

Introducción

Express es un framework web transigente, escrito en JavaScript y alojado dentro del entorno de ejecución NodeJS. El módulo explica algunos de los beneficios clave de este framework, como configurar tu entorno de desarrollo y cómo realizar tareas comunes en desarrollo y publicación web.



¿Cuáles son los principales objetivos de este proyecto?

- Manejar Node js
- Entender cómo interactuar con Express js
- Aprender a manejar Express js
- Manejar NPM

1.1.Ejercicios

- Ejercicio 1
 - Crear un archivo con el nombre e1.js
 - Levantar un servidor utilizando Express, al levantar el servidor tiene que mostrar un mensaje que diga:
 - `Servidor levantado en el puerto \${puerto}`.
- Ejercicio 2
 - Crear un archivo con el nombre ej2.js
 - Levantar un servidor de Express
 - Manejar las siguientes rutas:
 - Ruta: Raíz del sitio ('/') ,Metodo: get, Acción: Mostrar un mensaje de bienvenida
 - Ruta: Productos, Método: get, Acción: Mostrar un mensaje que diga: listado de productos
 - Ruta: Productos, Método: post, Acción: Mostrar un mensaje que diga: crear un producto
 - Ruta: Productos, Método: put, Acción: Mostrar un mensaje que diga: actualizar un producto
 - Ruta: Productos, Método: delete, Acción: Mostrar un mensaje que diga: borrar un producto
 - Ruta: Usuarios, Metodo: get, Acción: Mostrar un mensaje que diga: listado de usuarios
 - Ruta: Usuarios, Método: post, Acción: Mostrar un mensaje que diga: crear un usuario
 - Ruta: Usuarios, Metodo: put, Acción: Mostrar un mensaje que diga: actualizar un usuario
 - Ruta: Usuarios, Metodo: delete, Acción: Mostrar un mensaje que diga: borrar un usuario
 - Utilizar Postman para probar todos los llamados

- Ejercicio 3
 - Crear un archivo con el nombre ej3.js
 - Levantar un servidor de Express
 - Al llamar a localhost:3000/products se debe mostrar el siguiente JSON:


```

{
  description: 'Productos',
  items: [
    { id: 1, nombre: 'Taza de Harry Potter' , precio: 300},
    { id: 2, nombre: 'FIFA 23 PS5' , precio: 1000},
    { id: 3, nombre: 'Figura Goku Super Saiyan' , precio: 100},
    { id: 4, nombre: 'Zelda Breath of the Wild' , precio: 200},
    { id: 5, nombre: 'Skin Valorant' , precio: 120},
    { id: 6, nombre: 'Taza de Star Wars' , precio: 220}
  ]
}
```
 - Crear endpoint para poder crear un producto nuevo
 - Crear endpoint para poder actualizar un producto
 - Crear endpoint para poder eliminar un producto
 - Crear filtro por precio de producto
 - Crear filtro que muestre los productos con un precio entre 50 y 250.
 - Crear un filtro que cuando busque en postman por parámetro el id de un producto me devuelva ese producto
 - Crear un filtro que cuando busque en postman por parámetro el nombre de un producto me devuelva ese producto

1.2. Extra

- Crea una interfaz con js para el ejercicio anterior utilizando Axios con los siguientes requisitos:
 - Pantalla Productos que muestre los productos
 - Pantalla para añadir productos
 - En la pantalla de productos que cada producto tenga la opción de ser eliminado

****Investigar error CORS api**

1.3. Github

- Subir el proyecto a un repositorio en github
- Hacer commits con frecuencia
- Hacer un pequeño readme