

Introducción al Desarrollo Web/Móvil

Cátedra II

Cliente Web y API

Listado de productos

Descripción:

En la primera cátedra de la asignatura, usted creó una pantalla con HTML y CSS para mostrar un listado de productos. Sin embargo, los datos de los productos como los nombres, imágenes, precio, y descripción eran estáticos. Ahora se necesita que esos productos sean administrables. Que los datos sean dinámicos según la necesidad del negocio. Para la presente evaluación usted debe crear el CRUD completo con backend y frontend para los productos. **El api debe contener una migración automática para crear la base de datos con la tabla de productos.**

Backend:

La API debe tener los siguientes endpoints:

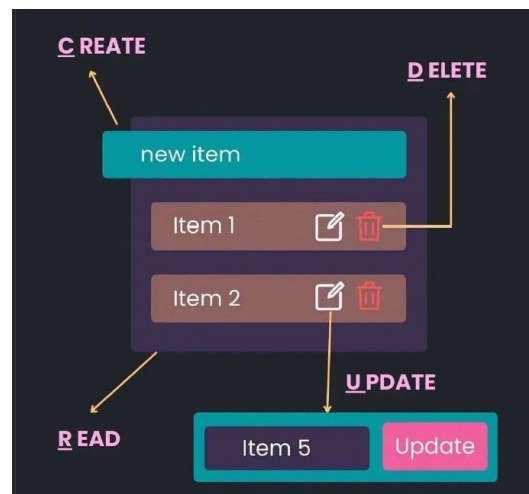
1. URL `"/api/products"` (método GET): Este endpoint debe retornar un listado de productos en formato JSON. Cada producto debe tener al menos los siguientes datos para mostrar en el frontend:
 - a. Name: Nombre del producto.
 - b. Price: Precio del producto.
 - c. Description: Descripción del producto.
 - d. Image: Imagen del producto.
2. URL `"/api/products/"` (método POST): Este endpoint agrega un nuevo producto al listado de productos los campos que debe tener cada producto son los siguientes:
 - a. Name: Nombre del producto (campo requerido).
 - b. Price: Precio del producto (Campo requerido y de tipo número).
 - c. Description: Descripción del producto.
 - d. Image: Imagen del producto (BLOB o URL) (campo requerido).
3. URL `"/api/products/{id}"` (método PUT): Este endpoint permite actualizar todos los campos de un producto. Es decir, el usuario debe poder editar cualquier producto que se liste en pantalla.
4. URL `"/api/products/{id}"` (método DELETE): Este endpoint eliminar un producto existente.

Frontend:

La aplicación web debe visualizar el listado de productos en la URL “/products”, además debe contener lo siguiente:

1. Un listado de productos en forma de tabla, card, list o la manera que usted encuentre mejor para mostrarlos. Cada producto debe tener un botón de eliminar y editar.
2. Se debe presentar en la parte superior un botón de agregar. Cuando se presiona este botón agregar se debe mostrar en la misma pantalla o en un modal un formulario para agregar un nuevo registro.
3. El botón de editar debe permitirle editar todos los campos de un registro seleccionado, consumiendo el endpoint generado en el API.
4. EL botón de eliminar debe permitirle al usuario eliminar un registro.

A continuación, se muestra una imagen referencial de como los botones que debe tener cada producto.



Consideraciones:

- Se debe usar GitHub para versionar su trabajo, debe ser solo un repositorio para el cliente web y la API. **No se permiten commits después de la fecha de entrega.**
- Se debe desarrollar el API utilizando el framework o librería como laravel, .NET, Spring, etc.
- Se debe desarrollar la interfaz de usuario utilizando el framework o librería web escogido como: React, Angular, Vue, Svelte, etc. Además, se debe integrar un framework o biblioteca de CSS como Bootstrap o Tailwind CSS para el diseño.

- Puede usar cualquier recurso disponible. Internet, ChatGPT, Plugins de editores, proyectos realizados en clase, etc.
- La aplicación de frontend y de backend deben ser nuevas. Es decir, se debe crear la evaluación a partir de un proyecto nuevo de backend y un proyecto nuevo de frontend. No se puede existir desarrollos en las aplicaciones que no correspondan a la evaluación.
- El primer commit de su aplicación debe ser a partir de las 14h30 y el último commit máximo hasta las 16h30.
- La aplicación debe tener un diseño agradable para la vista de un usuario, debe tener un diseño elegante y vistoso. **Si se permite el uso de templates de terceros.**

Entregables:

La entrega de la práctica debe contener:

1. Ambos componentes, cliente web y API en un mismo repositorio de **GitHub**.
2. Archivo **README.md** en la raíz del proyecto con las instrucciones para levantar los proyectos de web y de la API. Estas instrucciones deben estar dirigidas a un usuario que **no tiene las tecnologías ocupadas instaladas** en su máquina personal. **Si no se puede levantar localmente la aplicación la evaluación no será calificada obteniendo la nota mínima**

Imagen referencial de como se debe mostrar su interfaz gráfica. El diseño es personal. Usted puede hacer en tabla, lista, card, etc. El botón View No es necesario por cada item. Únicamente editar y eliminar. Además, se debe mostrar una imagen por cada uno de los productos.

Item Management

ID	Name	Price	Description	Action
4	Banana	50	Yummy	View Edit Delete
5	Straberry	600	Red and Sweet	View Edit Delete
8	Melon	56	Refreshing	View Edit Delete
10	Avocado	12	Loaded with fiber	View Edit Delete

[ADD ITEM](#)

Rúbrica de evaluación:

- La Tabla I corresponde a la rúbrica de la evaluación.

Tabla I: Rúbrica de evaluación.

Componente	Descripción		Puntaje
Uso de Git	El proyecto se encuentra versionado con Git en GitHub o BitBucket. El proyecto se encuentra en un único repositorio de GitHub que contiene tanto el cliente web como la API. El proyecto debe ser público con un archivo README.md que evidencia los pasos a seguir para ejecutar localmente su aplicación.		5
Endpoint GET "/api/products" (BACKEND)	Un controlador atiende la solicitud de la URL "/api/products" con el método GET. Este método debe poder retornar en un formato JSON el listado de productos.		10
Endpoint POST "/api/products" (BACKEND)	Un controlador atiende la solicitud de la URL "/api/products" con el método POST. Para agregar un nuevo registro.		10
Endpoint PUT "/api/products/{id}" (BACKEND)	Un controlador atiende la solicitud de la URL "/api/products/{id}" con el método PUT. Para editar un registro.		10

Endpoint DELETE “/api/products/{id}” (BACKEND)	Un controlador atiende la solicitud de la URL “/api/products/{id}” con el método DELETE. Para eliminar un registro.	10
Visualización del listado de productos (FRONTEND)	La aplicación web visualiza el listado de productos cada producto listado debe mostrar todos los campos de cada producto.	10
Formulario para agregar un producto (FRONTEND)	La aplicación web presenta un formulario para agregar un nuevo producto a la base de datos mediante el ENDPOINT para agregar del backend. Se debe enviar todos los campos de cada producto.	10
Formulario para actualizar un producto (FRONTEND)	La aplicación web presenta un formulario para editar los datos un producto en la base de datos mediante el ENDPOINT para actualizar un registro mediante el API. Se debe enviar todos los campos de cada producto.	10
Formulario para eliminar un producto (FRONTEND)	El usuario puede eliminar mediante la interfaz gráfica de la aplicación Frontend un producto dando clic en el botón eliminar de cada producto.	10
Responsividad	La aplicación web tiene un diseño agradable, elegante y vistoso, que se adapta a diferentes dispositivos y tamaños de pantalla (responsividad).	15
Total		100

Good Luck! 😊 🍀