

# Práctica Frontend avanzado

## BeerFlix



## Briefing

Acabamos de cerrar un contrato con uno de nuestros mejores clientes. Dicho cliente, tiene como idea de negocio, revolucionar el mundo del consumo de cerveza dando un acceso rápido a los mejores productos del mercado. Para ello tiene una gran base de datos la cual es proporcionada por un API de un proveedor externo.

Las historias de usuario que contempla el cliente que tienen que ser necesarias (Requeridas para la práctica) son:

1. Un usuario deberá acceder a una web pública desde cualquier dispositivo. Una gran parte de los usuarios lo hará desde un dispositivo móvil. (IMPORTANTE -> Tener en cuenta al crear el proyecto)
2. Un usuario deberá ver listadas las cervezas que nuestro proveedor nos ofrece en un listado.
3. El usuario en la página principal deberá filtrar por texto y limitar los resultados a 10 max. Además de añadir otro filtro por fecha.

4. El cliente tiene como única especificación de diseño header con una imagen de fondo. (Dicha imagen tendrá que ser adaptada como en el curso)
5. Un usuario podrá navegar al detalle de una cerveza. Dicho detalle tiene que tener más información del API de nuestro proveedor que en el listado.
6. En dicho detalle el usuario tiene que tener la posibilidad de guardar un comentario acerca de ese producto o añadir likes a dicho producto. El cliente para la primera fase quiere al menos una de esas funcionalidades. Si es posible, tener ambas sería un plus para nuestros usuarios.

## Notas para el desarrollador

Nuestro lead developer nos ha dado estas especificaciones para el proyecto:

1. El proyecto es necesario que se realice sin framework (Vue, Angular, Ember, Polymer, Backbone etc...) y sin las librerías como React y jQuery.
2. Se podrán utilizar frameworks de CSS como Bootstrap, Foundation, Materialize, semantic entre otros. De estos framework solo podrá ser usado el css y html **NO EL JAVASCRIPT**
3. Será necesario el uso del API de nuestro proveedor. Las especificaciones serán explicadas más abajo.
4. Será necesario el uso de localStorage para guardar el estado de los filtros.
5. Añadir ESLINT al proyecto. (Opcional)
6. Estaría bien añadir workbox para tener soporte offline. (Opcional)

Se valorará, que las especificaciones por parte del cliente sean correctas y por supuesto se valorará la calidad del código y estructura del proyecto. En cuanto a la calidad del código se valorará que sea sencillo para nuevos desarrolladores, reusable y que sea fácil de mantener.

# Uso del API

Con el correo que me disteis por privado. (EJ [nombre@keepcoding.es](mailto:nombre@keepcoding.es)) tendréis que acceder a [https://web-bootcamp-exercise-beer-swagger-pgjtqotgtl.now.sh/api-docs/#/user/post\\_user\\_login](https://web-bootcamp-exercise-beer-swagger-pgjtqotgtl.now.sh/api-docs/#/user/post_user_login)

**POST** `/user/login` Register a new user into the app

Register a new user into the app

Parameters Try it out

Name	Description
<b>body</b> <small>required</small> (body)	Pet object that needs to be added to the store  Example Value   Model <pre>{   "email": "string" }</pre> Parameter content type <span>application/json</span>

En esta pantalla tendréis que hacer click en try it out y añadir vuestro mail. Y ejecutar el comando. Esto os devolverá un API key.

Dicha API key la tendréis que enviar como cabecera de vuestras peticiones. Tal y como se explico en el curso para poder resolver la historia de usuario 6 y 2.

En la documentación del proveedor (Swagger) podréis hacer las pruebas que queráis. Añadir likes o comentarios para familiarizaros con el API.

Recordar que para cualquier duda o curiosidad, estaré disponible por slack. Tanto durante el curso o por si hay alguna inquietud acerca de los puntos extras o alguna curiosidad o duda con la práctica.

**Mucha suerte con el ejercicio!!!**