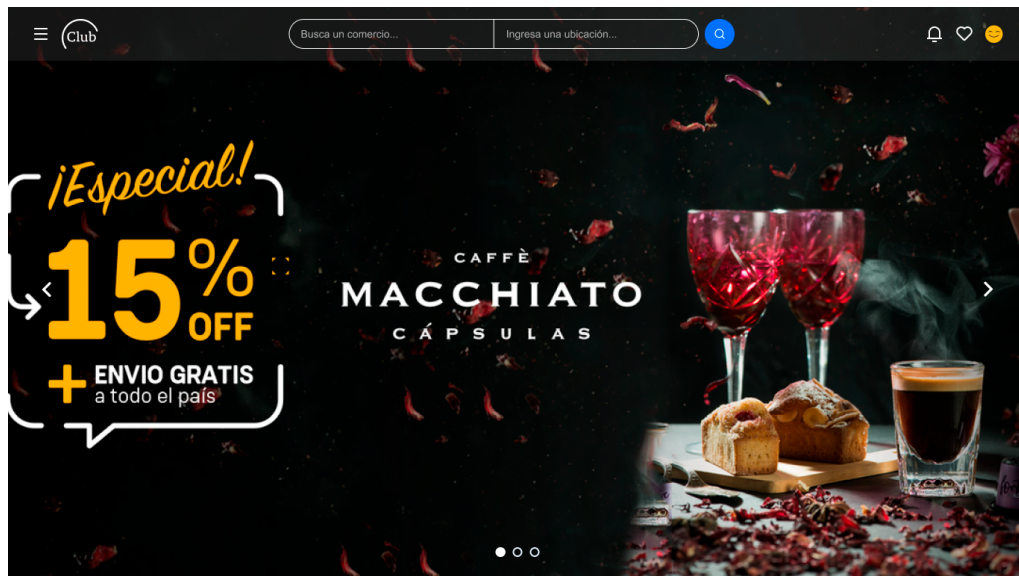


Assets

JSON:

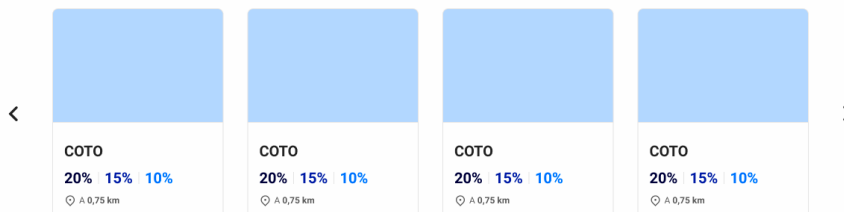
https://drive.google.com/file/d/1CXSPd7qqT3HvawoXTXMFn_1Uke7mtL6u/view?usp=sharing

Diseño:



Turismo en Buenos Aires

[TODOS LOS BENEFICIOS](#)



Códigos de descuento

¿Sos socio de Club LA NACION? Descargá tu código y disfrutá beneficios exclusivos en tus marcas favoritas

[TODOS LOS CÓDIGOS](#)



Objetivos:

- Se solicita maquetar a partir del diseño entregado, el layout provisto (sin necesidad de un slider funcional, ni un input de búsqueda funcional, pero los mismos deben estar en el proyecto entregado de forma estática.
- Componentizar todos los elementos que considere necesarios sin la utilización de librerías de terceros.
- Generar la API a partir del json provisto, que tenga los endpoints necesarios para cumplir los siguientes objetivos:
 - El json contiene un array de Cuentas. Cada cuenta contiene, a su vez, un array de sucursales y un array de beneficios. Cada sucursal tiene un valor "location" que indica la distancia (en metros) hacia la misma. Cada beneficio contiene un valor "type" que indica el valor del descuento. A su vez, las cuentas tienen un array de "tags"
 - Generar un Endpoint para el primer Carrusel del sitio, que permita filtro, ordenamiento y paginado; y que cumpla con los siguientes requerimientos:
 - Solo debe devolver las cuentas con el tag Turismo en Buenos Aires
 - Debe tener un ordenamiento de tal manera que queden siempre primeras aquellas que cuenten con la sucursal más cercana.
 - En base al ordenamiento, debe devolver los primeros 4 elementos.
 - Debemos alimentar las CARDS de Cuentas con los siguientes datos:
 - Nombre de la cuenta
 - Imagen de la cuenta
 - Url hacia la cuenta: <https://club.lanacion.com.ar/{CRMID}> (Aplicado sobre la imagen, y sobre el nombre de la cuenta)
 - Valor de beneficio más alto por cada categoría (Classic, Premium o Black)
 - Distancia a la sucursal más cercana (en metros o, en caso de superar el kilómetro, en kilómetros).
 - Generar un Endpoint para el segundo Carrusel del sitio (o utilizar el mismo que el anterior) que permita filtro, ordenamiento y paginado; el mismo debe cumplir con los siguientes requerimientos:
 - Solo debe tener cuentas que tengan activo el FLAG "haveVoucher".
 - Debe tener un ordenamiento de las cards de forma descendiente. (por nombre alfabéticamente)
 - En base al ordenamiento, debe devolver los primeros 4 elementos.
 - Debemos alimentar las CARDS de Cuentas con los siguientes datos:
 - Nombre de la cuenta
 - Imagen de la cuenta
 - Button de Promocode

- Url hacia la cuenta: `https://club.lanacion.com.ar/{CRMID}`
(Aplicado sobre el button)
- Finalizar entonces con el ensamble completo de la API Creada y del Proyecto Front, alimentado con los datos de esa api y cumpliendo los objetivos provistos.

En caso de no poder resolver el carrusel, se espera encontrar 4 cards estáticas, sin movimiento pero que cumplan con todos los objetivos previstos.

Se espera observar:

Una web app hecha en React from Scratch o bien usar Next.js

Debe cumplir con Server Side Rendering

Repositorio con el código en GitHub

Implementar Hooks

Usar solo React y/o Redux si lo considera necesario para manejo de estados entre componentes, evitando usar dependencias de terceros.

Será un plus si agrega (aunque es opcional)

Uso de programación declarativa

Para manejo de estado entre componentes, se valorará Context API de React .

Implementar Async/Await para la consulta de los datos remotos

Test unitarios o implementar TDD

Usar docker o bien implementar el código resultante en alguno Paas gratuito como heroku o now.sh, etc.

Un layout para responsive (mobile) resuelto a criterio del entrevistado.