

# CHALLENGE JAVASCRIPT - ALKEMY LABS

## Objetivo

---

Crear una aplicación web similar a [Google Play Store](#), es decir, un catálogo online donde los usuarios puedan ver un listado de Apps, información adicional de una app individual y un carrito y/o lista de deseos donde sea posible agregarlas.

## Requerimientos técnicos

---

Deberás desarrollar una API en Node.js junto a cualquiera de los siguientes frameworks, en sus versiones estables:

- [Express](#)
- [Adonis](#)
- [Koa](#)

En el caso de querer utilizar otro framework, por favor consultarlo a [contacto@alkemy.org](mailto:contacto@alkemy.org)

Los datos mostrados en la Aplicación deben ser persistidos en una base de datos relacional. El esquema de datos puede armarse según se considere apropiado en base a los requerimientos del Challenge.

La API deberá exponer URLs que devuelvan datos en JSON, para ser consumidos desde el Frontend.

## Requerimientos Funcionales

---

En la aplicación deberán existir dos tipos de usuarios: Desarrollador y Cliente.

### Registro

Tanto desarrollador como cliente se registrarán a través del mismo formulario, especificando su rol en el mismo.

## Usuario Desarrollador

El usuario registrado como desarrollador puede realizar las siguientes acciones:

- Dar de alta una aplicación indicando:
  - Categoría.
  - Nombre.
  - Precio de venta,
  - Imagen o logo.
- Editar los datos de una aplicación. (La categoría y el nombre no pueden ser modificados).
- Dar de baja una aplicación.

## Usuario Cliente

El usuario registrado como cliente puede realizar las siguientes acciones:

- Listar todas las categorías de apps.
- Ingresar a una categoría y ver las apps de esa categoría.
- Ingresar al detalle de una app.

## Javascript Frontend

El frontend deberá ser desarrollado en React.js, y realizar llamadas a la API desarrollada, utilizando fetch o alguna librería para realizar peticiones de forma asíncrona.

Cuando el usuario cliente clickee en "Comprar" deberás enviar por AJAX el id de la app y guardarla en la base de datos. Para esto, deberás crear un endpoint que se acceda de la siguiente forma: <https://urlServidor/api/buy>

Esa URL solo se puede invocar por:

- POST si es para comprar o agregar a la lista de deseos.
- DELETE si es para cancelar la compra o eliminar de la lista de deseos.

En el body del request se debe comunicar solamente el id de la app a comprar.

## Rutas y Seguridad

Todas las rutas que solo puedan navegar usuarios autenticados, deben estar bajo el prefijo /me. Por ejemplo:

En la ruta /me/apps, si el usuario logueado es el desarrollador, verá las apps que creó, mientras que si es el usuario cliente, verá las apps que compró. En cambio, si ingresa a/apps (sin el prefijo) va a ver todas las apps sin distinción.

Si un usuario no autenticado intenta ingresar a una ruta que no le corresponde, por ejemplo a la url para crear aplicaciones, entonces debe ser redirigido a donde vino.

## Conclusión

---

Esperamos contar con una pantalla de registro y una de login. Luego, ingresando al sistema como desarrollador debo ver la lista de las aplicaciones que publiqué, ingresar al detalle de cada una, editar información o eliminarla. Si ingreso como usuario cliente, debo ver la lista de aplicaciones que compré y ver el detalle.

## Criterios a evaluar

---

- Diseño responsive, moderno, intuitivo.
- Puede ser algo minimalista, sencillo, pero funcional. Se puede usar cualquier framework CSS: Bootstrap, Materialize, Bulma.
- Conocimientos básicos/intermedios de NodeJs.
- Validación de los formularios.
- Protección de rutas según el rol del usuario.
- Buenas prácticas de codificación.
- Buenas prácticas para nombre de rutas.
  
- Uso correcto de verbos HTTP.
- Correcto diseño de la base de datos.

## Bonus

---

Se requiere que implementes al menos uno de estos puntos (a elección). Mientras más y mejor implementes, mejor score obtendrás.

Estos son algunos puntos extra que puedes implementar:

- En listado de categorías.
- Que aparezcan en orden alfabético.
- Mostrar la cantidad de apps que tiene relacionadas.
- En el listado de apps.
- Se debe paginar cada 10 resultados.
- Por defecto deben aparecer en el orden de creación, de más reciente a más antigua.
- En el detalle de la app.
- Mostrar un botón para "Comprar!" y otro para "Agregar a la lista de deseos": Estos dos botones solo deben aparecer en caso de que el visitante no sea el mismo usuario desarrollador de la app.
- Si un usuario ya compró la app no puede volver a comprarla y si ya la agregó a la lista de deseos, tampoco.
- Cuando el usuario developer edita el precio de una app debe quedar guardado el registro histórico de todos los precios de la aplicación, incluyendo la fecha de cuando se produjo el cambio.
- Desarrollo de formulario para ABM de Categorías.

¡Mucha suerte!

**Equipo de Admisiones de Alkemy.**