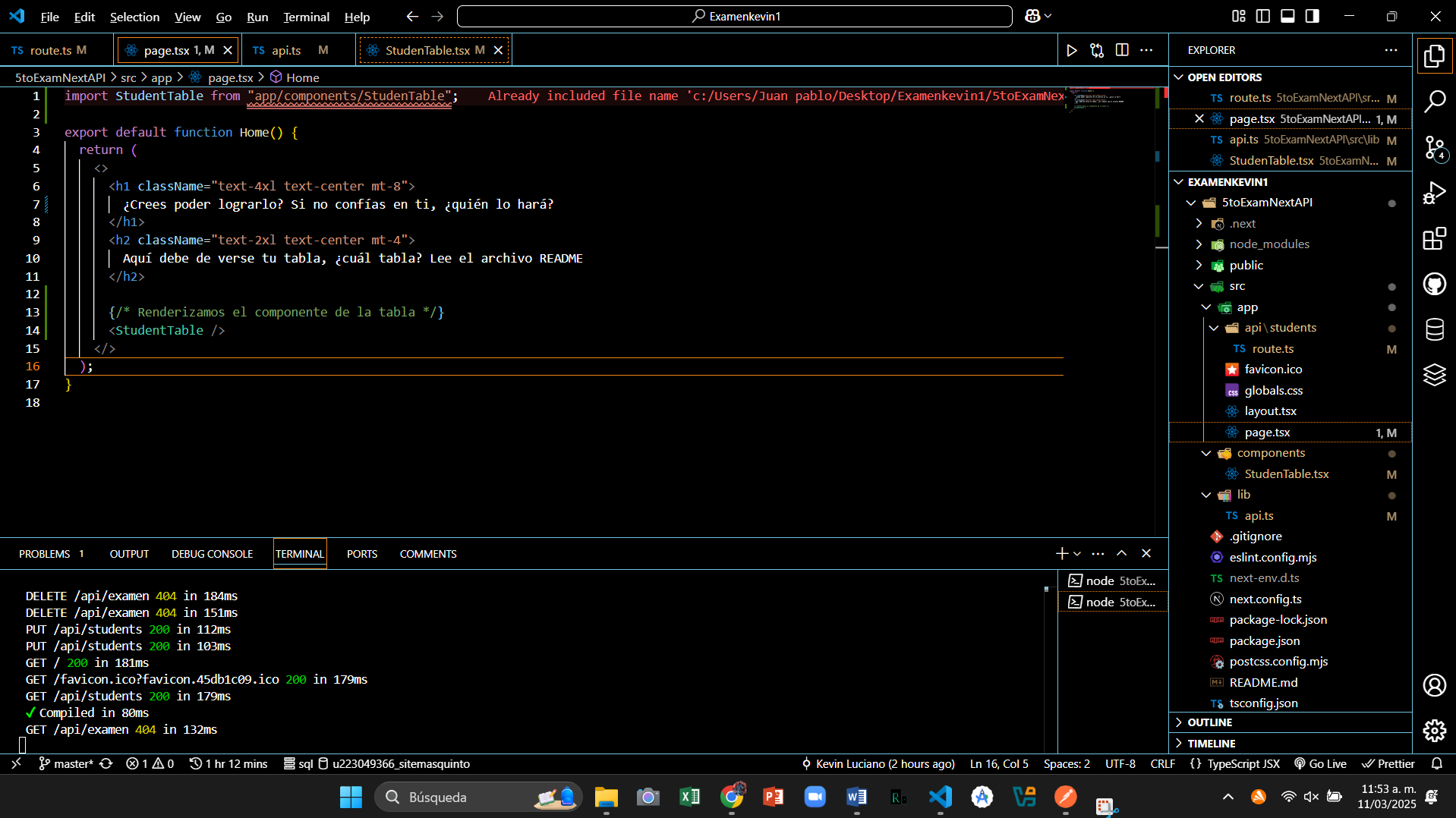
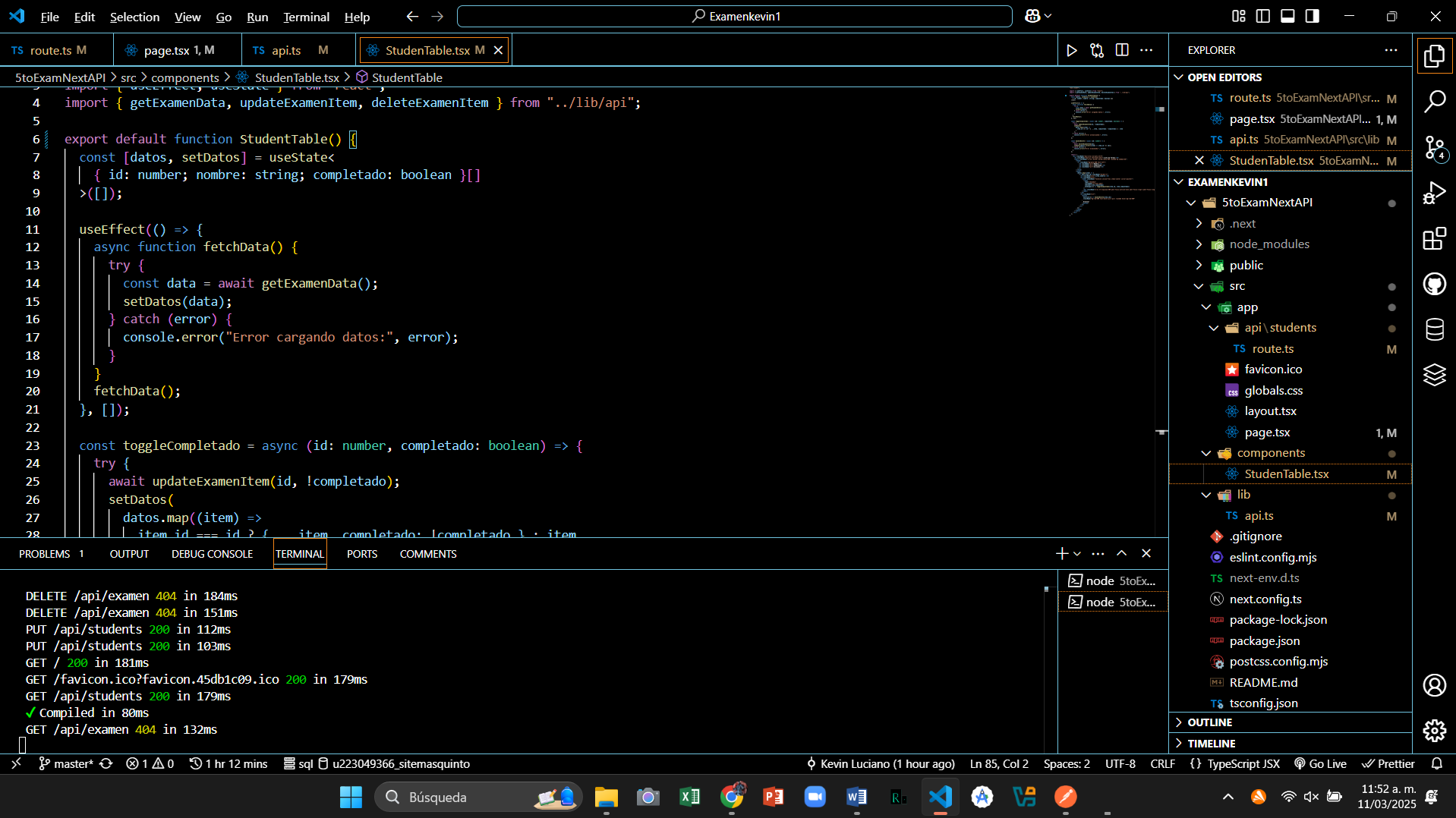
Documentación





El error que encontraste en tu proyecto tiene que ver con la **importación incorrecta del archivo StudentTable.tsx en tu archivo page.tsx**. Aquí te explico qué fue lo que ocurrió y cómo se resolvió:

**El error**

El error que encontraste fue el siguiente:

Module not found: Can't resolve 'app/components/studenTable'

Esto indica que el sistema no podía encontrar el archivo StudentTable.tsx cuando tratabas de importarlo en tu archivo page.tsx.

**Causas del error**

1. **Ruta incorrecta de importación**: El error ocurrió porque la ruta de importación no coincidía con la ubicación real del archivo. Cuando escribiste:

import StudentTable from "app/components/studenTable";

El sistema trató de encontrar el archivo en una ubicación que no existía, ya que tu archivo StudentTable.tsx probablemente no estaba dentro de la carpeta app/components/.

1. **Error tipográfico**: También había un error tipográfico en el nombre del archivo importado. Estabas usando studenTable, pero el nombre correcto es StudentTable. Los sistemas de archivos (especialmente en Linux o macOS) son sensibles a mayúsculas y minúsculas, por lo que studenTable no es lo mismo que StudentTable.

**Solución**

La solución fue **corregir la ruta de importación** y asegurarse de que el nombre del archivo fuera correcto.

1. **Verifica la estructura de carpetas**: Debes asegurarte de que el archivo StudentTable.tsx esté en la carpeta correcta dentro de tu proyecto, y que tu archivo page.tsx pueda acceder a él correctamente.

**Estructura de carpetas esperada**:

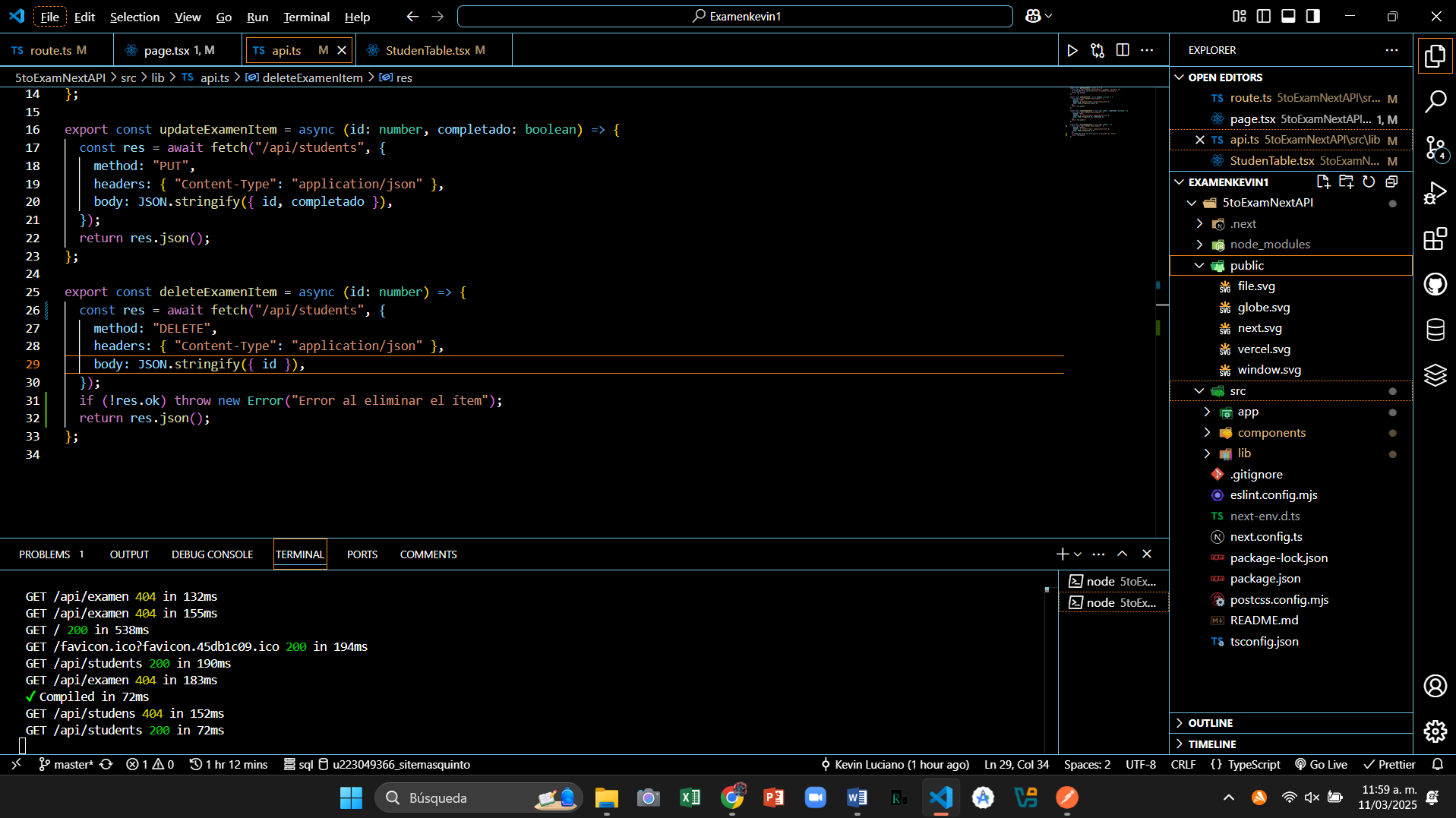
/src

├── /app

│ └── page.tsx

└── /components

└── StudentTable.tsx



**Documentación de la solución**

**Problema original**

El código original tenía un error en la función deleteExamenItem. El endpoint utilizado era /api/examen, pero en las otras funciones (como getExamenData, addExamenItem, y updateExamenItem) se utilizaba /api/students. Esto causaba inconsistencias en las solicitudes al servidor, lo que probablemente generaba errores.

Además, no había un manejo adecuado de errores en caso de que la solicitud DELETE fallara.

**Solución implementada**

Se corrigió el código para asegurar que:

1. **El endpoint sea consistente**: Se cambió el endpoint de /api/examen a /api/students para que coincida con las otras funciones.
2. **Se manejen errores correctamente**: Se agregó una verificación para asegurarse de que la respuesta del servidor (res) sea válida. Si no lo es, se lanza un error.
3. **Se retorne la respuesta del servidor**: La función ahora retorna res.json() para manejar la respuesta del servidor de manera adecuada.

Aquí está el código corregido:

export const deleteExamenItem = async (id: number) => {

// Realiza una solicitud DELETE al endpoint /api/students

const res = await fetch("/api/students", {

method: "DELETE", // Método HTTP DELETE

headers: { "Content-Type": "application/json" }, // Cabeceras para indicar que el cuerpo es JSON

body: JSON.stringify({ id }), // Cuerpo de la solicitud con el ID del ítem a eliminar

});

// Verifica si la respuesta no es exitosa (status code diferente de 200-299)

if (!res.ok) throw new Error("Error al eliminar el ítem");

// Retorna la respuesta del servidor en formato JSON

return res.json();

};

**Explicación del código**

1. **Solicitud DELETE**:
   * Se utiliza fetch para realizar una solicitud HTTP DELETE al endpoint /api/students.
   * El método DELETE se usa para indicar que se desea eliminar un recurso en el servidor.
2. **Cabeceras (headers)**:
   * Se especifica que el cuerpo de la solicitud (body) está en formato JSON mediante la cabecera "Content-Type": "application/json".
3. **Cuerpo de la solicitud (body)**:
   * El cuerpo de la solicitud contiene el id del ítem que se desea eliminar. Este id se convierte a formato JSON usando JSON.stringify.
4. **Manejo de errores**:
   * Si la respuesta del servidor no es exitosa (res.ok es false), se lanza un error con el mensaje "Error al eliminar el ítem".
5. **Retorno de la respuesta**:
   * Si la solicitud es exitosa, se retorna la respuesta del servidor en formato JSON usando res.json().