**CONSTRUCIÓN DE UN CLIENTE 'GRUESO' CON UN API REST, HTML5, JAVASCRIPT Y CSS3. PARTE I.**

**AUTORES:**

YORKS GOMEZ – CESAR VÁSQUEZ

# PROFESOR:

JAVIER IVAN TOQUICA

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

ARQUITECTURAS DE SOTFWARE - GRUPO # 1

INGENIERÍA DE SISTEMAS

BOGOTÁ D.C.

2023

# INTRODUCCIÓN

# DESARROLLO

# Incluimos dentro de las dependencias de Maven los 'webjars' de jQuery y Bootstrap (lo cual nos permite tener localmente dichas librerías de JavaScript al momento de construír el proyecto)

# 

# Front-End – Vistas

# Creamos el directorio donde residirá la aplicación JavaScript. Como se está usando SpringBoot, la ruta para poner en el mismo contenido estático (páginas Web estáticas, aplicaciones HTML5/JS, etc) es: src/main/resources/static

# 

# Creamos, en el directorio anterior, la página index.html, sólo con lo básico: título, campo para la captura del autor, botón de 'Get blueprints', campo donde se mostrará el nombre del autor seleccionado, la tabla HTML donde se mostrará el listado de planos (con sólo los encabezados), y un campo donde se mostrará el total de puntos de los planos del autor. Recuerde asociarles identificadores a dichos componentes para facilitar su búsqueda mediante selectores.

# 

# Interfaz de usuario gráfica, Aplicación Descripción generada automáticamente

# Front-End – Lógica

# Añadimos más planos a los autores johnconnor y maryweyland

# 

# Agregamos la importación de los dos nuevos módulos de la pagina html

# 

# Agregamos al módulo 'app.js' una operación pública que permita actualizar el listado de los planos, a partir del nombre de su autor (dado como parámetro). Para hacer esto, dicha operación debe invocar la operación 'getBlueprintsByAuthor' del módulo 'apimock' provisto, enviándole como callback esta función:

# 

# Tomamos el listado de los planos, y le aplicamos una función 'map' que convierta sus elementos a objetos con sólo el nombre y el número de puntos.

# 

# Sobre el listado resultante, haga otro 'map', que tome cada uno de estos elementos, y a través de jQuery agregue un elemento <tr> (con los respectvos <td>) a la tabla creada anteriormente.

# 

# Sobre cualquiera de los dos listados (el original, o el transformado mediante 'map'), aplicamos un 'reduce' que calcule el número de puntos. Con este valor, use jQuery para actualizar el campo correspondiente dentro del DOM.

# 

# Asociamos la operación antes creada (la de app.js) al evento 'on-click' del botón de consulta de la página. Lo realizamos en el index.html

# Texto Descripción generada automáticamente

# Verificamos el funcionamiento de la aplicación

# Interfaz de usuario gráfica, Texto Descripción generada automáticamente

# 

# DIAGRAMA DE COMPONENTES

# CONCLUSIONES

# BIBLIOGRAFÍA