

Supermercado Megamaxi

Desarrollar una Aplicación Web con **dos** páginas navegables mediante un NavBar:

- Crear nuevo Producto (Página Web con **un componente vue**)
- Consultar Productos (Página Web con **dos componentes vue**)

Desarrollar las siguientes funcionalidades:

P1: Página "Crear nuevo Producto"

Página Web que contiene un componente con funcionalidad que permitirá crear un nuevo Producto (Botón **Guardar**) con la siguiente información:


- Código de barras
- Nombre
- Stock (cantidad de unidades que existen en bodega)
- Fecha de caducidad (se debe usar el elemento adecuado para manejo de fechas)

Nota: Luego de que se guarde el Producto, se debe presentar nuevamente la página web vacía para ingresar un nuevo Producto y un mensaje informativo indicando que se ha guardado exitosamente el Producto.

P2: Página "Consultar Productos" (componente 1)

Componente 1: Debe tener un botón que permita consultar (Botón **Consultar**) todos los Productos ingresados en la aplicación, mismos que se presentaran en una **lista** en el mismo componente 1; a continuación, una ilustración del **componente 1**:

Componente 1



Código de barras	Nombre	Acción	Acción
121214554	Coca Cola	Actualizar	Eliminar
...	...	Actualizar	Eliminar
...	...	Actualizar	Eliminar

- Botón Actualizar:** Cada Producto de la lista debe tener un botón "**Actualizar**", que al dar click se comunica con el **componente 2** (funcionalidad y componente se describe en P3).
- Botón Eliminar:** Cada Producto de la lista debe tener un botón "**Eliminar**" que al dar click permita eliminar el Producto, y se refresque la lista de Productos.

Nota: Al eliminar un Producto se debe presentar un mensaje informativo indicando que se ha eliminado exitosamente el Producto.

Este **componente 1** debe consumir una capacidad de un API que retorne una lista de Productos con únicamente los siguientes atributos:

- Código de barras
- Nombre
- Link Hipermedia (link a si mismo)

Nota: El link Hypermedia, es un link a si mismo que permite consultar todos los atributos completos del Producto (los 4 atributos de **P1** y que servirán para llenar el **componente 2**) a partir de su código de barras.

P3: Página “Consultar Productos” (componente 2)

Componente 2: Componente que recibe la comunicación del **componente 1** (cuando da click en el botón Actualizar en la lista de Productos) y permite visualizar la información completa de un Producto, a continuación, una ilustración del **componente 2**:

Componente 2

Código de barras

Nombre

Stock

Fecha de caducidad

Actualizar

Nota: La página Web “Consultar Productos” con sus **dos componentes** se ve de acuerdo con la siguiente ilustración:

Componente 1

Consultar			
Código de barras	Nombre	Acción	
121214554	Coca Cola	Actualizar	Eliminar
...	...	Actualizar	Eliminar
...	...	Actualizar	Eliminar

Se consume el link hypermedia, que permite consultar todos los atributos del Producto que permitirá llenar las cajas de texto del componente 2

Componente 2

Código de barras

Nombre

Stock

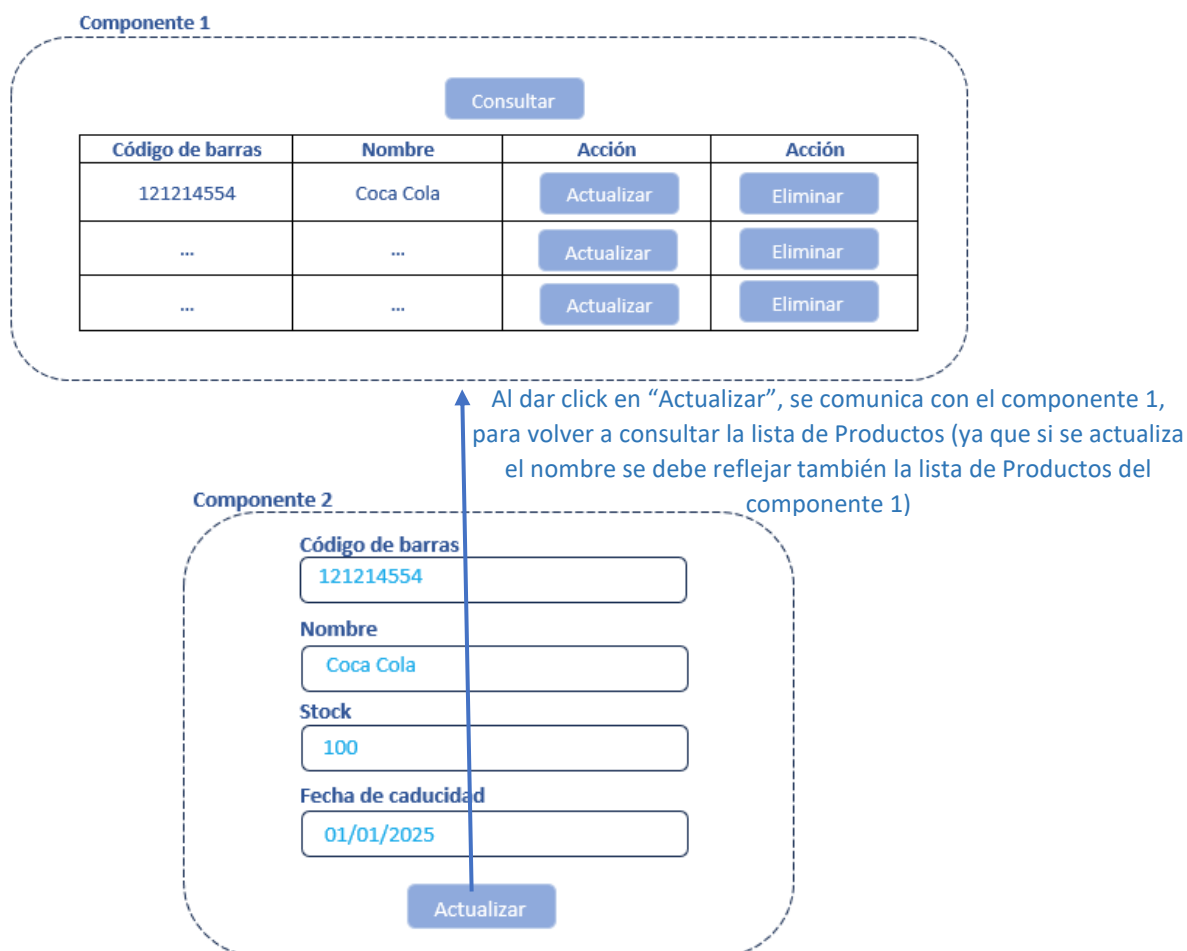
Fecha de caducidad

Actualizar

P4: Página “Consultar Productos” (componente 2) (botón Actualizar)

Botón Actualizar: Las cajas de texto del componente 2, deben permitir modificar sus valores, a excepción del código de barras, y al dar click en el botón “Actualizar” se debe actualizar los datos en la aplicación web.

Adicional al dar click en el botón “Actualizar”, el componente 2 se debe comunicar con el componente 1, para volver a consultar todos los productos ingresados en la aplicación, a continuación, se muestra una ilustración de la comunicación:



Consideraciones:

1. Implementar un API RestFul **Inventario** con un servicio: Producto y sus respectivas capacidades.
2. Implementar el FrontEnd con Vue.JS y que consuma el API RestFul de **Inventario**.

Notas generales:

1. En todas las páginas web usar estilos CSS de Bootstrap, que permita al usuario tener una experiencia agradable al interactuar con las páginas.
2. Subir todo el código desarrollado a nuevos repositorios de GitHub (2 repositorios) creados exclusivamente para la evaluación.
3. Realizar una captura de cada página web funcionando e incluir su respectiva URL en texto en su entrega.
4. Aplicar los siguientes conceptos: VUE.js, Spring Boot, RESTful, Modelo de Madurez de Richardson, Clientes RESTful y demás conceptos revisados en clase.
5. La evaluación es **individual**, de existir **sospecha** de copia se anula la nota completa del examen, tampoco se puede copiar código de internet ni de otros talleres, el código del examen debe ser desarrollado.