROLLO	30 DÍAS	11/11/2017	10/12/2017
Desarrollo BDD Firebase (continuamente)	30 días	11/11/2017	10/12/2017
SplashScreenActivity	½ día	11/11/2017	11/11/2017
Conexión Firebase-Android	½ día	11/11/2017	11/11/2017
Clase ReferenciasFirebase (continuamente)	29 días	11/11/2017	10/12/2017
MainActivity	3 días	12/11/2017	14/11/2017
AdaptadorSeccionesInicio	½ día	12/11/2017	12/11/2017
Clase Usuario	½ día	12/11/2017	12/11/2017
Fragments Login y Registro	2 días	13/11/2017	14/11/2017
HomeActivity (continuamente)	25 días	15/11/2017	06/12/2017
ListaPersonalizada	1 día	15/11/2017	15/11/2017
Clase Cerveza (continuamente)	14 días	16/11/2017	29/11/2017
DetalleCervezaActivity (continuamente)	15 días	16/11/2017	30/11/2017
NuevaCervezaActivity	2 días	01/12/2017	02/12/2017
MapsActivity	2 días	09/12/2017	10/12/2017
Clase UbicacionCerveza	1 día	09/12/2017	09/12/2017
PRESENTACIÓN DEL PROYECTO	1 DÍA	16/12/2017	16/12/2017

Leyenda:

CABECERA DE LA TABLA
SECCIONES DE LA TABLA
Cabecera de colección de tareas relacionadas
Apartado de la colección de tareas relacionadas

11. Manual de instalación

a. Requisitos de hardware y software

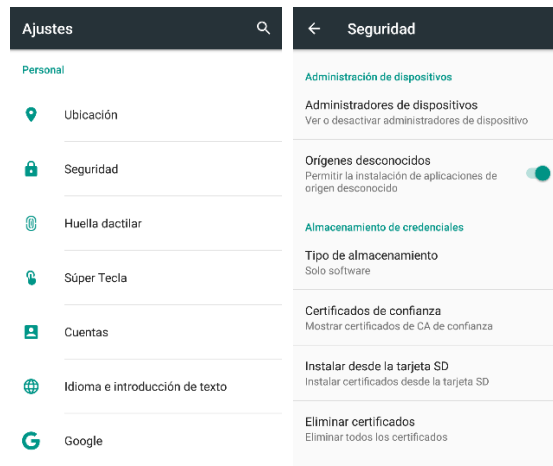
- Dispositivo Android con versión de Android 5.0 al menos.
- Conexión a Internet.

b. Instalación

La guía de instalación se realizará en un dispositivo Android 6.0, lo que podría significar que haya diferencias en cuanto a localizaciones de ajustes, en concreto en referencia al primer paso.

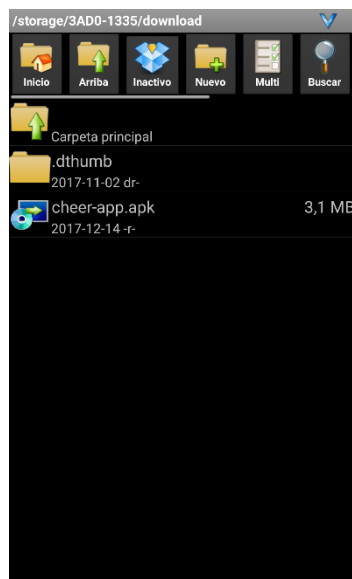
1. Configurar el dispositivo para permitir instalar aplicaciones de origen desconocido

Se debe navegar a **Ajustes > Seguridad > Administración de dispositivos** y marcar la opción **Orígenes desconocidos**.



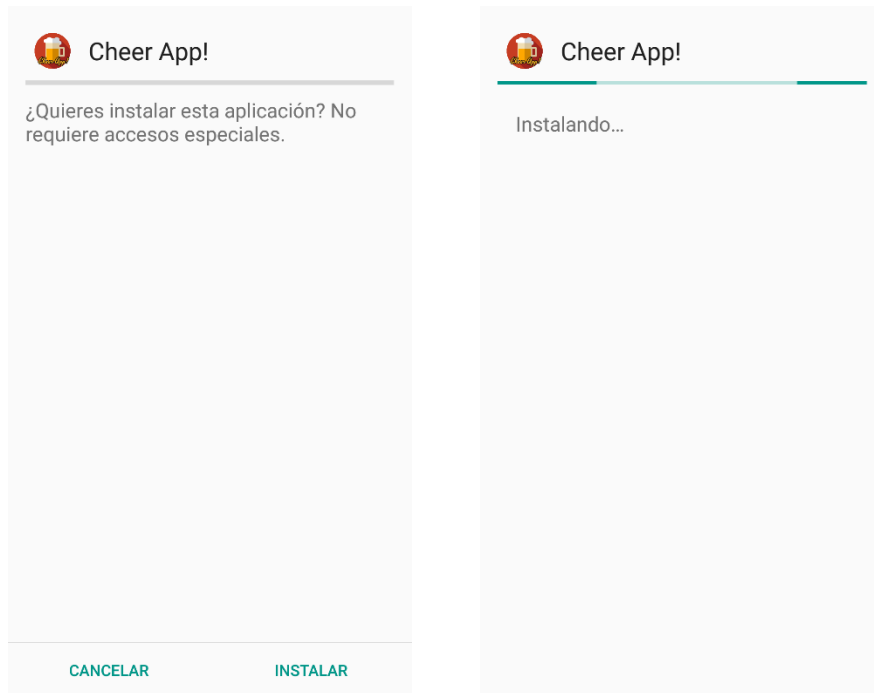
2. Localizar el archivo ejecutable .apk

Usando un sistema gestor de archivos, **localizar** el archivo de instalación con formato **.apk** en el dispositivo.



3. Instalar la aplicación

Una vez localizado el archivo, se debe instalar. Para ello, hay que seleccionar “Confirmar en el menú de instalación”



4. La aplicación ya está lista para usarse.

12. Manual de usuario

Las funcionalidades del usuario han sido explicadas en puntos previos del apartado [“6.1 Interfaz de usuario”](#).

13. Análisis de competencia

Existen varias aplicaciones similares a la desarrollada en este proyecto en el mercado de aplicaciones, con la diferencia de que **están orientadas a públicos de otras nacionalidades**, y al ser la de este proyecto orientada a público español, no existe competencia.

14. Conclusiones

Como ya he mencionado previamente, **esta aplicación no cubre todas las ideas que tenía en mente al realizarla**, pero dichas ideas **serán añadidas posteriormente para la publicación oficial**. Es evidente **que queda mucho por mejorar y pulir**, y **pretendo alcanzar un matiz de red social con esta app gracias a la interacción entre usuarios**, y hacer una enumeración de esas ideas podría resultar beneficioso para el desarrollo de la app.

a. Mejora de Firebase:

El modelo de datos actual **será revisado para tratar de otra manera los mismos datos que contiene**. Para empezar, **las cervezas que añade el usuario serán añadidas a la BDD como “sugerencias de cerveza”** en lugar de como cervezas definitivas, a la espera de ser moderadas y revisadas para ser añadidas al catálogo definitivo.

Lo mismo ocurrirá con las ubicaciones de las cervezas. Además, en el momento de la entrega de este proyecto, **todas las cervezas comparten las mismas 2 ubicaciones, pero cada cerveza tendrá sus propias ubicaciones, cada una de ellas etiquetada según su naturaleza** (bodega especializada, cervecería, supermercado, etc.) **para que el usuario, al entrar en la MapsActivity, pueda seleccionar el criterio a la hora de encontrar cervezas según sus preferencias**.

Se implementará un **sistema de reseñas/comentarios de las cervezas**, de tal manera que **todos los usuarios puedan comentar acerca de sus cervezas favoritas, sus ubicaciones, etc. y puedan valorar los comentarios de otros usuarios**.

El punto anterior lleva a la mención del **sistema de promoción de roles de usuario que se prevé llevar a cabo**. Dicho sistema hará que **los usuarios nuevos se inicien con un rol básico que sólo les permita buscar cervezas y guardarlas como favoritas, así como votar comentarios y añadir sugerencias de cervezas**. A medida que vayan **recibiendo puntos de sus comentarios y sugerencias de cervezas**, podrán ser promovidos a otros roles.

Por el momento mi decisión es la de **no incluir a los usuarios como moderadores de contenido la BDD, sino contar con la ayuda de conocidos y personas cercanas a mí que estén interesados en ayudar, ya que dar ese poder a los usuarios podría resultar en problemas para el proyecto**.

Se comenzarán a **almacenar imágenes comerciales de las cervezas en el sistema de Firebase Storage, para poder mostrarlas en la aplicación en lugar de una imagen por defecto, como sucede ahora, y también para las descripciones comerciales**. Este cambio no se ha realizado antes porque **suponía una inversión de tiempo que no era posible de realizarse por cuestiones de organización y fechas límite, y era preferible invertir en otras funciones de desarrollo de la app**.

b. Mejora de la app de Android:

Se mejorarán varios apartados de la app. Para empezar, **se mejorará el sistema de inserción de edad para poder recoger la fecha de nacimiento del usuario, de tal manera que se le pueda felicitar por su cumpleaños.**

Se mejorarán **aspectos visuales, como controles, elementos de la barra superior, del Navigation Drawer** (los elementos de la cabecera, sobre todo), **y ventanas explicativas del funcionamiento de la app** (se planea hacer usando [la librería de APL Developers, AppIntro](#)), y también las solicitudes de permisos de los usuarios (usando la [librería PermissionsHelper, también de APL Developers](#)).

Funcionalidades como **administración de la cuenta para solicitar cambios de contraseña, nick de usuario, etc., entre otras.**

También el **sistema de búsqueda de cervezas de la app**; ahora mismo logra guardar las búsquedas recientes de los usuarios a modo de historial, y se codificará de tal manera que puedan ser mostradas como sugerencias cuando el texto de búsqueda esté vacío, y al escribir pueble las sugerencias con el catálogo de la app. Asimismo, **implementar un sistema de búsqueda basado en distintos criterios** además del nombre, tales como tipo de cerveza, gradación, país de origen o cercanía al usuario en ese momento. De esta manera se generarán nuevas formas de encontrar la cerveza deseada.

Implementar un sistema de sugerencias de cervezas para mostrar al usuario basadas en distintos criterios, tal como funciona en Amazon, por ejemplo.

El sistema de ubicación de usuarios cambiará en algunos aspectos, tales como que el mapa recogerá la ubicación actual del usuario para usarla como centro del mapa, para así poder usar las ubicaciones de las cervezas más cercanas al usuario, aunque por supuesto también podrá seleccionar puntos más lejanos si así lo desea.

He buscado APIs y colecciones de datos para las cervezas, pero ha sido una búsqueda infructuosa. No obstante, **buscaré o generaré los míos propios para otros aspectos de la app**, como por ejemplo para usar en **controles desplegables en NuevaCervezaActivity para los tipos de las cervezas** ("Trigo", "Cebada", "Lúpulo", etc.) y **los países de origen** (confío en que este sea más fácil de encontrar).

Se planea añadir **otros entornos de la app**, tales como la **posibilidad de añadir amigos o guardar ubicaciones frecuentes además de solo cervezas**, e incluir **notificaciones Push para informar al usuario de acciones relacionadas con su cuenta.**

Como se puede observar, el proyecto podría haberse ampliado, y es por eso que solo he incluido lo que verdaderamente consideraba suficiente para presentar como TFG.

15. Bibliografía:

- [Stackoverflow](#)
- [Android Developers](#) y [Android Developers YouTube](#)
- [Firebase Documentation](#)