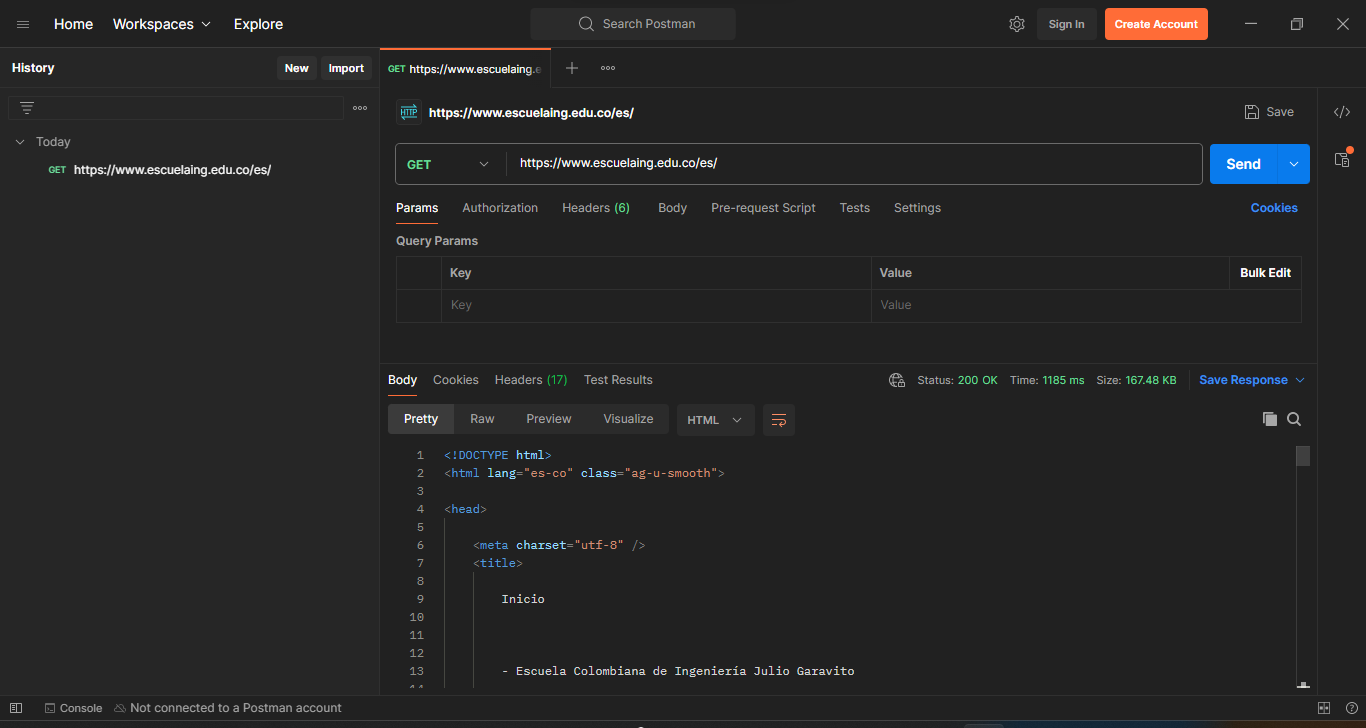
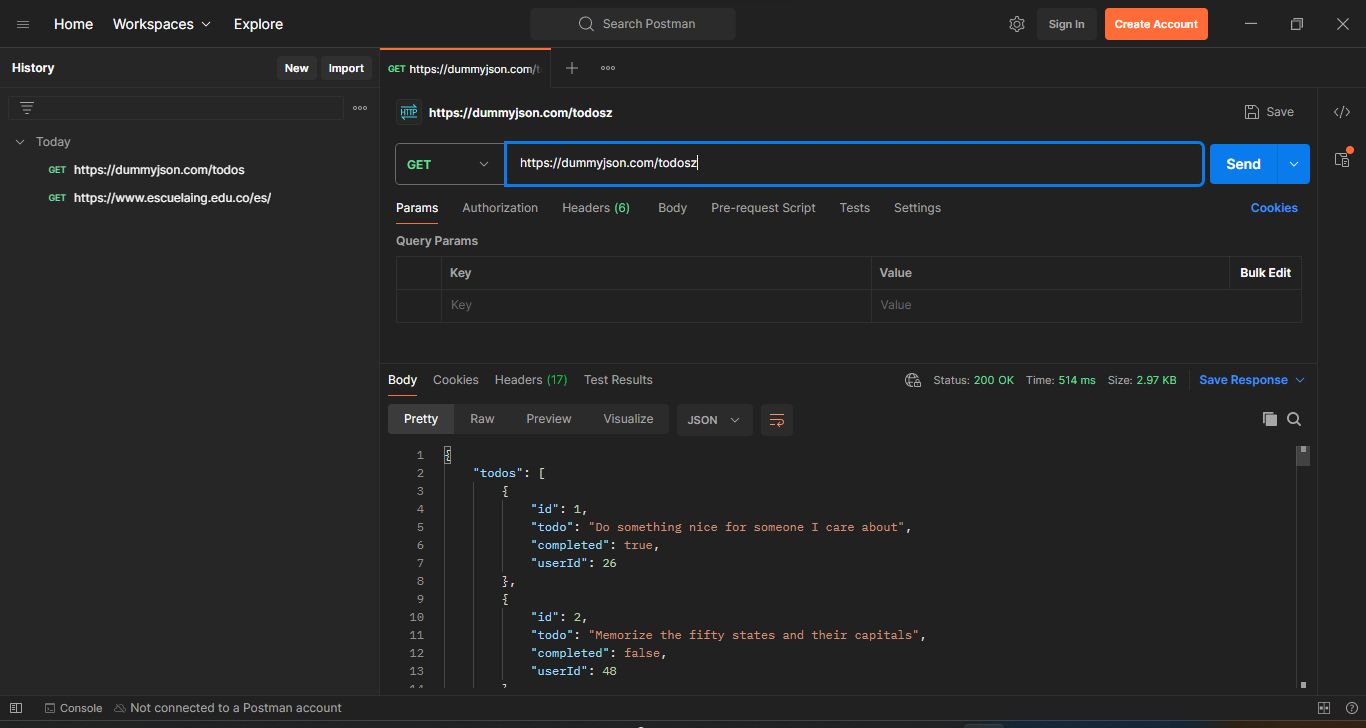
Al hacer el get de escuelaing nos aparece lo siguiente:



Donde podemos ver variada información del sitio web, como su body, sus headers, etc.

Se puede ver de modo Pretty, Raw, Preview; donde cada uno nos da una información variada de la página.

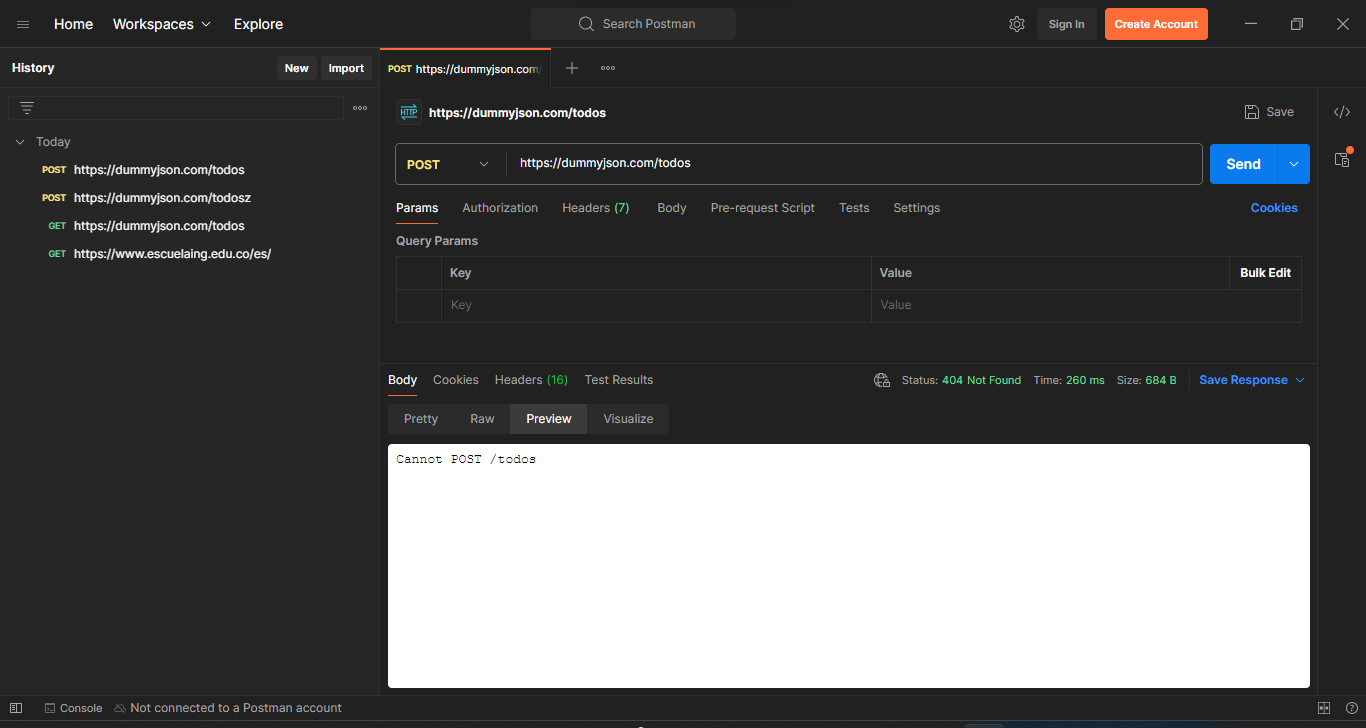
Al hacer el get de DummyJson nos aparece lo siguiente:



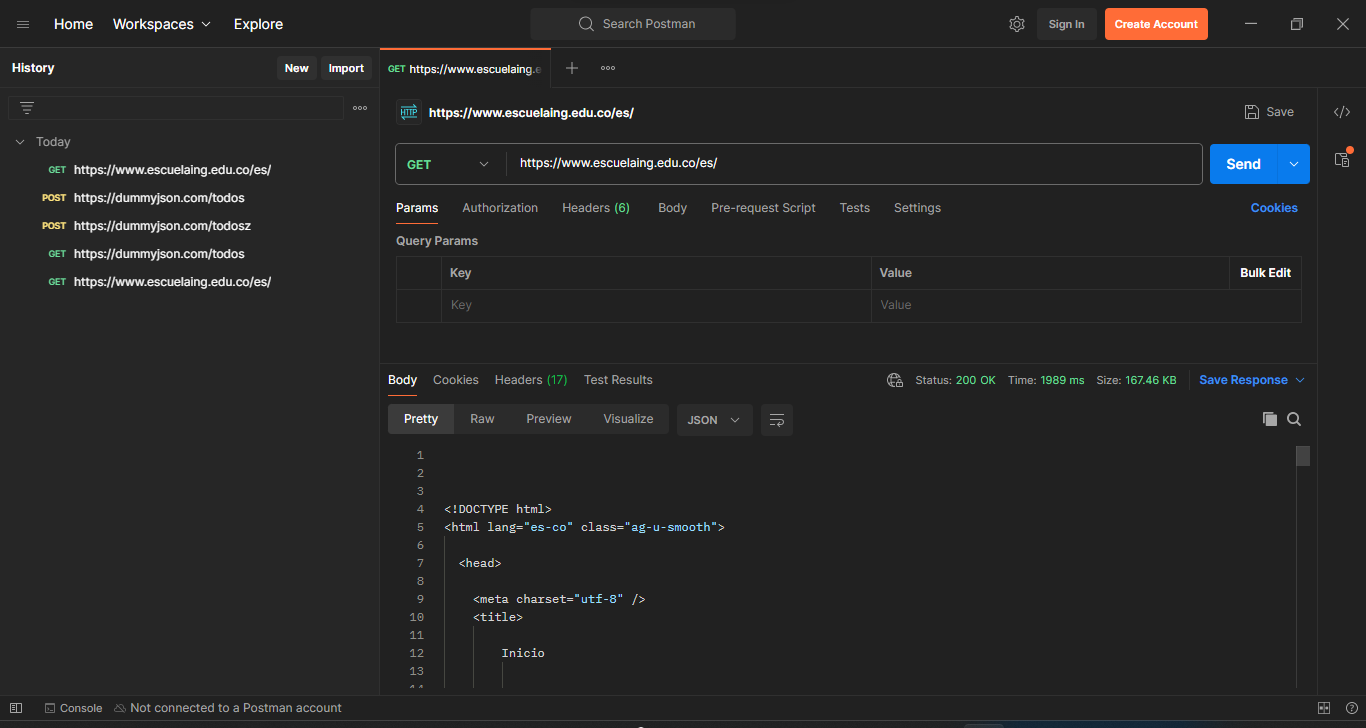
Algo muy similar al de la Escuela, pero vemos que cambia a Json, además aca el raw se ve muy apeñuscado y además el preview es casi igual al raw.

¿Qué pasa si no envío el método correcto?

Manda un Not found y un rechazo



¿Qué pasa si al body response HTML lo fuerzo a leerse como JSON?

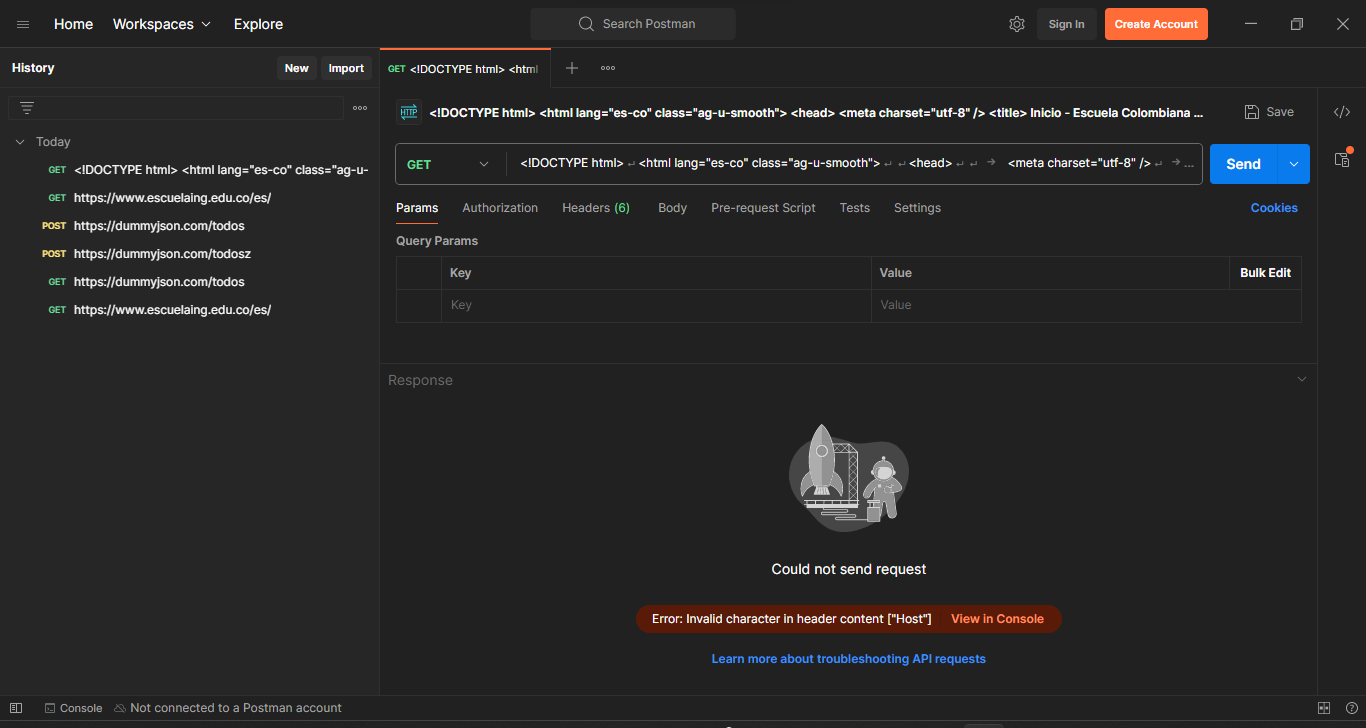


Sigue mostrándose como un HTML

¿Por qué el preview de HTML no se ve igual a cuando accedo a la URL en un navegador web?

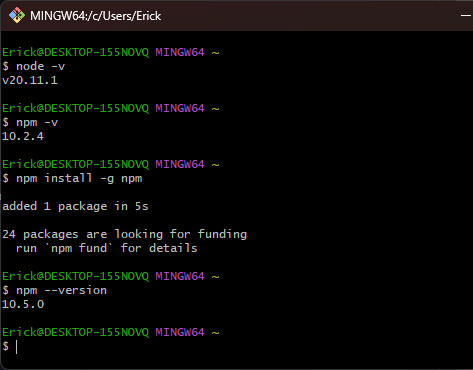
Una posible razón por la que el preview de HTML de postman no se ve igual a cuando accedes a la URL en un navegador web es que postman no carga los recursos externos, como los archivos CSS, que pueden afectar al estilo y la apariencia de la página

¿Qué pasa si le envías un body a una solicitud GET?

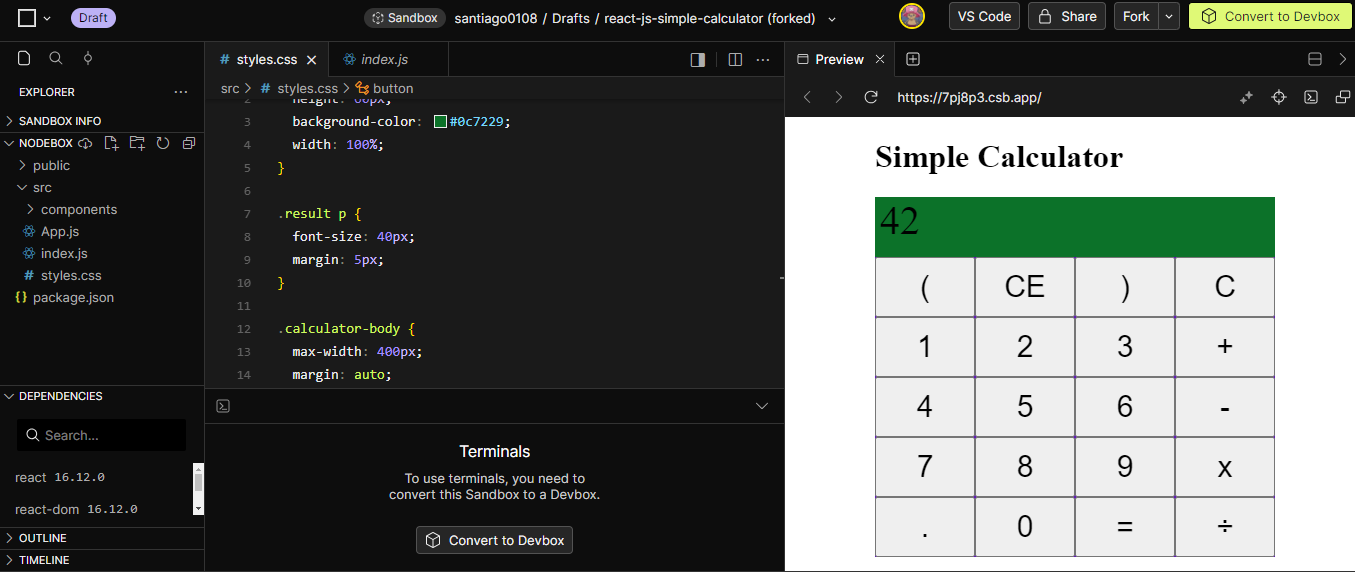


PARTE II. - CLIENT SIDE RENDERING CON REACT

Vemos que si temenos npm



Hago las pruebas de la app de react



¿Qué hace cada uno de los archivos en la aplicación?

package.json: Este archivo contiene la configuración del proyecto, incluyendo las dependencias necesarias para ejecutar la aplicación,

public/index.html: Es el archivo HTML principal que se sirve cuando se accede a tu aplicación web. Actúa como la página de entrada para la aplicación y es donde se vinculan otros recursos como hojas de estilo y scripts de JavaScript

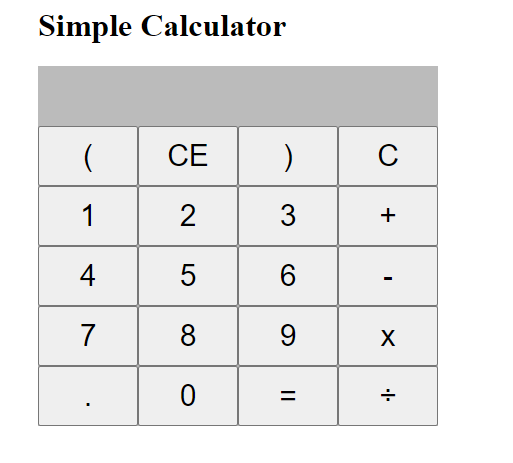
src/App.js: Es el componente principal de React que actúa como la raíz de tu aplicación.

src/components/KeyPadComponent.js y src/components/ResultComponent.js: Estos son componentes de React que tienen los componentes de la app

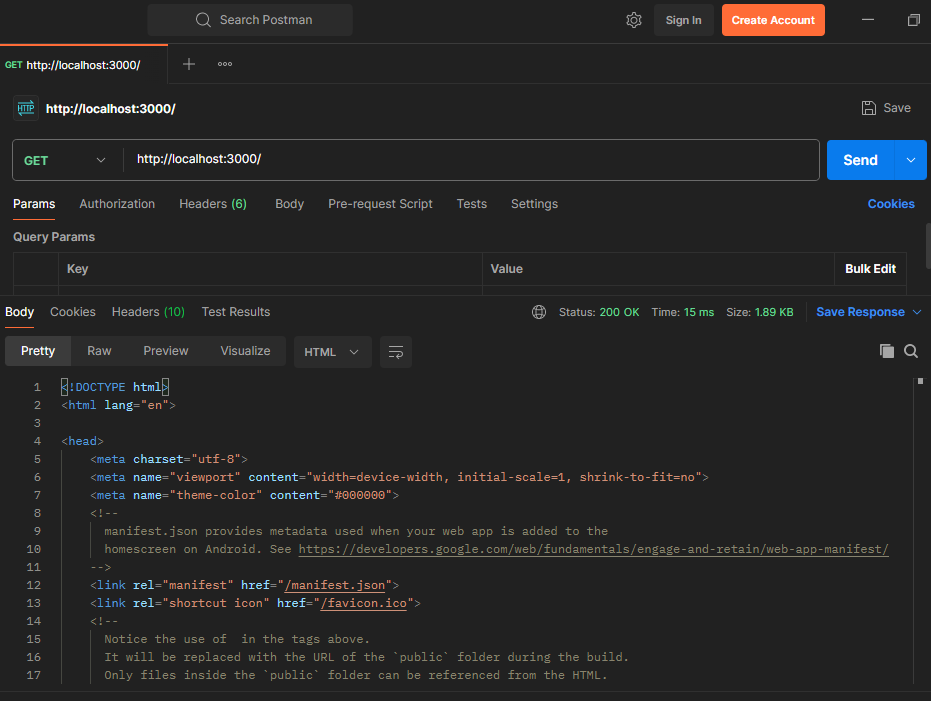
src/index.js: Este es el punto de entrada de JavaScript para la aplicación React. Aquí es donde se renderiza el componente App.js

src/styles.css: Esta hoja de estilo contiene los estilos CSS para personalizar la apariencia de tu aplicación.

Hicimos la prueba y funciono

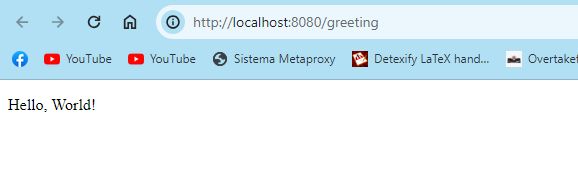


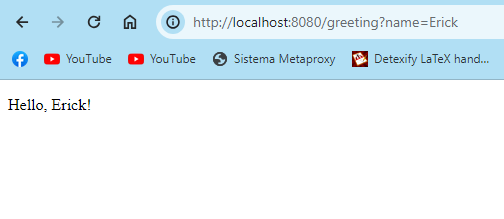
En postman aparece lo siguiente

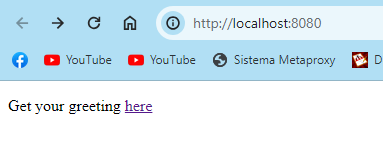


Lo cual es el mismo index.html, lo cual nos dice que es un recurso estático.

PARTE III. - HACIENDO UNA APLICACIÓN WEB DINÁMICA USANDO EL PATRÓN MVC







¿Por qué MVC obtiene ese nombre?

MVC obtiene su nombre del acrónimo en inglés para Model-View-Controller (Modelo-Vista-Controlador), que son los tres componentes principales que conforman este patrón de diseño de software.

¿Cuáles son las ventajas de usar MVC?

Separación de responsabilidades: MVC divide claramente la lógica de negocio (Modelo), la presentación de la interfaz de usuario (Vista) y la gestión de la interacción del usuario (Controlador).

Reutilización de código: Al separar la lógica de negocio del resto del código, se facilita la reutilización de componentes en diferentes partes de la aplicación

Mayor escalabilidad: El patrón MVC hace que sea más fácil agregar nuevas características y funcionalidades sin tener que reescribir grandes secciones de código existentes.}

¿Qué diferencia tiene la estructura de directorios de este proyecto comparado con las de proyectos pasados (con solo maven y java EE)?

La estructura típica de Maven incluye:

src/main/java: Contiene el código fuente Java del proyecto.

src/main/resources: Alberga recursos que no son código, como propiedades o archivos de configuración.

src/test/java: Para el código fuente de las pruebas unitarias.

src/test/resources: Recursos utilizados durante las pruebas.

target/: Directorio donde Maven almacena los outputs de la compilación

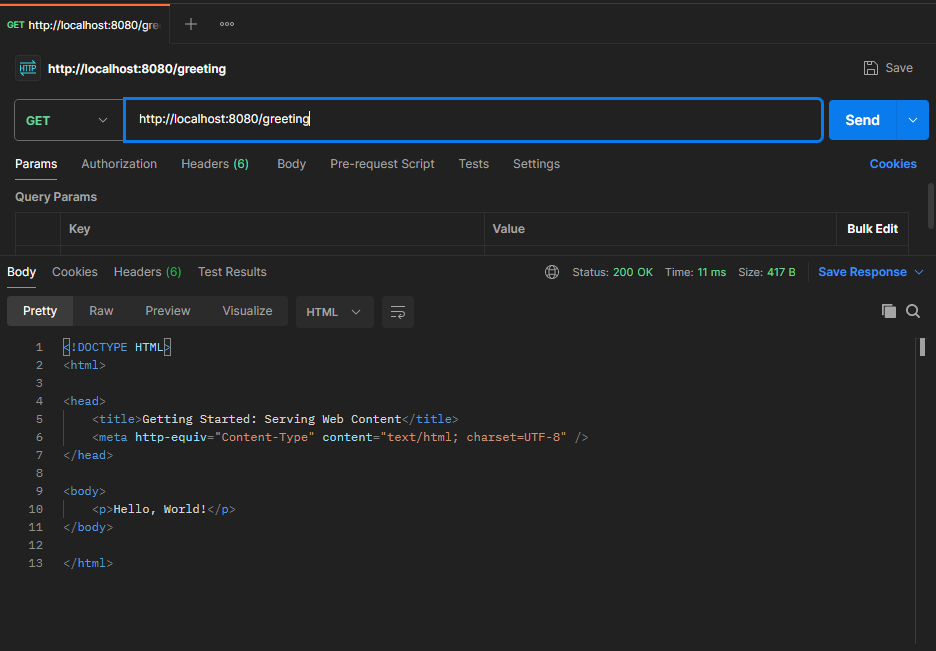
en java EE

src/: Contiene el código fuente Java.

WEB-INF/: Incluye archivos de configuración web, como web.xml, y el directorio classes para archivos compilados.

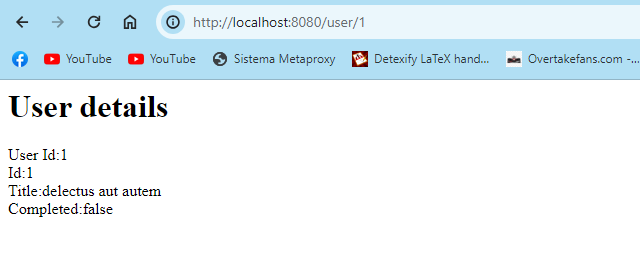
lib/: Almacena las bibliotecas y dependencias del proyecto.

Ahora, haz el request GET http://localhost:8080/greeting usando Postman, y revisa si el body de la respuesta es igual a alguno de los archivos del proyecto. Significa eso que es un recurso web dinámico o estático?



No hay ninguno similar y es dinámico

PARTE IV. - APLICACIÓN MVC PARA CONSUMO DE SERVICIO RESTful



¿Qué es RESTful?

RESTful se refiere a un estilo de arquitectura de software que se utiliza para diseñar servicios web. Se basa en REST, que son las siglas en inglés de “Representational State Transfer” (Transferencia de Estado Representacional). En una API RESTful, se utilizan métodos HTTP estándar como GET, POST, PUT y DELETE para realizar operaciones sobre los recursos representados por URLs. Las respuestas suelen estar en formatos como JSON o XML.

Si utilizo un framework como Boostrap CSS para qué el apartado gráfico se vea más profesional, ¿en qué capa se haría su uso?

Vista o presentación.

PARTE V. - APLICACIÓN MVC JUEGO

¿Qué pasa si abro el sitio de juegos en dos navegadores difententes?

Cada uno guardara una memoria diferente pero la vista y lo relacionado será igual en cada uno

Si quisiera qué a cada jugador le aparecieran independientemente sus respectivos saldos. ¿Qué habría que hacer?