

Trabajo Práctico #3

SPA en React utilizando APIs

Objetivo

El objetivo de este trabajo práctico es desarrollar una SPA (Single Page Application) en React consumiendo APIs de terceros.

Requisitos técnicos generales

IMPORTANTE: TODOS puntos deben cumplirse para que el trabajo tenga una nota de aprobación de 6 (seis).

- Posee al menos 5 componentes de React
- Debe tener al menos 2 vistas gestionadas con React Router
- Debe llamar al menos a 2 endpoints distintos de una API
- Debe haber al menos 1 commit de cada integrante del grupo en el log de git
- **Agrupamiento:** grupos de 2 personas
- **Forma de entrega:**
 - Enlace al repositorio en el buzón de entrega de la tarea
 - El repositorio debe poseer un README.md con un enlace a donde se encuentre subida la aplicación para poder probarla

Se puede elegir entre varias opciones para realizar el trabajo práctico:

Opción #1

Aplicación de pronóstico del clima

Una aplicación que permite conocer detalles del clima durante los próximos 7 días para una determinada ciudad.

Requisitos técnicos

- Posee una vista con una entrada de texto que el usuario puede utilizar

para buscar una ciudad por nombre (usar endpoint de búsqueda de ciudades)

- El listado de productos debe el nombre de las ciudades que coinciden con la búsqueda (se puede agregar información adicional que se considere relevante)
- Al hacer clic sobre una ciudad en la lista (o se puede también agregar un botón) se redirige a la vista con los detalles del clima de dicha ciudad
- En la vista de detalle del clima de la ciudad se muestra nombre de la ciudad, fecha actual y una tabla con las temperaturas máximas y mínimas dentro de los próximos 7 días (usar endpoint de pronóstico climático)
- La vista de de detalle del clima de la ciudad recibe el el nombre de la misma y sus coordenadas (latitud y longitud) y utiliza estos para consultar al endpoint de pronóstico climático
- En la vista de detalle del clima de la ciudad se muestra un botón para volver a la vista de búsqueda de ciudades

Power Ups

- (+1) En la vista de detalle del clima de la ciudad se muestra también, para cada uno de los días los valores de sensación térmica y humedad
- (+1) En base a los valores obtenidos de la API, mostrar un ícono al lado de cada día según esté soleado, nublado, ventoso o con lluvia.

NOTA: no se puede obtener una calificación superior a 8 (ocho) mediante esta opción.

Documentación de las APIs REST utilizadas

- El siguiente endpoint permite buscar ciudades por nombre (devuelve una lista) y entre sus datos devuelve las coordenadas de la misma (latitud y longitud) lo cual necesitaremos para llamar a la segunda API

```
curl --request GET \
  --url https://nominatim.openstreetmap.org/search?city=San
Francisco&format=json
```

- El siguiente endpoint obtener el pronóstico climático para un par de coordenadas dadas (en este caso lat=5.26902235 y lon=-72.77388372073908)

```
curl --request GET \
http://api.open-meteo.com/v1/forecast?
latitude=5.26902235&longitude=-72.77388372073908&current=temperatu
re_2m,relative_humidity_2m,weather_code&hourly=temperature_2m&dail
y=weather_code,temperature_2m_max,temperature_2m_min,precipitation
_probability_max
```

Se puede consultar <https://open-meteo.com/en/docs> para conocer qué otros valores se pueden solicitar en la llamada

Opción #2

Aplicación de e-commerce utilizando la API de Mercado Libre

Una aplicación que permite buscar productos, ver sus detalles y, opcionalmente, gestionar un carrito de compras.

Requisitos técnicos

Además de cumplir con todos los puntos especificados en "Requisitos técnicos generales", debe cumplir con lo siguiente:

- Posee una vista con una entrada de texto que el usuario puede utilizar para buscar productos (usar endpoint de búsqueda de productos)
- El listado de productos debe mostrar nombre, precio y foto de cada uno (se puede agregar información adicional que se considere relevante)
- Al hacer clic sobre un producto en la lista (o se puede también agregar un botón "Comprar") se redirige a la vista de detalle del producto
- En la vista de detalle del producto se muestra nombre, precio, foto y descripción (usar endpoint de detalle de producto)
- La vista de detalle de producto recibe el ID del producto especificado dentro de la ruta y utiliza dicho valor para consultar al endpoint de detalle de producto
- En la vista de detalle del producto se muestra un botón para volver a la vista de búsqueda de productos

Power Ups

Siempre y cuando todos los requisitos técnicos se encuentren cumplidos, **la implementación de cada uno de los siguientes puntos proporciona puntos extras sobre la nota de aprobación.**

- (+1) En la vista de detalle del producto, si el mismo tiene más de una foto, las mismas se muestran mediante un carousel

- (+1) En la vista de búsqueda de productos se puede filtrar por categorías
- (+1) En la vista de detalle del producto se muestra una tabla con los atributos del mismo
- (+1) En la vista de detalle del producto el botón comprar funciona! La aplicación debe mantener un registro de los productos que el usuario desea comprar mostrando el icono de un carrito de compras en la parte superior que indique la cantidad de productos en el carrito
- (+1) Al hacer clic en el icono del carrito se redirige al usuario a una nueva vista donde se muestra una lista (nombre, foto, precio) de los elementos que están en el carrito y *el precio total de la compra*
- (+1) En la vista correspondiente a los elementos del carrito de compras hay un botón que permite eliminar elementos del carrito

Documentación de la API REST de Mercado Libre

- Búsqueda de productos (ej: teclado):

```
curl --request GET \
  --url https://api.mercadolibre.com/sites/MLA/search?q=teclado
```

- Detalles de un producto (id de producto: MLA1377454181)

```
curl --request GET \
  --url https://api.mercadolibre.com/items/MLA1377454181
```

Opción #3

Aplicación de música utilizando la API de Spotify

Una aplicación que permite buscar artistas por nombre y ver sus álbumes y canciones.

Requisitos técnicos

- Posee una vista con una entrada de texto que el usuario puede utilizar para buscar artistas (usar endpoint de búsqueda de artistas)
- El listado de artistas debe mostrar foto y nombre del artista
- Al hacer clic sobre un artista de la lista se redirige a la vista de detalle del artista
- En la vista de detalle del artista se muestra nombre y foto del artista
- La vista de detalle de artista recibe el ID del artista especificado dentro de la ruta y utiliza dicho valor para consultar al endpoint de álbumes del artista, con dicha información muestra nombre, foto de portada y año de cada uno de los álbumes

Power Ups

- (+1) En la vista de detalle del producto se muestra un botón para volver a la vista de búsqueda de artistas
- (+1) En la vista de detalles de artista, se lo puede marcar como artista favorito, los artistas favoritos aparecen listados al costado en la vista de búsqueda de artistas y al hacer clic sobre uno conduce a la la vista de detalle para ese artista
- (+1) La seguridad ante todo: las credenciales (CLIENT_ID y CLIENT_SECRET) no deberían estar "hardcodeadas" dentro del código de la aplicación, crear una vista tipo pantalla de login que permita almacenarlas en el local storage del navegador y tomarlas de allí para pedir el token de autenticación
- (+1) Existe una vista para ver los detalles de un album, en ella se muestra nombre del artista, nombre del album y lista de temas, la vista recibe el ID del album especificado dentro de la ruta y utiliza dicho valor para consultar al endpoint de detalles del album
- (+1) En la vista de detalle del album se muestra un botón para volver a la vista de detalle de ese artista
- (+1) En la vista de detalle del album se pueden marcar canciones como favoritas, existe una vista adicional para ver todas las canciones marcadas como favoritas, en la misma se muestra el nombre de la canción, del artista y el album, al hacer clic sobre la canción lleva a la vista de detalle del album

Documentación de la API REST de Spotify

1. Es necesario tener una cuenta (free o premium) de Spotify
2. Crear una aplicación (para obtener un CLIENT_ID y CLIENT_SECRET) en <https://developer.spotify.com/dashboard/create>
 - En App name y App description se puede poner cualquier cosa
 - En Redirect URIs se puede poner cualquier cosa, pero tiene que ser una URL válida
 - **IMPORTANTE:** tiene que estar seleccionada la casilla Web API

The screenshot shows the 'Create app' page on the Spotify for Developers dashboard. The page has a dark header with the Spotify logo and 'for Developers' text, and links for 'Documentation' and 'Community'. Below the header, a breadcrumb trail reads 'Dashboard > Create app'. The main heading is 'Create app'. There are three input fields: 'App name *', 'App description *', and 'Website'. Each field is a simple text box with a light gray border and a small asterisk indicating it is required.

3. Se puede ver las apps creadas en <https://developer.spotify.com/dashboard>, seleccionar la app y hacer clic en el botón settings (arriba a la derecha)
4. Ahí podemos encontrar los valores para el CLIENT_ID y CLIENT_SECRET

The screenshot shows the 'Basic Information' page for a specific app on the Spotify for Developers dashboard. The header is the same as the previous screenshot. The breadcrumb trail is 'Dashboard > Prueba > Basic Information'. The main heading is 'Basic Information' with a pink square icon containing a white 'P'. Below the heading are three tabs: 'Basic Information' (selected), 'User Management', and 'Extension Requests'. The 'Basic Information' tab shows two main sections. The first section is 'Client ID' with a value of '57bf76bf54eb4fe0b3b4decda341e80d' and a 'Copy' icon. To the right of this is 'App Status' with a value of 'Development mode'. Below the 'Client ID' is a link 'View client secret'. The second section is 'App name' with a text box.

5. **IMPORTANTE:** todas las llamadas a la API de Spotify deben ser autenticadas mediante un token, el mismo se obtiene de una llamada a un endpoint específico de la API
- Obtener token de autenticación (reemplazar \$CLIENT_ID y \$CLIENT_SECRET por los valores de nuestra app)

```
curl -X POST "https://accounts.spotify.com/api/token" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d
"grant_type=client_credentials&client_id=$CLIENT_ID&client_secret=
$CLIENT_SECRET"
```

- Obtener lista de artistas (ej: metallica, reemplazar 12345678 por el token obtenido en el paso anterior)

```
curl --request GET \
  --url 'https://api.spotify.com/v1/search?
q=metallica&type=artist' \
  --header 'Authorization: Bearer 12345678'
```

- Obtener álbumes de un artista (reemplazar \$ID_ARTIST por el id del artista, reemplazar 12345678 por el token obtenido en el paso anterior)

```
curl --request GET \
  --url 'https://api.spotify.com/v1/artists/$ID_ARTIST/albums' \
  --header 'Authorization: Bearer 12345678'
```

- Obtener detalles de un álbum (reemplazar \$ID_ALBUM por el id del álbum, reemplazar 12345678 por el token obtenido en el paso anterior)

```
curl --request GET \
  --url 'https://api.spotify.com/v1/albums/$ID_ALBUM' \
  --header 'Authorization: Bearer 12345678'
```

Documentación general de la API:

<https://developer.spotify.com/documentation/web-api/tutorials/getting-started>