Laboratorio N° 7. Construción de un cliente 'grueso' con un API REST, HTML5, Javascript y CSS3. Parte II.

Santiago Cárdenas Amaya

Juan Pablo Fonseca

Escuela Colombiana de ingeniería Julio Garavito

Bogotá 2023

Introducción

Los CRUD son de las operaciones más comunes y a la vez mas importantes cuando de implementar aplicaciones de cara al usuario se trata. Estas operaciones permiten que el cliente puede tener una gran capacidad de manipulación y consulta de los datos de un servicio.

A lo largo de este laboratorio se seguirá con el proyecto del laboratorio que precede a este, pero implementando más operaciones, añadiendo detalles que harán que conozcamos más sobre estas tecnologías.

# Desarrollo del laboratorio

Para empezar, se separó el front del proyecto en el que estaba, por lo cual tocó solucionar distintas cosas tales como:

* Problemas de CORS policy
* Importaciones de Bootstrap y JQuery

1. Agregue al canvas de la página un manejador de eventos que permita capturar los 'clicks' realizados, bien sea a través del mouse, o a través de una pantalla táctil. Para esto, tenga en cuenta este ejemplo de uso de los eventos de tipo 'PointerEvent' (aún no soportado por todos los navegadores) para este fin. Recuerde que a diferencia del ejemplo anterior (donde el código JS está incrustado en la vista), se espera tener la inicialización de los manejadores de eventos correctamente modularizado, tal como se muestra en este codepen.



1. Agregue lo que haga falta en sus módulos para que cuando se capturen nuevos puntos en el canvas abierto (si no se ha seleccionado un canvas NO se debe hacer nada):



1. Agregue el botón Save/Update. Respetando la arquitectura de módulos actual del cliente, haga que al oprimirse el botón:
   1. Se haga PUT al API, con el plano actualizado, en su recurso REST correspondiente.
   2. Se haga GET al recurso /blueprints, para obtener de nuevo todos los planos realizados.
   3. Se calculen nuevamente los puntos totales del usuario.



(Como en el lab anterior toca darle dos veces al boton de “Save/Update” para que haga la funcion de save)

1. Agregue el botón 'Create new blueprint', de manera que cuando se oprima:
   1. Se borre el canvas actual.
   2. Se solicite el nombre del nuevo 'blueprint' (usted decide la manera de hacerlo).

Estado inicial:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al clickear “Create new blueprint”:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Finalmente al clickear “create”:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Agregue el botón 'DELETE', de manera que (también con promesas):
   1. Borre el canvas.
   2. Haga DELETE del recurso correspondiente.
   3. Haga GET de los planos ahora disponibles.

Estado inicial:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al clickear en “DELETE”:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Se adjunta la URL del repositorio:

<https://dev.azure.com/juanpablofonsecac5/arsw/_git/ARSW>

# Conclusiones

Se puede concluir que la aplicación de mayores funcionalidades a un sitio le da “mas vida” y sentido a los desarrollos; en paralelo, la generación de servicios API facilitan muchísimo el desarrollo del front, puesto que así este solo se encarga de mostrar y transmitir datos.