

# Práctica Final 1ª Convocatoria - Descripción de la App

Un centro dedicado al estudio y promoción de la cerveza necesita visualizar de forma sencilla las distintas variedades de cerveza que se comercializan en la actualidad. Con este fin, se desea construir una aplicación para iOS que ofrezca al usuario las vistas que a continuación se describen. La aplicación deberá ser construida utilizando Swift y SwiftUI, siguiendo las mejores prácticas de diseño y programación.

## Vista 1: Inicio - Lista de Fabricantes

- Mostrar una lista interactiva utilizando List en SwiftUI, con secciones claramente diferenciadas para "Nacionales" e "Importadas".
- Cada entrada en la lista debe incluir el logotipo del fabricante y su nombre, implementados respectivamente con Image y Text.
- Incluir un botón Button para "Añadir fabricante" que lleve al usuario a una AddManufacturerView para ingresar los datos de un nuevo fabricante mediante un formulario Form.
- Proporcionar la opción de eliminar fabricantes y todas sus cervezas con un gesto de deslizamiento, utilizando las funcionalidades de SwiftUI.

## Vista 2: Detalle del Fabricante

Al pulsar una de las filas de la primera vista se pasa a una vista de detalle del fabricante seleccionado. En esta vista aparecen varias secciones, una por cada tipo de cerveza que ofrezca el fabricante (lager, Pilsen, IPA, Pale Ale etc). En cada una de estas secciones se muestran filas que describen la cerveza en cuestión. En cada fila se muestra el nombre de la cerveza y una fotografía o símbolo).

- Esta vista detallada se activa al seleccionar un fabricante de la Vista 1.
- Mostrar categorías de cerveza por tipo (Lager, Pilsen, IPA, etc.) en secciones individuales dentro de una List.
- Cada tipo de cerveza se representa con una fila que incluye una imagen o símbolo y el nombre de la cerveza, usando Image y Text.
- Incluir botones para "Añadir cerveza", "Eliminar cerveza", "Ordenar" y "Volver", cada uno con su funcionalidad correspondiente.
- El botón "Ordenar" puede presentar un selector o una vista auxiliar para organizar las cervezas por nombre, graduación alcohólica o aporte calórico.
- Integrar una función de búsqueda para filtrar los tipos de cervezas, posiblemente usando SearchBar de SwiftUI.
- El botón "Volver" retorna a la vista principal.

### Vista 3: Detalles de la Cerveza

- Al seleccionar una cerveza de la Vista 2, se presentará un detalle más profundo de la cerveza seleccionada.
- Mostrar y permitir la edición de los datos de la cerveza (nombre, graduación, aporte calórico, tipo, fotografía) a través de un formulario Form.
- Proporcionar botones para "Aceptar" o "Cancelar" cambios, regresando a la vista anterior.

### Funcionalidades Extras

#### \*Vista 4: Reseñas de Cervezas

Esta vista permitirá a los usuarios crear y visualizar reseñas sobre las cervezas que han probado.

- Agregar Reseñas: Los usuarios podrán tocar un botón "+" que abrirá un formulario donde podrán escribir su reseña, puntuar la cerveza y añadir una imagen.
- Visualización de Reseñas: Las reseñas agregadas se mostrarán en una lista, donde cada entrada es seleccionable para ver detalles completos.
- Edición de Reseñas: Los usuarios tendrán la opción de editar sus reseñas existentes utilizando gestos de deslizamiento o a través de un botón de edición en la vista de detalle de la reseña.
- Eliminar Reseñas: De manera similar a la edición, se podrán eliminar reseñas con un gesto de deslizamiento o desde la vista de detalle.
- Ordenar y Filtrar: Se podrán aplicar filtros y ordenar las reseñas por fecha, puntuación o nombre de la cerveza.

#### \*Utilizar una API

A la alternativa de utilizar serialización o ficheros se puede cargar datos desde un servidor mediante una API REST, utilizando **URLSession** para las peticiones HTTP y **JSONDecoder** para el procesamiento de los datos JSON recibidos.

---

### Requisitos de Desarrollo:

- Aplicar MVVM para la arquitectura de la aplicación, separando la lógica de la interfaz de usuario de la lógica de negocio.
- Comentar el código fuente para explicar las decisiones de diseño y la funcionalidad de los componentes clave.
- Asegurarse de que la interfaz de usuario cumpla con las directrices de la interfaz humana de Apple y sea accesible y fácil de usar.

# Consideraciones Generales

## Gestión de Datos Iniciales:

La aplicación cargará su conjunto inicial de datos de fabricantes y cervezas desde archivos precargados en el bundle del proyecto. Este proceso se llevará a cabo únicamente en el primer lanzamiento de la aplicación. Posteriormente, se deben preservar los datos actualizados en la carpeta **Documents** del dispositivo cada vez que la aplicación se minimice o esté a punto de cerrarse. En los lanzamientos subsiguientes, la aplicación verificará primero la existencia de datos en Documents para su carga; si no se encuentran, recurrirá a los datos del bundle. Se descarta la necesidad de reescribir datos en Documents si estos ya existen, lo cual optimiza el rendimiento. Como alternativa avanzada, se sugiere la implementación de CoreData para una gestión más eficiente y robusta de la persistencia de datos.

**Optimizaciones y Mejoras Adicionales:** Aunque la especificación proporcionada establece los requisitos básicos, se incentiva la incorporación de características adicionales que enriquezcan la funcionalidad y la experiencia del usuario. Todas las optimizaciones realizadas serán debidamente valoradas.

**Documentación y Manual del Desarrollador:** Se deberá entregar un documento en formato PDF que destaque las funcionalidades destacadas y novedosas de la aplicación. Este documento debe incluir un manual para desarrolladores, explicando aspectos importantes como modificaciones necesarias para la adición de múltiples fotografías por cada tipo de cerveza, entre otros.

**Control de Versiones:** Es altamente recomendable utilizar GitHub para el control de versiones del proyecto. El proyecto puede ser compuesto por múltiples ramas nombradas secuencialmente por su versión (v1, v2 y v3), cada una representando diferentes etapas de mejora del proyecto. Cada rama subsiguiente debe incluir las mejoras de su predecesora. La rama principal, conteniendo la versión base del proyecto, se denominará 'main'.

## Instrucciones de Entrega

- El proyecto debe ser creado con Xcode 14.x para asegurar la compatibilidad y el uso de las últimas características de desarrollo.
- Debe ser codificado en Swift 5.x, aprovechando las actualizaciones más recientes del lenguaje.
- El sistema operativo de destino será iPadOS o iOS 16.x, y se espera que la aplicación funcione sin problemas en esta plataforma.
- Se debe garantizar que el proyecto sea compilable y ejecutable en los dispositivos del **Aula 5**
- Aunque se permite el uso de bibliotecas de terceros, se debe asegurar que estas no comprometan la compilación y ejecución del proyecto.

## Proceso de Compresión:

Es crucial que el proceso de compresión no altere ni dañe los archivos del proyecto. Para comprimir los archivos, simplemente selecciona el proyecto y los PDFs relevantes en el Finder y utiliza la opción "Comprimir archivos" disponible mediante el clic derecho o el menú de acciones.

## Fecha Límite de Entrega:

La práctica final debe ser entregada antes o en la fecha del **21 de diciembre de 2023**. Se recomienda encarecidamente no esperar hasta el último momento para evitar cualquier problema técnico que pueda surgir con la compresión o entrega del proyecto.