**Entregable numero 1**

Para este siguiente entregable, me propuse crear esta documentación para tener una mejor resolución sobre las definiciones y módulos que instalaré a lo largo de este proyecto

Primero que nada, me gustaría definir las partes de un MVC (modelo-vista-controlador), patrón de diseño que propone separar nuestro código por responsabilidad.

Graphical user interface, diagram

Description automatically generated

Aclarar que para nuestro proyecto no necesitamos de las vistas, haremos uso solamente del controlador y el modelo consumiendo un api externa.

**Modelo**: Es el encargado de manejar la información de nuestra aplicación (por ejemplo, se encarga de guardar información en un archivo).

El trabajo del modelo es simplemente administrar los datos. Ya sea que los datos provengan de una base de datos, una API o un objeto JSON, el modelo es responsable de administrarlos.

En nuestro proyecto, tomamos la api “[fakestoreapi.com”](https://fakestoreapi.com/) contiene un arreglo de objetos producto, carrito y usuario con toda la información (datos) necesario para la aplicación

**Vista:** El trabajo de la vista es decidir qué verá el usuario en su pantalla y cómo.

**Controlador**: Es el encargado de unir el modelo y la vista, y también posee la lógica para transformar los datos para que los entienda tanto el modelo como la vista.

La responsabilidad del controlador es extraer, modificar y proporcionar datos al usuario. Esencialmente, el controlador es el enlace entre  y el modelo.

**Conclusión**

El concepto más atractivo del patrón MVC es la separación de preocupaciones.

Las aplicaciones web modernas son muy complejas, y hacer un cambio a veces puede ser un gran dolor de cabeza.

Administrar el frontend y el backend en componentes separados más pequeños permite que la aplicación sea escalable, mantenible y fácil de expandir.

**NPM instaladas del proyecto:**

1)

**Express:**

Express es un marco de aplicación web de Node.js mínimo y flexible que proporciona un conjunto sólido de funciones para desarrollar aplicaciones web y móviles. Facilita el desarrollo rápido de aplicaciones web basadas en node. Las siguientes son algunas de las características principales del marco Express:

1. Permite configurar middlewares para responder a solicitudes HTTP.
2. Define una tabla de enrutamiento que se utiliza para realizar diferentes acciones según el método HTTP y la URL.
3. Permite representar dinámicamente páginas HTML en función de pasar argumentos a plantillas.

Comando instalación del paquete:

npm install express --save

2)

**Nodemon**:

es una utilidad que monitorea los cambios en el código fuente que se esta desarrollando y automáticamente re inicia el servidor. Es una herramienta muy útil para desarrollo de aplicaciones en node js.

Comando instalación del paquete:

npm install nodemon --save-dev

Configuración:

Agregar el comando “start” en el archivo package.json

Text

Description automatically generated

Ejecutar en la terminal, con el comando:

npm run start

3)

**Fetch:**

se usa para el acceso a recursos del servidor de manera asíncrona, basado en promesas. Es básicamente la nueva interfaz para realizar funcionalidades Ajax con Javascript, que ya podemos usar para facilitar la organización del código en nuestras aplicaciones.

El método fetch() depende directamente del objeto window del navegador. Su uso más simple consiste en pasarle una URL, cuyo contenido se traerá el cliente web de manera asíncrona.

Comando instalación:

npm install node-fetch --save

4)

**Morgan**:

Es un Middleware de nivel de solicitud HTTP. Es una gran herramienta que registra las requests junto con alguna otra información dependiendo de su configuración y el valor predeterminado utilizado. Demuestra ser muy útil durante la depuración y también si desea crear archivos de registro.

Comando instalación del paquete:

npm install morgan --save