



Instituto Tecnológico de Costa Rica

Diseño de Software

Tarea Individual #3

Estudiante:

Joshua Esteban Sancho Burgos

Grupo 50

Campus San Carlos

12/11/2023

## Contenido

Elección de Tecnologías:.....	2
Frontend.....	2
Backend.....	3
Patrones de Diseño.....	4
Pruebas.....	7
Backend.....	7
Frontend:.....	7
Documentación .....	8
Backend.....	8
Frontend.....	8

## Elección de Tecnologías

### Frontend

Tecnología	Resumen	Justificación
React JS	React es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario (UI) interactivas y reactivas.	React utiliza un paradigma de programación declarativa para construir componentes reutilizables que gestionan su propio estado. Utiliza el concepto de Virtual DOM para mejorar el rendimiento al actualizar solo las partes necesarias de la interfaz de usuario en lugar de recargar toda la página. React se utiliza comúnmente en el desarrollo de aplicaciones de una sola página (SPA) y se integra fácilmente con otras bibliotecas o frameworks.
Vite	Vite es un entorno de desarrollo rápido para aplicaciones web basadas en JavaScript y TypeScript. Es especialmente conocido por su velocidad y eficiencia durante	Vite se centra en la velocidad de desarrollo al utilizar importación de módulos nativa (ESM) y proporcionar un servidor de desarrollo con recarga en caliente (hot module replacement). Permite

	el desarrollo.	a los desarrolladores construir aplicaciones frontend de manera rápida y eficiente, facilitando la configuración cero y admitiendo tecnologías como React, Vue y Svelte.
Bootstrap	Bootstrap es un framework de diseño frontend que proporciona componentes y estilos predefinidos para facilitar el desarrollo web.	Bootstrap está construido sobre HTML, CSS y JavaScript, y ofrece una colección de componentes como botones, formularios, barras de navegación y más. Su enfoque es el diseño responsivo, lo que significa que las aplicaciones creadas con Bootstrap son visualmente atractivas y se adaptan a diferentes tamaños de pantalla. Bootstrap simplifica la creación de interfaces consistentes y atractivas.
React-Bootstrap	React-Bootstrap es una biblioteca que integra Bootstrap con React, permitiendo el uso de los componentes de Bootstrap en aplicaciones React.	React-Bootstrap traduce los componentes de Bootstrap a componentes de React, lo que facilita su integración en proyectos basados en React. Esto permite aprovechar la estética y funcionalidad de Bootstrap mientras se trabaja en un entorno de desarrollo basado en React. Cada componente de React-Bootstrap es un componente de React nativo y puede ser utilizado como tal.

## Backend

Tecnología	Resumen	Justificación
PostgreSQL	PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto y altamente extensible.	PostgreSQL es conocido por su robustez, extensibilidad y soporte para características avanzadas de bases de datos. Es utilizado comúnmente en entornos de desarrollo web como sistema de gestión de bases de datos (DBMS) debido a su capacidad para manejar grandes cantidades de datos y su compatibilidad con estándares SQL.
Express JS	Express.js, o simplemente	Express proporciona una capa

	Express, es un marco de aplicación web para Node.js que simplifica el desarrollo de aplicaciones web y APIs.	delgada sobre Node.js que facilita la creación de rutas, gestión de middleware y manejo de solicitudes y respuestas HTTP. Es altamente flexible y se utiliza comúnmente para construir el backend de aplicaciones web y APIs RESTful. Express facilita la creación de servidores web y el manejo de rutas, middleware, y otras tareas comunes en el desarrollo web.
Axios	Axios también se mencionó en el contexto del frontend, pero es una biblioteca que se puede utilizar tanto en el frontend como en el backend. En el contexto backend, se utiliza para realizar solicitudes HTTP desde el servidor a otros servicios o APIs externas.	Axios proporciona una interfaz simple basada en promesas para realizar solicitudes HTTP en el servidor. Puede ser útil cuando se necesitan realizar solicitudes a otros servicios desde el backend, como acceder a una API externa o realizar operaciones HTTP desde el servidor.
Cors	CORS (Cross-Origin Resource Sharing) es un mecanismo que permite o restringe las solicitudes HTTP realizadas desde un dominio distinto al dominio del servidor.	En el contexto de Node.js y Express, el middleware cors se utiliza para habilitar CORS en una aplicación. Permite o restringe el acceso a recursos del servidor desde diferentes orígenes. Esto es crucial cuando se construyen APIs que deben ser consumidas por clientes en dominios diferentes al del servidor.
Helmet	Helmet es un conjunto de middleware para Express que ayuda a mejorar la seguridad de una aplicación web mediante la configuración de diversos encabezados HTTP.	Helmet ayuda a proteger la aplicación contra diversas vulnerabilidades al establecer encabezados HTTP como Content Security Policy (CSP), Strict-Transport-Security (HSTS), y otros. Mejora la seguridad predeterminada de una aplicación Express al proporcionar medidas de seguridad adicionales para proteger contra ataques comunes en la web.

### Patrones de Diseño

Nombre	Explicación
--------	-------------

Patrón HOC	<p>El patrón de diseño HOC es comúnmente utilizado en React. Consiste en una función que toma un componente y devuelve un nuevo componente con funcionalidades adicionales. Estas funcionalidades pueden incluir lógica de estado compartida, manipulación de props, o cualquier otra mejora de comportamiento. Los HOC permiten reutilizar la lógica entre componentes de manera eficiente, ya que encapsulan la funcionalidad común en una función independiente.</p>
Compound	<p>El patrón Compound es un patrón de diseño estructural que se centra en la composición de objetos para crear estructuras más complejas.</p> <p>En el contexto de componentes de interfaz de usuario, como los mencionados (select, dropdown, menu items), el patrón Compound permite organizar estos componentes en una estructura jerárquica.</p> <p>Consiste en un componente principal, llamado compuesto, que contiene subcomponentes individuales (simples o compuestos) como partes de sí mismo. Cada componente, ya sea simple o compuesto, se trata de manera uniforme, lo que facilita la</p>

	creación de interfaces de usuario complejas y estructuras de objetos organizadas.
Render Props	El patrón Render Props implica pasar una función (el "render prop") como prop a un componente, permitiendo que el componente hijo invoque esa función para renderizar parte de su salida. Esto proporciona una forma flexible de compartir lógica entre componentes y permite una composición más dinámica. Es una alternativa a los HOC y se utiliza para lograr la reutilización de comportamiento.
API REST	Aunque no es un patrón de diseño en el sentido clásico, es importante mencionar API REST ya que es un enfoque arquitectónico comúnmente utilizado para diseñar servicios web. REST se basa en la representación de recursos y las operaciones CRUD (Crear, Leer, Actualizar, Borrar) realizadas en estos recursos utilizando los métodos HTTP estándar (GET, POST, PUT, DELETE). Los servicios RESTful suelen ser simples, escalables y ampliamente utilizados para construir APIs en la web.

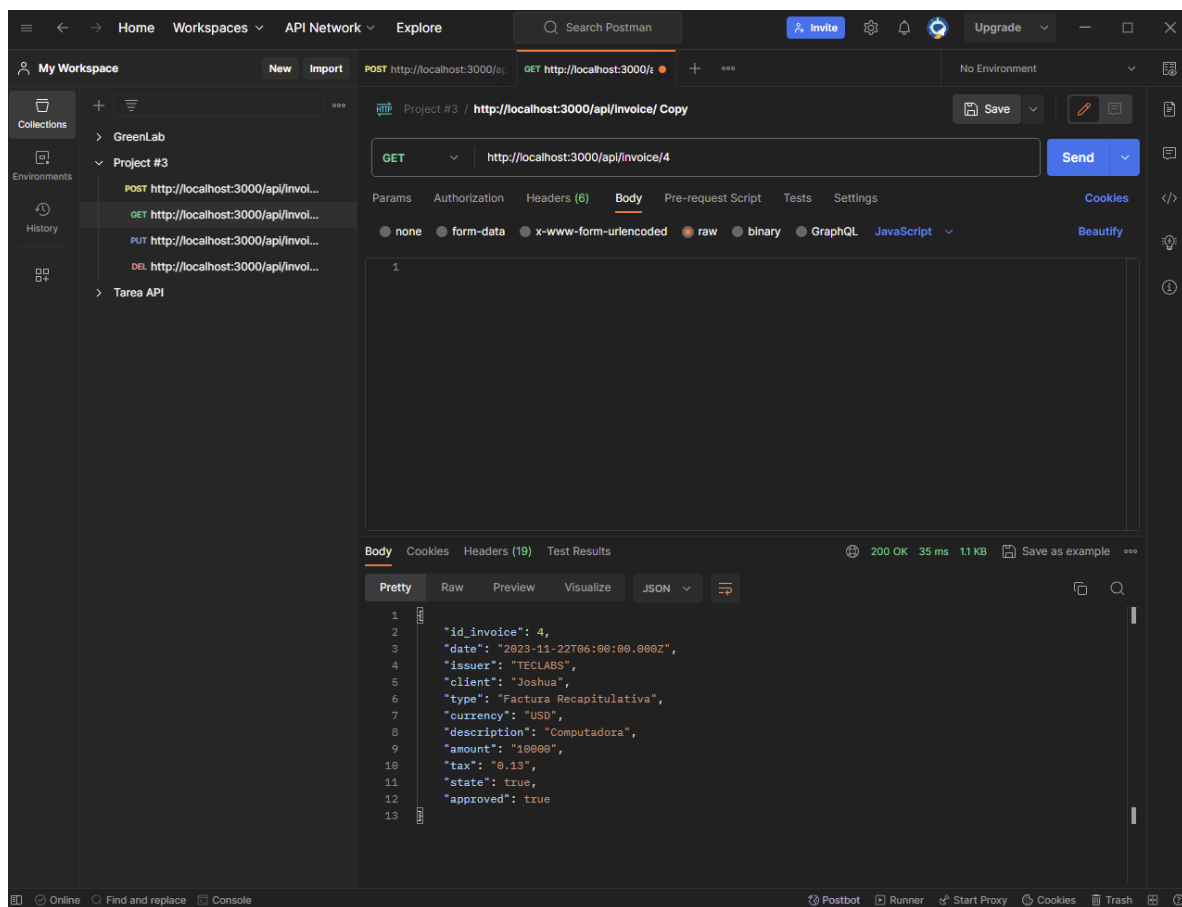
## Pruebas

### Backend

Se realizaron las siguientes solicitudes http con Postman para probar la API REST:

- POST: <http://localhost:3000/api/invoice/>
- GET: <http://localhost:3000/api/invoice/>
- PUT: <http://localhost:3000/api/invoice/1>
- DELETE: <http://localhost:3000/api/invoice/3>

Todas las pruebas dieron respuesta y funcionaron según los parámetros establecidos.



### Frontend

Se realizaron múltiples pruebas que aseguraron el funcionamiento correcto de todos los componentes.

ID	Fecha	Issuer	Cliente	Tipo	Moneda	Descripción	Monto	Impuesto	Estado	Aprobada	Opciones
4	22/11/2023	TECLABS	Joshua	Factura Recapitulativa	CRC	Computadora	10000	13%	SI	No	<a href="#">Editar</a> <a href="#">Eliminar</a>

## Documentación

### Backend

- server.js: Este archivo contiene el servidor Node.js con Express que actúa como la API para la aplicación de facturas. Utiliza PostgreSQL como base de datos para almacenar y recuperar datos de facturas. Proporciona rutas para obtener todas las facturas, obtener una factura por ID, agregar una **nueva factura, actualizar una factura existente y eliminar una factura.**

### Frontend

- InvoiceForm.jsx: Este archivo contiene un componente de formulario en React (InvoiceForm). Este formulario se utiliza para agregar y editar facturas. Utiliza las bibliotecas react-bootstrap y moment para la interfaz de usuario y el manejo de fechas.
- InvoiceTable.jsx: Aquí se encuentra otro componente de React (InvoiceTable). Este componente muestra una tabla que representa las facturas existentes. Además, proporciona botones para editar y eliminar facturas. Al igual que InvoiceForm, utiliza react-bootstrap para la interfaz de usuario.
- NavMenu.jsx: El archivo NavMenu contiene un componente de navegación (NavMenu). Este componente representa la barra de navegación superior de la aplicación. Utiliza react-bootstrap para estructurar la barra de navegación y estilizarla. Muestra el nombre de la aplicación y enlaces a secciones como "Documentos" y "Reportes".



- InvoiceApp.jsx: Este es el componente principal de la aplicación (InvoiceApp). Utiliza los componentes anteriores (InvoiceForm, InvoiceTable, NavMenu) para construir la interfaz de usuario completa. También incluye lógica para manejar el estado de las facturas, la factura actual y si se está editando o no. Además, utiliza axios para realizar solicitudes HTTP a la API del servidor.