

# **BEER APP**

**Desarrollo de Aplicaciones Avanzadas**  
Práctica final - Curso 2023/24

# Tabla de contenidos

BEER APP	1
1. Introducción	3
2. Manual de usuario	4
3. Manual del programador	6
4. GitHub	7

# 1. Introducción

Se trata de una aplicación en Swift para dispositivos iOS cuyo objetivo es mostrar de forma sencilla las distintas variedades de cerveza que se comercializan en la actualidad.

Cuenta con tres vistas principales gobernadas por un *TabView* que divide la aplicación en los apartados de Inicio, Cervezas y Usuarios:

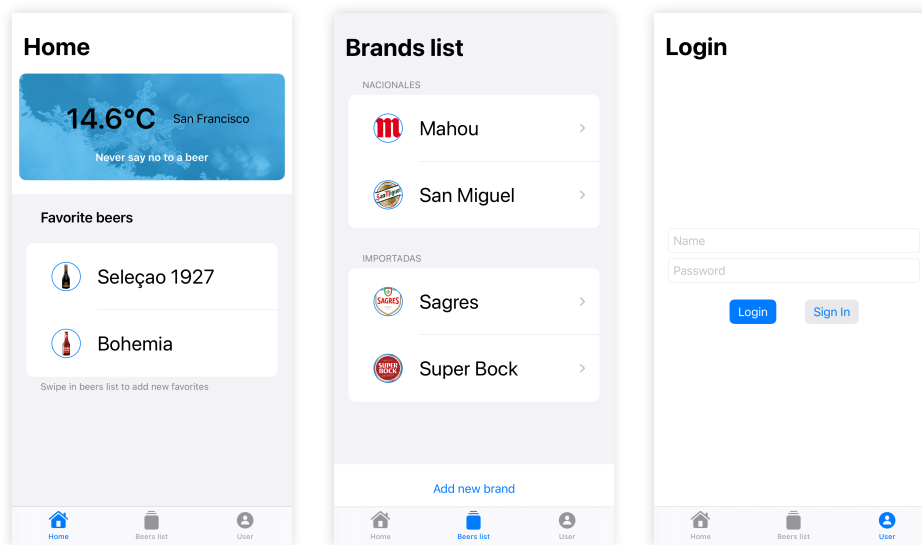
- Inicio: es la pantalla principal que se muestra al abrir la aplicación. Cuenta con un widget del tiempo que, en función de la temperatura actual, muestra un mensaje personalizado con la temática de las cervezas, para dotar a la aplicación de un carácter fresco y divertido. También, sitúa debajo un listado con las cervezas marcadas como favoritas en el apartado siguiente.
- Cervezas: es la interfaz más relevante de la aplicación, donde se muestra, en primer lugar, el listado con los principales fabricantes que incluyen a su vez un listado con las cervezas ofrecidas por cada uno. Al instalar la aplicación por primera vez, esta cargará un conjunto de fabricantes y sus cervezas por defecto entre los que se encuentran las portuguesas importadas “Sagres” y “SuperBock” junto a las nacionales “San Miguel” y “Mahou”. El listado se encuentra categorizado por esta misma división y el de cervezas por los tipos: negra, pilsen, mixta y lager. Cuenta con las funcionalidades de búsqueda mediante una SearchBar y de ordenación según grados, calorías o nombre. También permite añadir, eliminar o editar nuevas cervezas a las ya existentes.
- Usuarios: es un apartado adicional que permite registrarse o iniciar sesión con un nombre de usuario y contraseña, para acceder a un espacio privado donde poder añadir y consultar tus propias reseñas sobre todas las cervezas de las que dispone la aplicación. Cuenta con un formulario y puntuaciones por estrellas, siendo 1 estrella la puntuación menor y 5 la más alta.

Funcionalidades añadidas: además de las ya mencionadas como el uso de la API del tiempo, las reseñas o la gestión de usuarios, la aplicación es totalmente adaptable al modo oscuro del iPhone y está íntegramente traducida al español y el inglés.

Tras esta breve introducción, a continuación se incluye un pequeño manual de usuario para explicar todas las funcionalidades y sacarle el máximo partido, aunque se tratan de interfaces de usuario intuitivas y fáciles de usar, que cumpla con las directrices de la interfaz humana de Apple. Por último, en el manual del programador, se detallarán los principales componentes y decisiones de diseño.

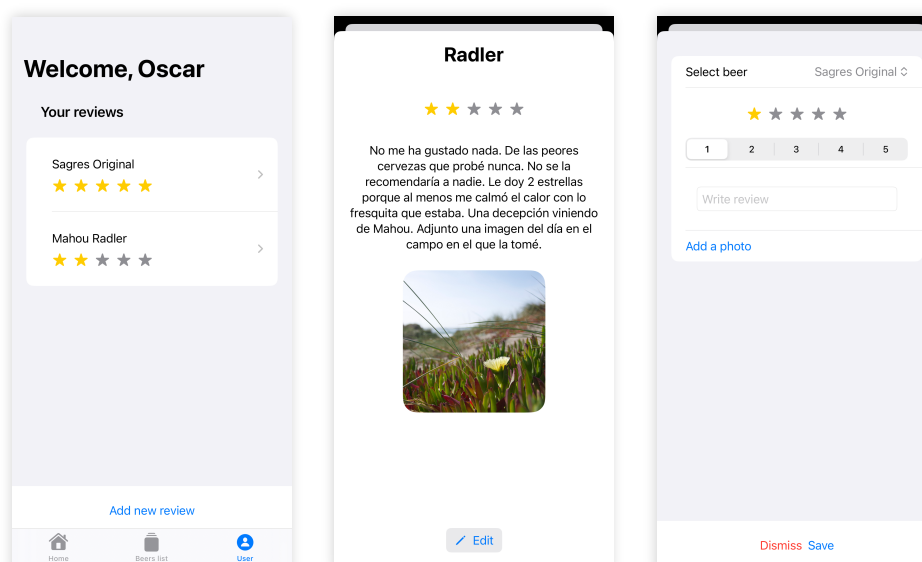
## 2. Manual de usuario

En primer lugar, se muestran las tres pantallas principales de las que se compone la aplicación: Inicio, Cervezas y Usuarios, de izquierda a derecha.

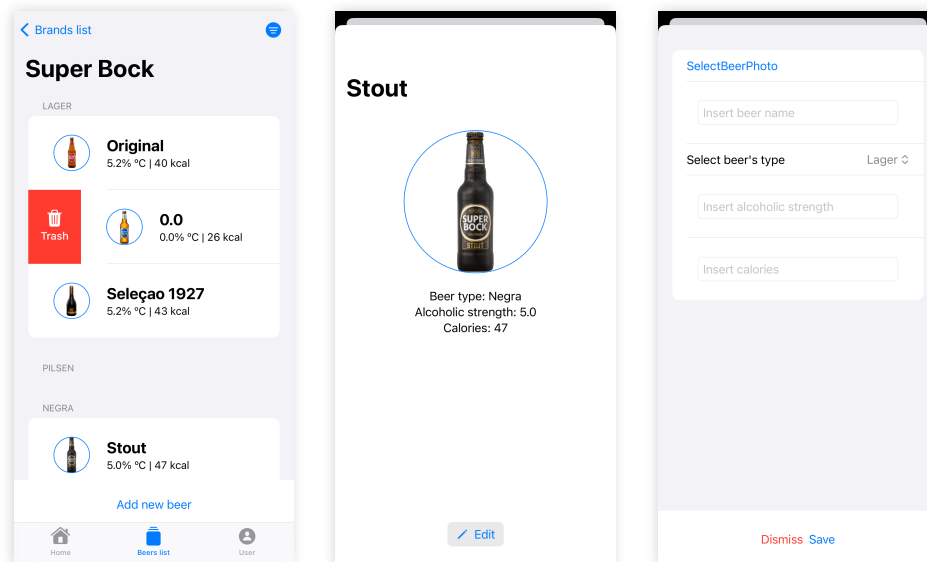


En la primera de ellas, se muestra el widget del tiempo y el listado de las cervezas marcadas como favoritas. Con un deslizamiento de derecha a izquierda se pueden eliminar de esta lista. La segunda, muestra el listado de fabricantes. Con un deslizamiento de izquierda a derecha pueden eliminarse un fabricante y todos sus datos asociados, mientras que haciendo click sobre uno se muestra una nueva pantalla con sus cervezas ofertadas. Por último, la tercera, nos permite ingresar a la aplicación con un nombre de usuario y contraseña mediante el botón “Login” o creando una cuenta nueva si no existe a través de “Sign In”. Si alguno de los datos introducidos es erróneo, mostrará un mensaje informando sobre ello.

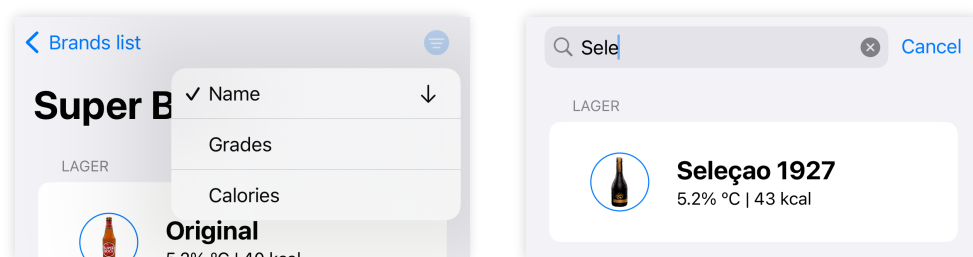
Continuando con esta última, una vez se accede correctamente, la aplicación mostrará el apartado privado del usuario con sus reseñas sobre distintas cervezas. Al igual que en el resto de listados de la aplicación, se podrán eliminar deslizando desde la izquierda y se mostrará la información detallada pulsando sobre una. Además, haciendo click sobre el botón de añadir se muestra un formulario para crear una nueva reseña, con un selector de la cerveza hacia la que va a ir dirigida y otro para la puntuación por estrellas.



Regresando a la pantalla general del listado de fabricantes, de manera similar que con las reseñas, se pueden añadir nuevos a través de un formulario, que solicita la información requerida del nombre, la categoría y una fotografía del logo. En la vista detallada muestra las cervezas de su catálogo con nombre, graduación y aporte calórico, que se amplían de igual manera en su vista detallada correspondiente. En esta se presenta la opción de poder editarla a través del mismo formulario utilizado para añadir nuevas cervezas, solo que en este caso con los campos ya rellenos con la información original. Nuevamente, al igual que en los listados anteriores, se podrán eliminar deslizando desde la izquierda y marcar como favoritas haciéndolo desde la derecha.



Para finalizar, la aplicación cuenta con funcionalidades de filtrado en los listados de cervezas, pudiendo seleccionar en el menú superior *Toolbar* el campo por el que ordenar todas las entradas de la *Lista* en cada categoría. También se puede buscar por nombre mediante la barra de búsqueda estándar de todas las aplicaciones iOS.



### 3. Manual del programador

El proyecto sigue la arquitectura MVVM, para separar la lógica de la interfaz de usuario de la lógica de negocio. Está claramente dividido en tres carpetas principales: M (Modelo), VM (Vista Modelo) y V (Vista).

#### MODELO

La aplicación cuenta con una base de datos CoreData denominada “Banking”, en la que se definen las entidades BeerE (cerveza), BrandE (fabricante), UserE (usuario) y ReviewE (reseña), con sus respectivos atributos y relaciones. Para controlar el acceso a la misma, se utiliza la clase CoreDataManager, que implementa el patrón Singleton para asegurar que exista una única instancia en todo momento. Cuenta con un contenedor donde se cargan todos los datos del CoreData y una serie de funciones agrupadas en tres grandes categorías: creación, eliminación y obtención.

Pese a la existencia de estas entidades, internamente el modelo utiliza unos tipos nuevos para las cervezas y los fabricantes, definidos en las clases Beer y Brand respectivamente. Esto se debe a que en una primera versión de la aplicación no se utilizaba aún CoreData y eran clases en un archivo json lo que se almacenaba. El código utilizado en esa versión anterior se ha mantenido comentado, por si pudiera ser de utilidad en un futuro. Para hacer la conversión de las clases a las entidades de la base de datos, se definen una serie de métodos en el *ViewModel*, pues se consideró que era la manera más sencilla en lugar de modificar toda la lógica del proyecto. Para los usuarios y las reseñas, sí que se utilizan directamente las entidades de CoreData, pues se implementaron a posteriori.

#### VIEWMODEL

Se divide el proyecto en dos grandes paquetes para separar la gestión de usuarios (y reseñas) en *UserViewModel* y la gestión de cervezas en *ViewModel*. Ambos con métodos de creación, eliminación, ordenación y carga u obtención de datos. La función `fetchAll` del *ViewModel* de cervezas es la encargada de obtener los datos desde CoreData y hacer la transformación a las clases propias del modelo explicadas anteriormente, que cuentan con los mismos atributos que sus entidades equivalentes.

#### VIEW

La aplicación está completamente traducida al español y el inglés por medio del archivo “Localizable.xcstrings” que almacena pares clave valor con todos los textos mostrados en las distintas interfaces. Para acceder a ellos se hace de esta manera:

```
String(localized: “Clave”)
```

Además, se define un conjunto de Views personalizadas en CustomViews que encapsula código reutilizable en otras partes del código, como pueden ser los items de cervezas en las listas, las imágenes con el estilo circular determinado o los distintos pickers de los formularios. Esto se hace para mantener un código limpio siempre con menos de 60 líneas de código en cada vista.

En ContentView, se establecen las instancias del ViewModel de cervezas y de usuarios como EnvironmentObject para que puedan ser accesibles desde cualquiera de las demás vistas y mantener correctamente la misma información. Tanto las reseñas como las cervezas y fabricantes cuentan con las vistas de añadir, que presenta un Form para introducir nuevos datos; las vistas de listados, que muestran los datos en un List con secciones para cada categoría; y las vistas de detalle, en las que se muestran los datos detallados del objeto en cuestión. Para navegar de unas a otras, se utilizan botones que invocan una sheet con la nueva interfaz, o en su defecto un NavLink.

## 4. GitHub

Por último, se ha utilizado Git y GitHub para el control de versiones de la aplicación. Adjunto el enlace al proyecto en esta plataforma, aunque se encuentra de momento en privado hasta que finalice la evaluación de la asignatura.

[https://github.com/okhergo/BeerApp\\_DAA\\_GII](https://github.com/okhergo/BeerApp_DAA_GII)