

CHALLENGE PHP - ALKEMY LABS

Requerimientos

Deberás crear una aplicación en PHP utilizando un framework.

El objetivo es simular una aplicación web como el PlayStore de Google: un mercado de aplicaciones donde los usuarios puedan listar apps, ver información adicional y agregarlas a un carrito y/o lista de deseos.

Frameworks

Puedes desarrollar la aplicación en cualquiera de los frameworks dados a continuación, y siempre en la última versión o las dos más recientes.

- [Laravel](#) - versión v6 || v7
- [Symfony](#) - v4 || v5
- [Yii](#) - v2
- [Cake PHP](#) - v4
- Otros frameworks - Consultar

Base de datos

Leyendo los requerimientos, deberás armar la base de datos que consideres apropiada para que todo funcione correctamente. El tipo de base de datos debe ser relacional, no importa que sea MySQL, Maria DB o Postgres. Todos los nombres de tablas, columnas, índices deben estar en inglés y usar underscore para separar palabras.

Registro

En la aplicación hay dos tipos de usuarios: desarrollador y cliente. Ambos se registran a través del mismo formulario, pero deben especificar si son desarrolladores o clientes.

Usuario Developer

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

Necesitamos que integres JavaScript en el front pero solo para hacer llamadas mediante AJAX. Puedes utilizar el metodo `fetch` o alguna librería como `axios` (deseable), o si no usar el `ajax` de `jQuery`. 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://localhost:3000/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.

Importante: En el body del request se debe comunicar solamente el `app_id`

Rutas y Seguridad

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

En la ruta `http://localhost:3000/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 `http://localhost:3000/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 quiero ver la lista de las aplicaciones que publiqué, ingresar al detalle de cada una, editar información o eliminarla. Sin embargo, si ingreso como usuario cliente, quiero ver la lista de aplicaciones que compré y ver el detalle. Los usuarios que no están registrados ni logueados pueden recorrer las categorías, ver las apps, el detalle y agregarla para comprar.

Criterios a evaluar

- Diseño responsive, moderno, intuitivo:
 - Puede ser algo minimalista, sencillo, pero funcional.
 - Se puede usar cualquier framework CSS: Bootstrap, Materialize, Bulma.
- Conocimientos generales de PHP.
- Conocimientos básicos/intermedios de Laravel:
 - Correcto uso de los controllers.
 - Correcto uso de los modelos, relaciones, atributos.
- Validación de todos los formularios.
- Seguridad:
 - Que usuarios developers no ingresen a rutas de usuarios comunes.
- Buenas prácticas de codificación.
- Buenas prácticas para nombre de rutas.
- Uso correcto de verbos HTTP.
- Correcto diseño de la base de datos.
- Optimización de las tablas.

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 mas reciente a más antigua.
 - Agregar una opción de ver solo las mas votadas y al listarlas que aparezcan en el orden de la mas votada a la menos, pero las que tengan puntaje por debajo de 2 no deben aparecer.
- En el detalle de la app:
 - Aquí deben aparecer dos botones, uno que diga "Comprar!" y otro que diga "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.
- Realizar el armado de la base de datos a través de **migrations**.
- Agregar seeders para popular la base de datos.
- 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.

¡Mucha suerte!

Equipo de Admisiones de Alkemy.