**¿Que es un gestor dependencias de codigo?**

se encarga de manejar el “ecosistema” de software que tu proyecto requiere para no tener que hacerlo manualmente

**¿Qué es npm?**

es el gestor de paquetes oficial para Node.js, utilizado para instalar, compartir y gestionar herramientas y bibliotecas JavaScript

**¿Para qué se utiliza principalmente npm?**

Para utilizar bibliotecas o copiar codigo de otros desarrolladores

**¿Qué es el versionado semántico?**

es un sistema de numeración para versiones de software que comunica claramente el tipo de cambios realizados en cada actualización

**Como está especificado el Versionado Semántico**

Para hacer adaptaciones tipo hacer una nueva version o corregir codigo o añadir algo a una version

**Qué son las dependencias locales**

Es cuando el proyecto esta dentro de tu consola y no se sube al remoto puedes descargar del remoto y tenerlo en el local modificar pero no se vera reflejado en el remoto

**Qué son las dependencias de desarrollo**

Las dependencias de desarrollo son bibliotecas o paquetes que un proyecto necesita solo durante el proceso de desarrollo, pero no para su funcionamiento final en producción

**Qué son las dependencias globales**

son paquetes o módulos que se instalan en un sistema de forma que están disponibles para cualquier proyecto o aplicación, sin necesidad de instalarlos localmente en cada uno

**¿Que es el archivo package.json y que apartados tiene y cual es su utilidad?**

es la base de cualquier proyecto Node.js o basado en JavaScript moderno y su utilidad es Instalación automática de dependencias, Control de versiones, Automatización de tareas, Compartir proyectos fácilmente, Integración con herramientas como Git, CI/CD, linters, bundlers

**que es la carpeta node\_modules en un proyecto de npm**

es donde NPM guarda todas las dependencias que tu proyecto necesita para funcionar. Se crea automáticamente cuando ejecutas y contiene los paquetes que has especificado en tu archivo

**como inicializar un proyecto de npm.**

Npm init

**Cómo instalar dependencias locales**

Npm install

**Cómo instalar dependencias de desarrollo**

Npm install nombre del paquete –save-dev

**Cómo instalar dependencias globales**

Npm install -g

**Cómo visualizar los dependencias instaladas**

Npm list

**Cómo instalar una versión específica de un paquete**

Npm install nombre del paquete@version

**Cómo crear un comando en tu proyecto scripts en el package.json**

"scripts": {

"start": "node index.js",

"dev": "nodemon index.js"

}

**Cómo actualizar dependencias**

npm update

**Cómo eliminar paquetes**

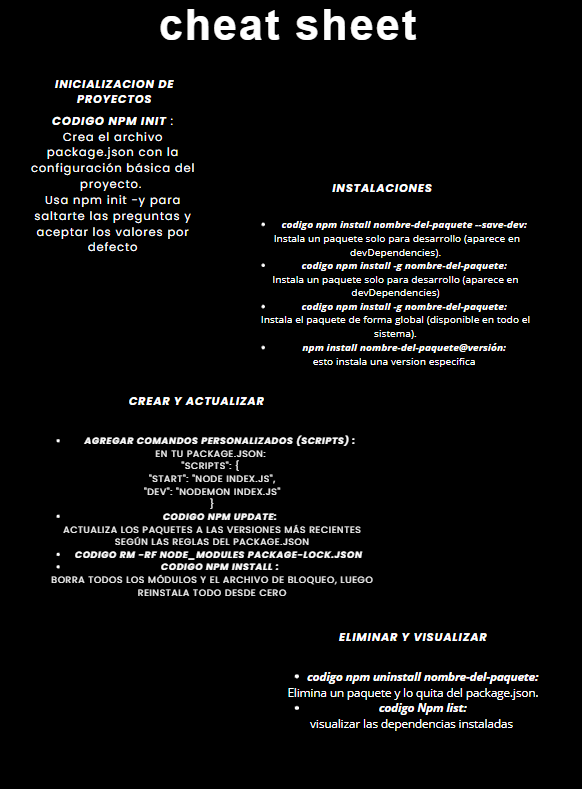
Npm unistall

**Cómo actualizar node\_modules**

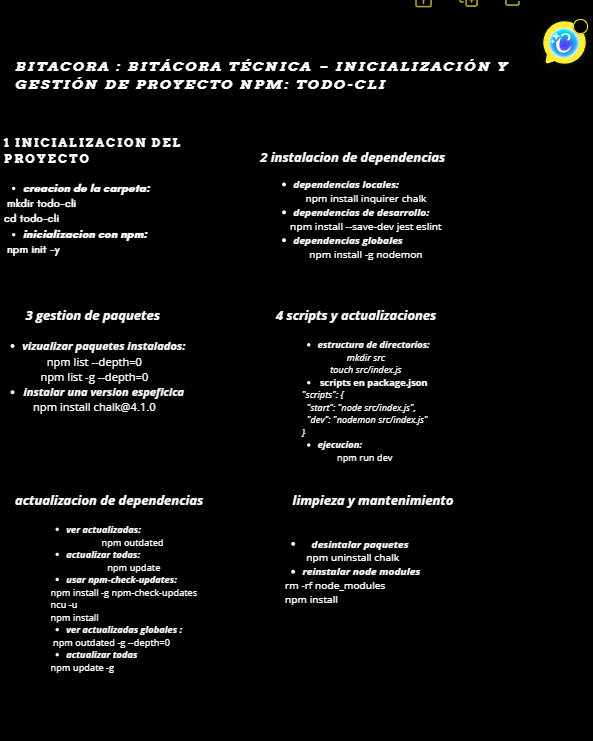
rm -rf node\_modules package-lock.json

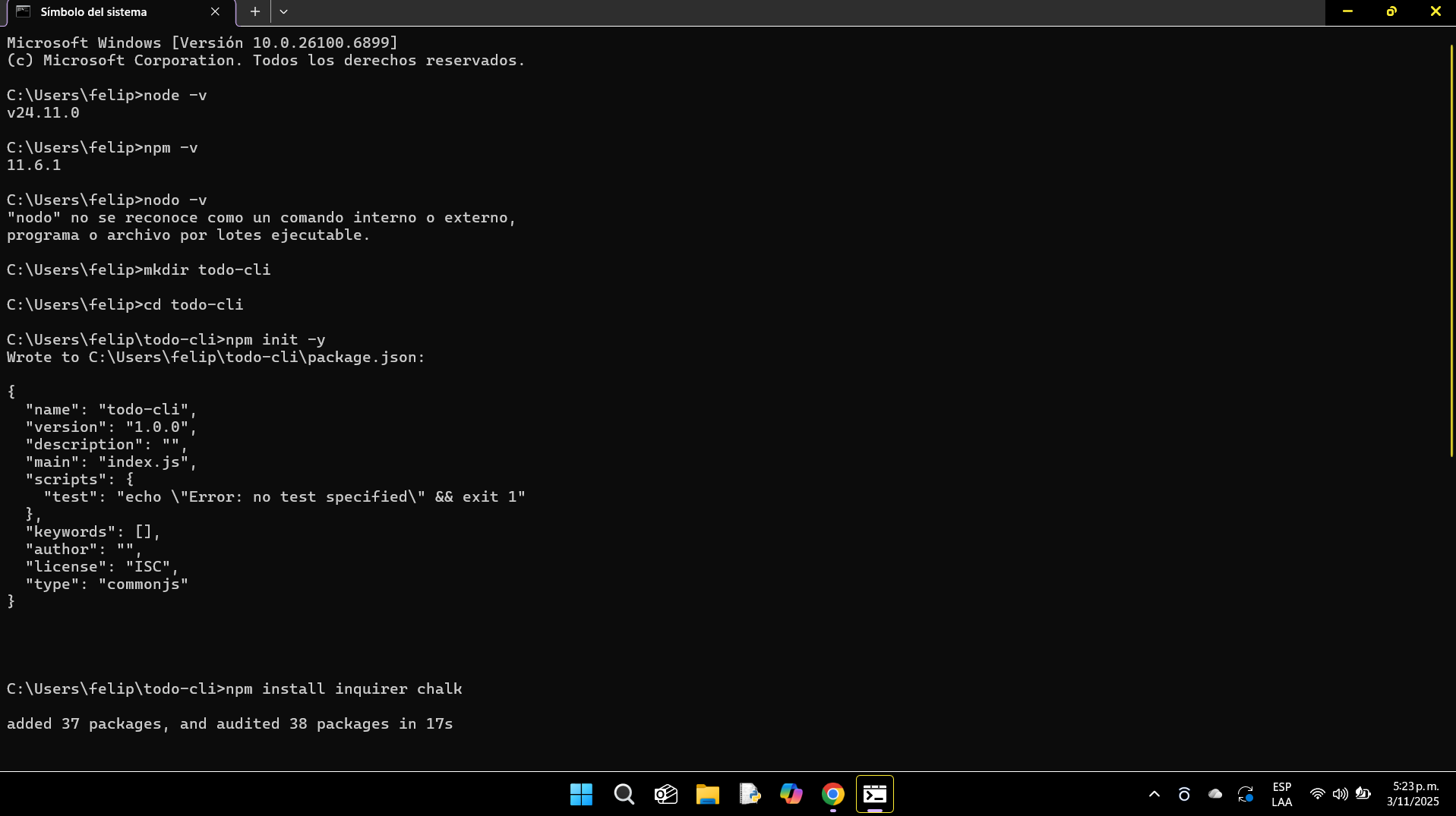
npm install

## **NPM Cheat Sheet – Inicialización y gestión de proyectos**

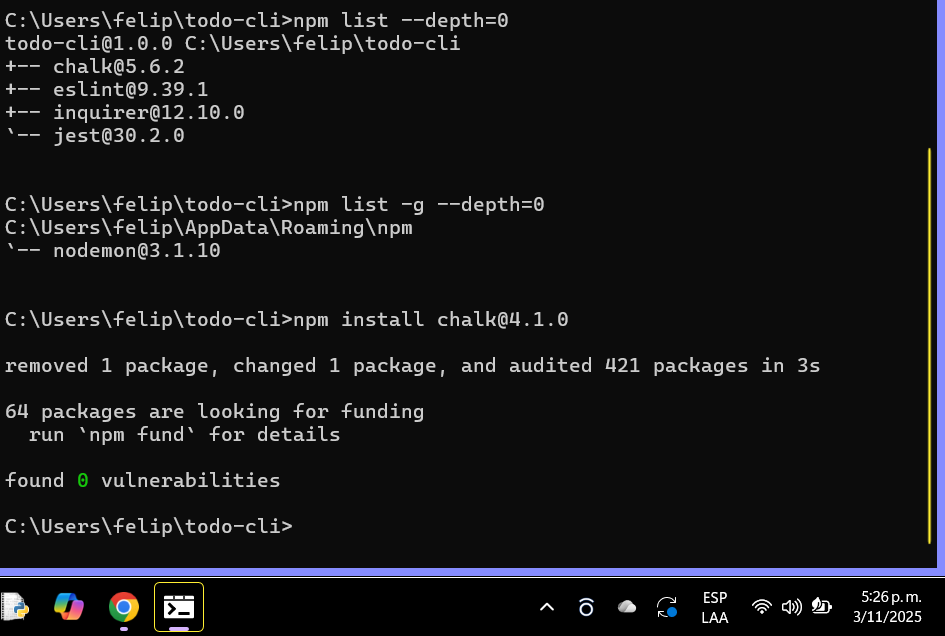


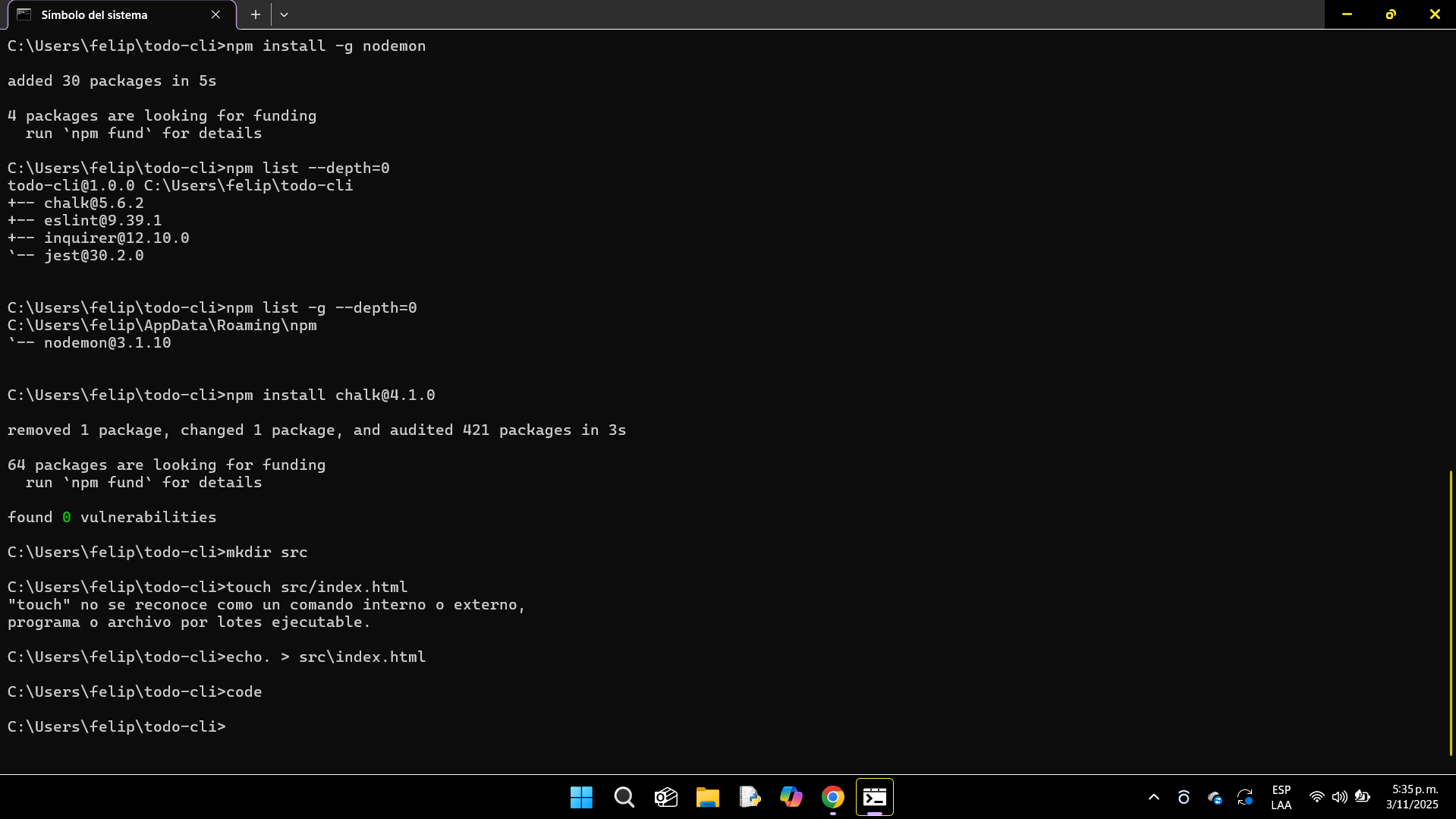
# **Informe tipo bitacora npm**

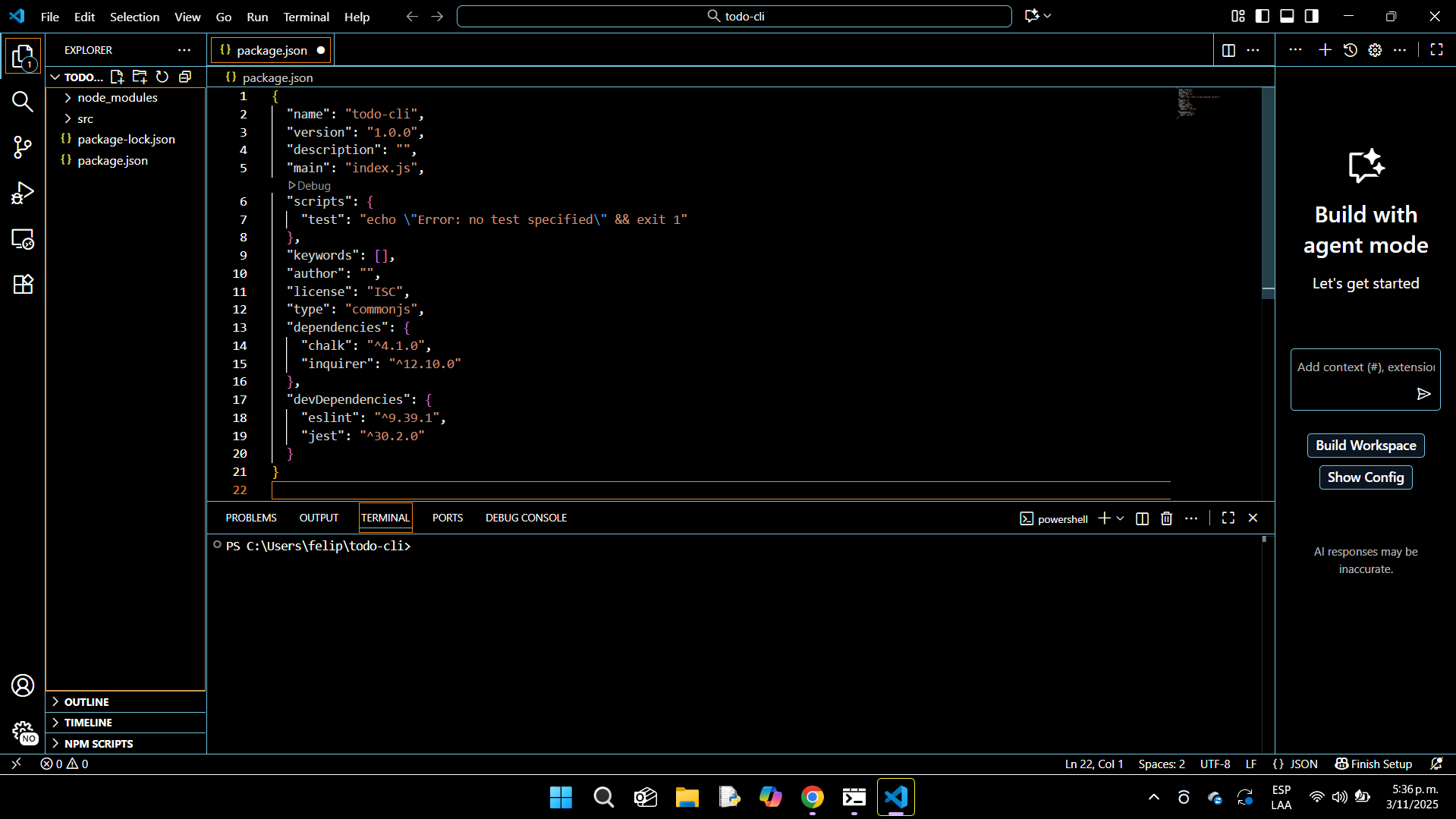


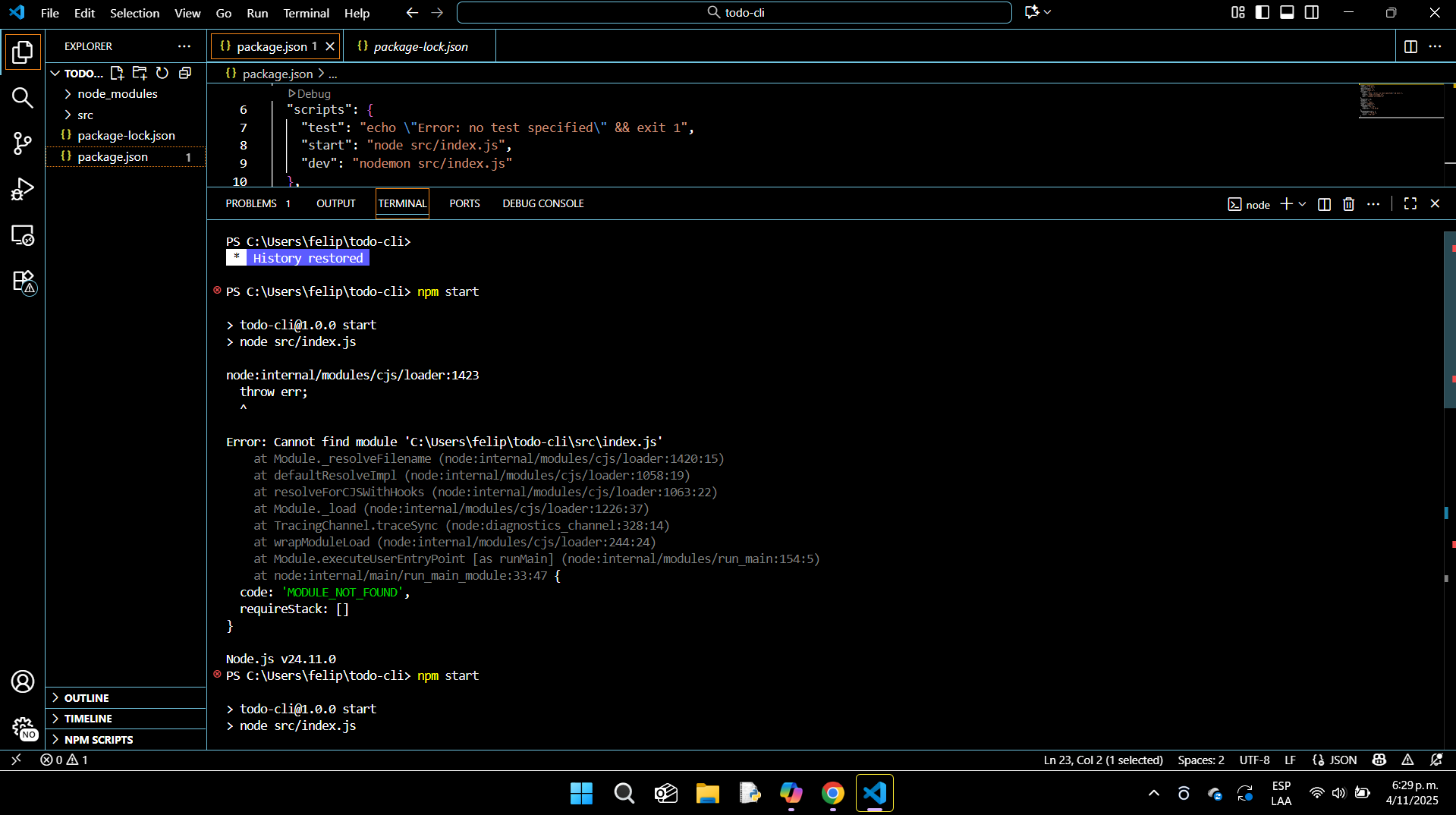


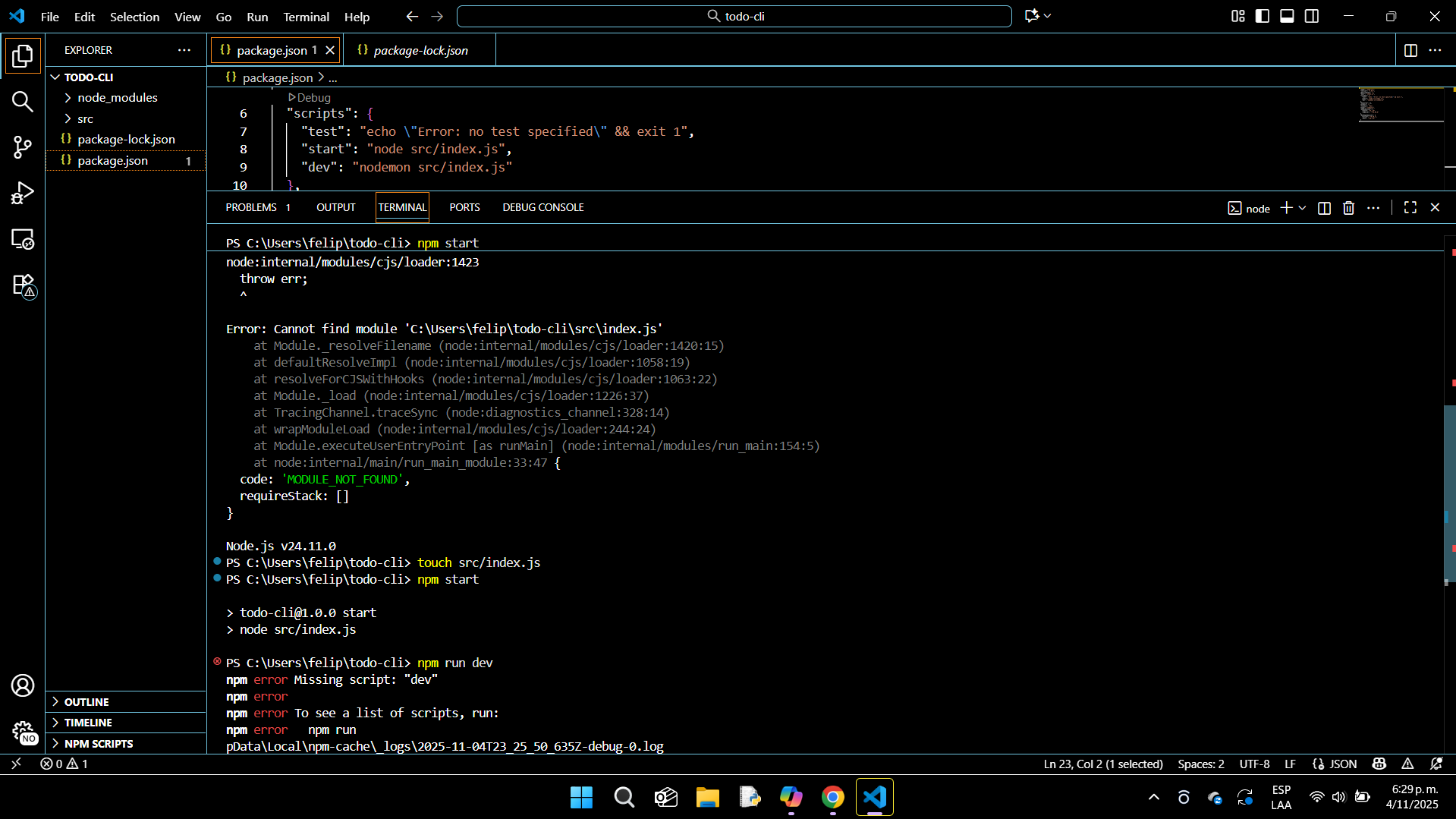






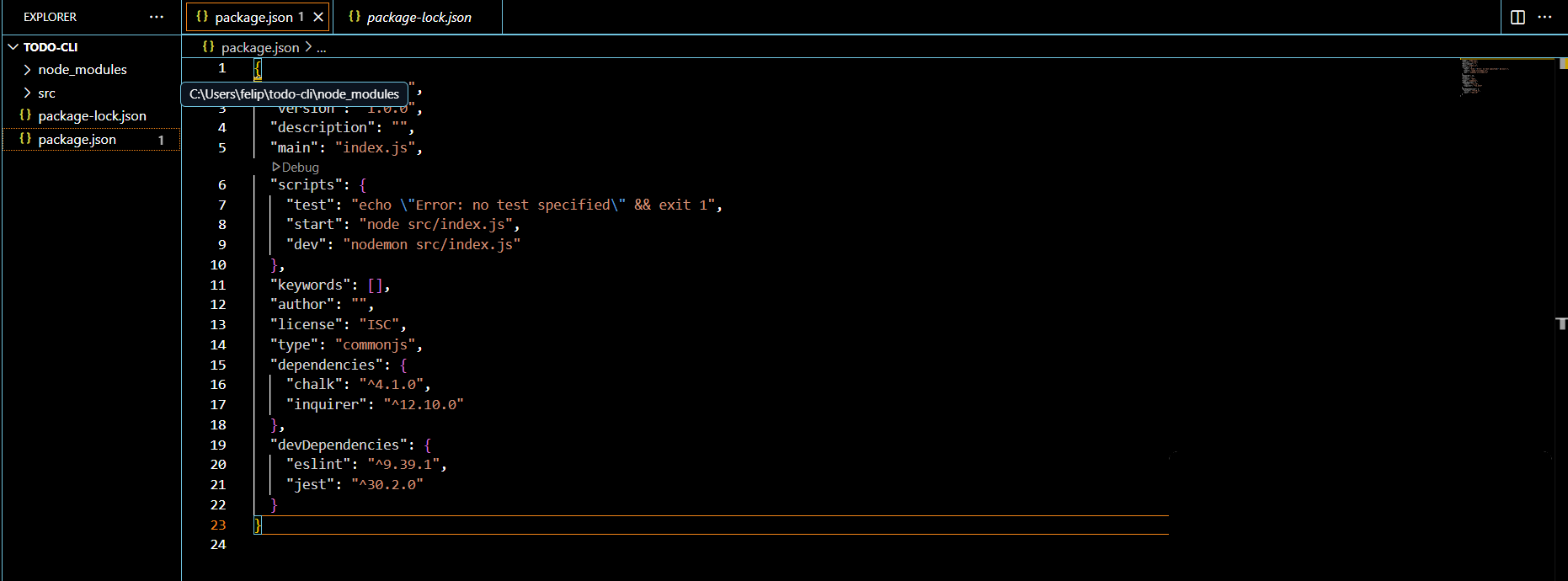


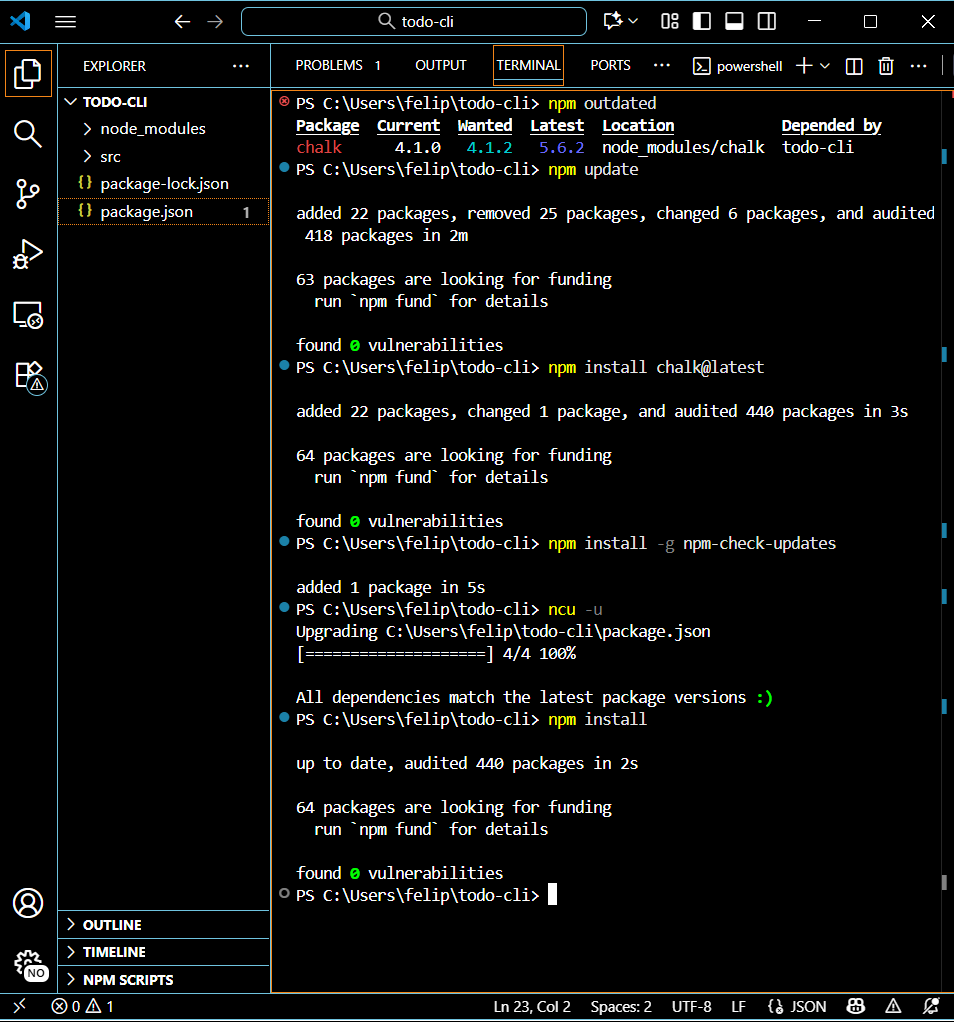


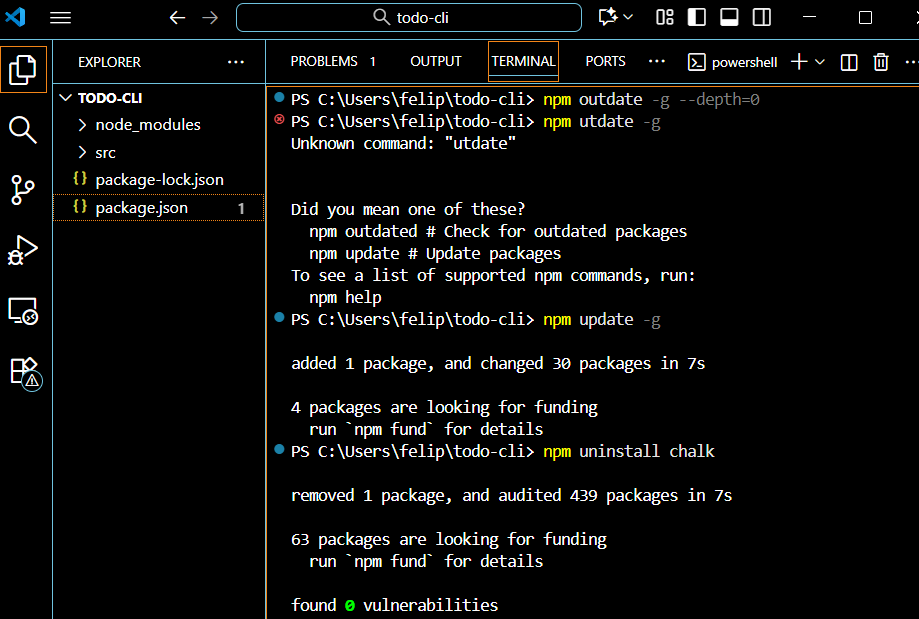


C:\Users\felip\Mi\_primera-\_Api-1\mi\_primera\_api\server.js

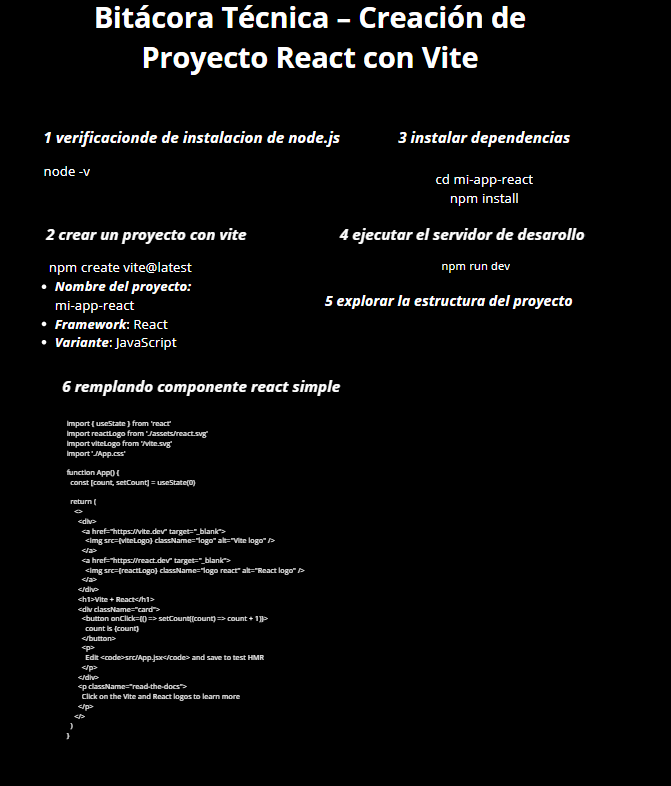




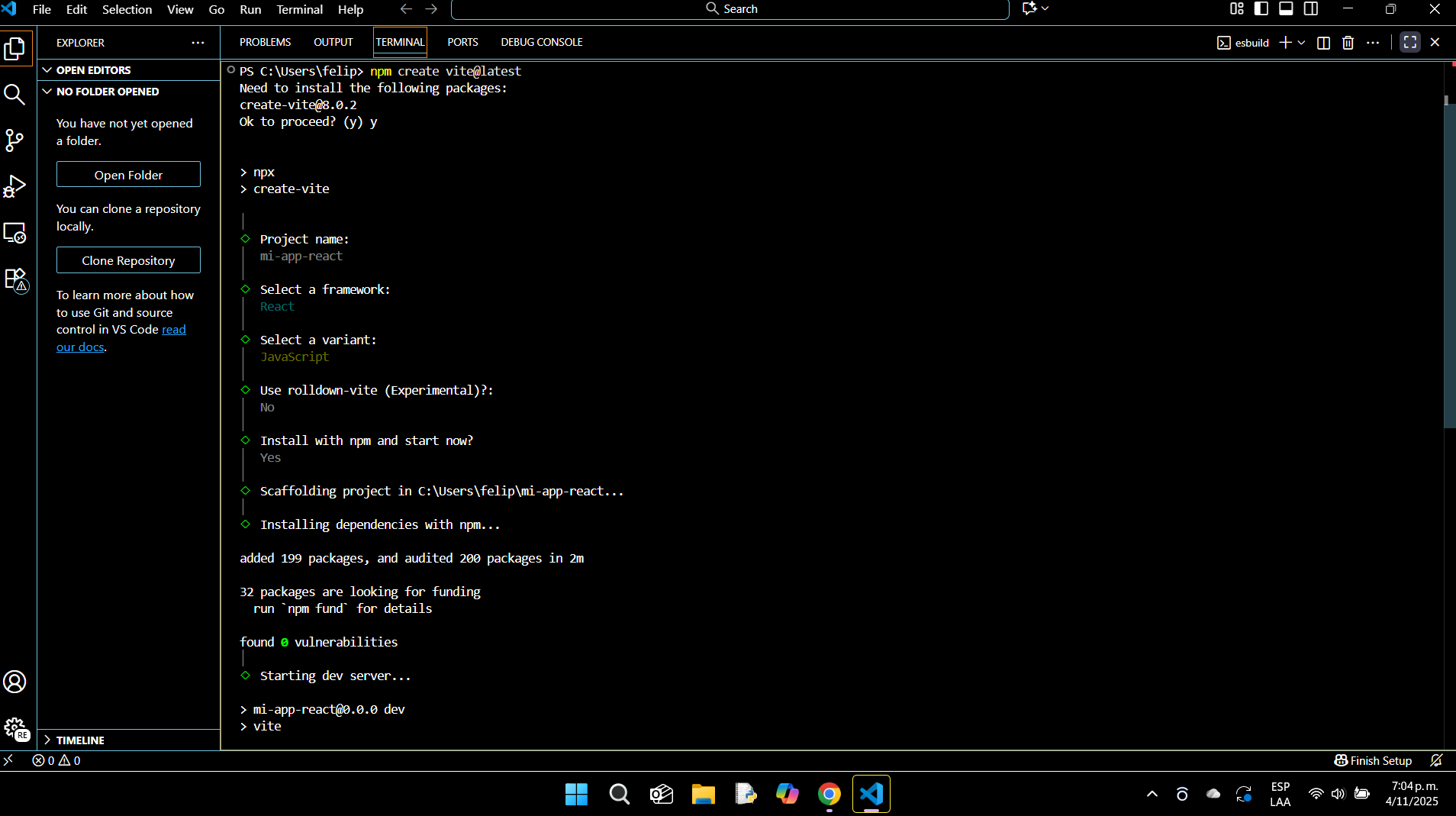


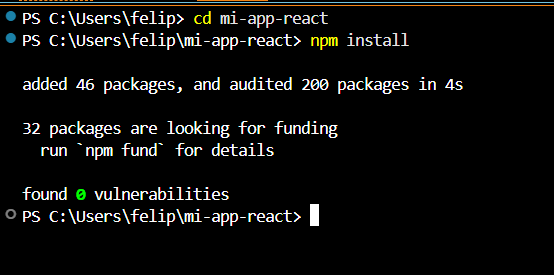


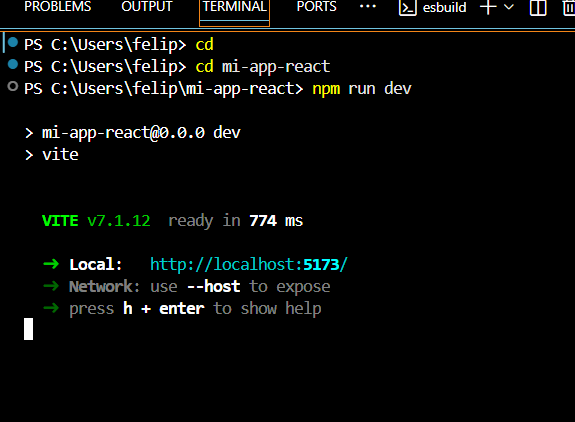
# Laboratorio tipo bitacora



Creando proyecto con vite

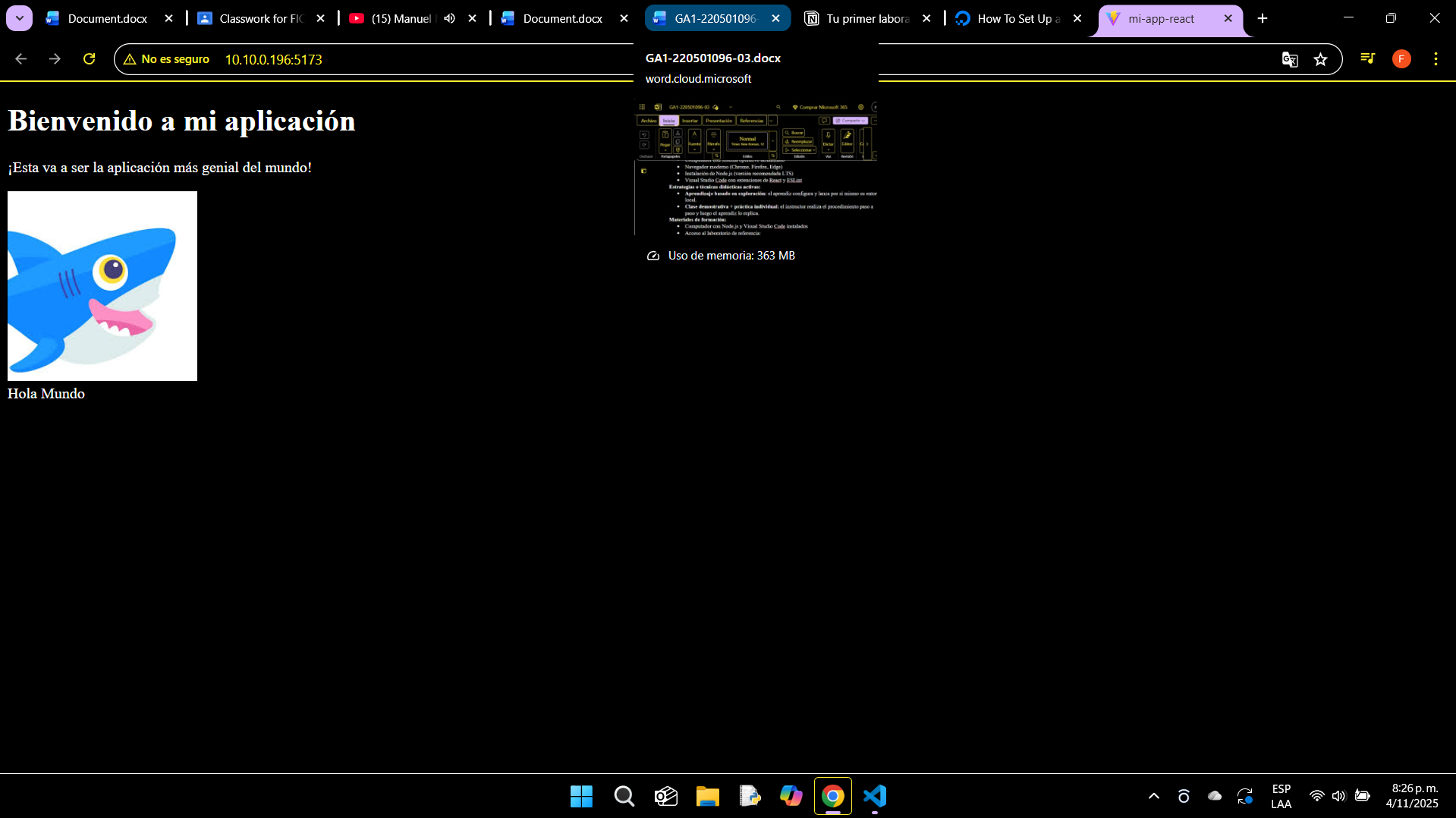


Instalacion de las dependencias 

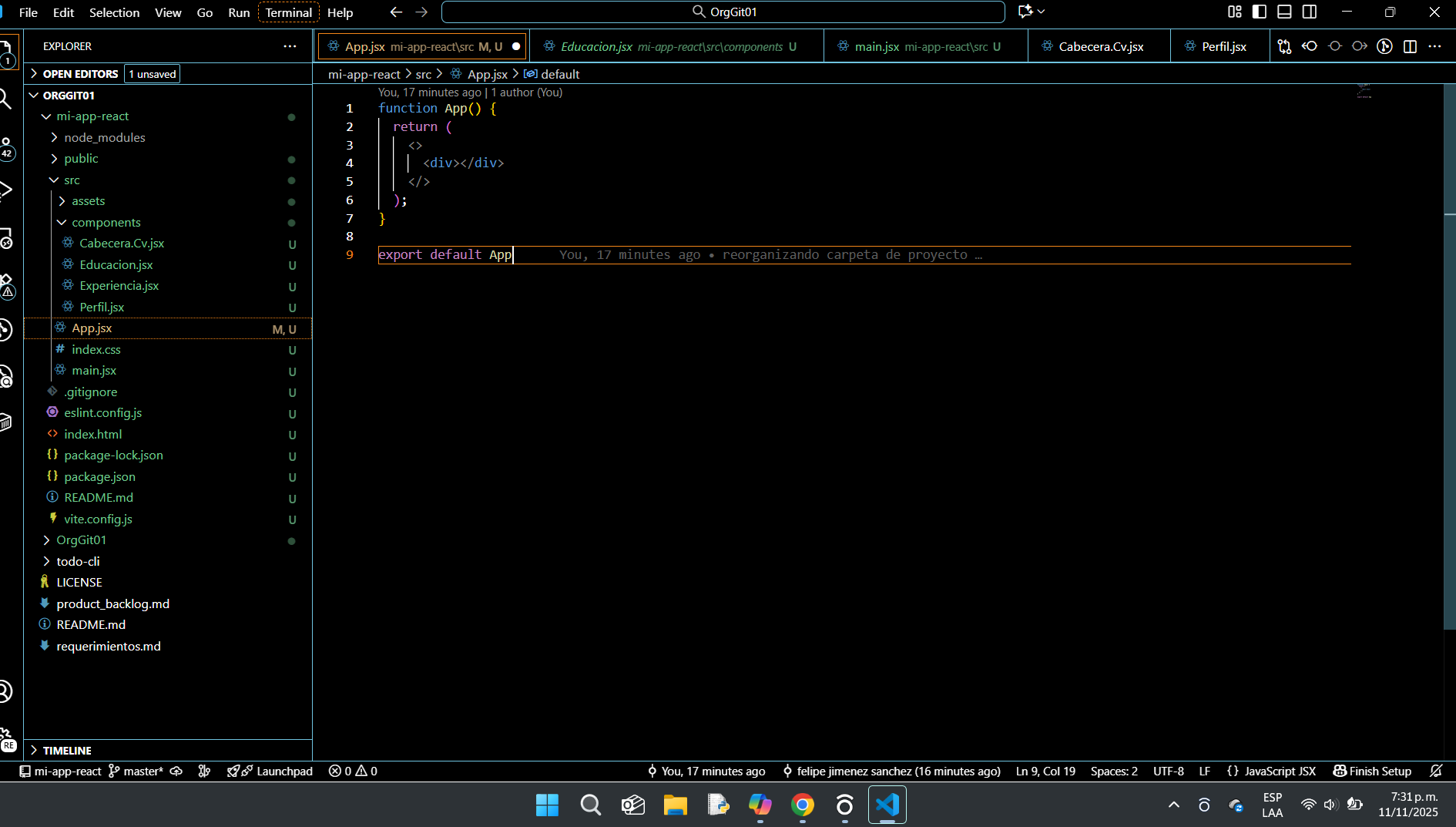
Ejecucion del servidor 

Creaccion del proyecto vite

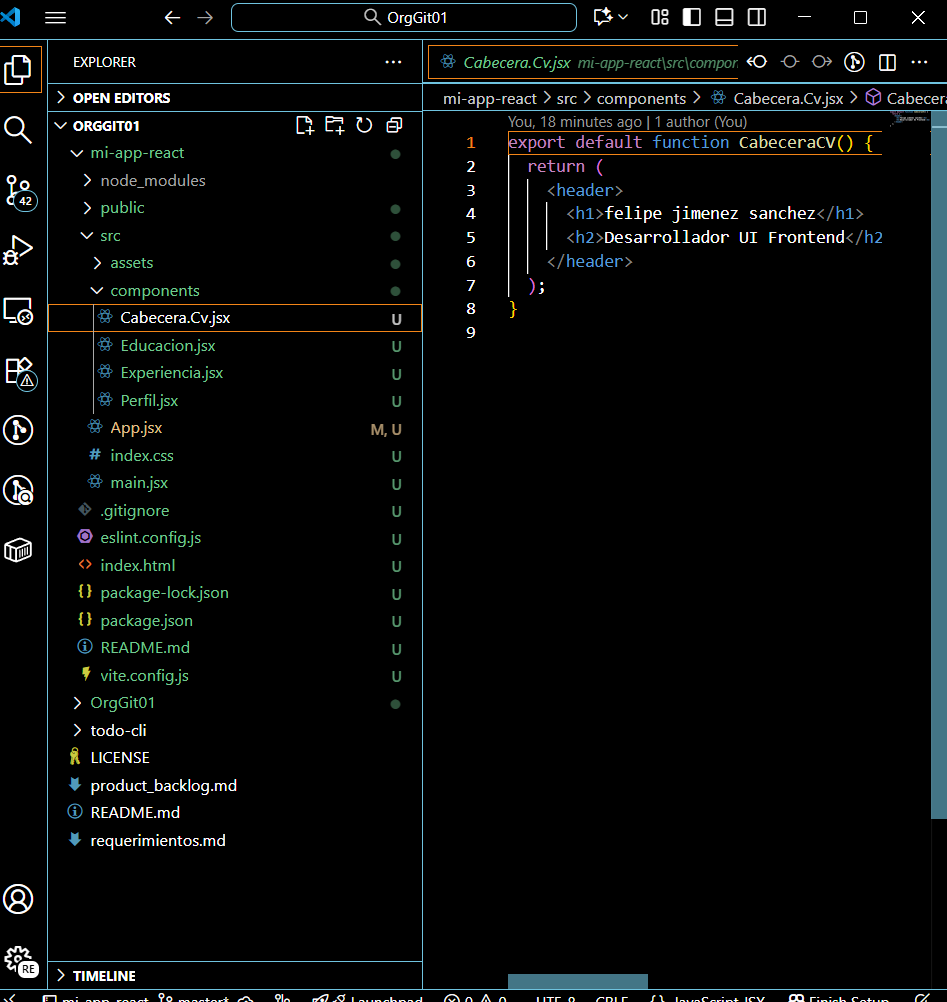


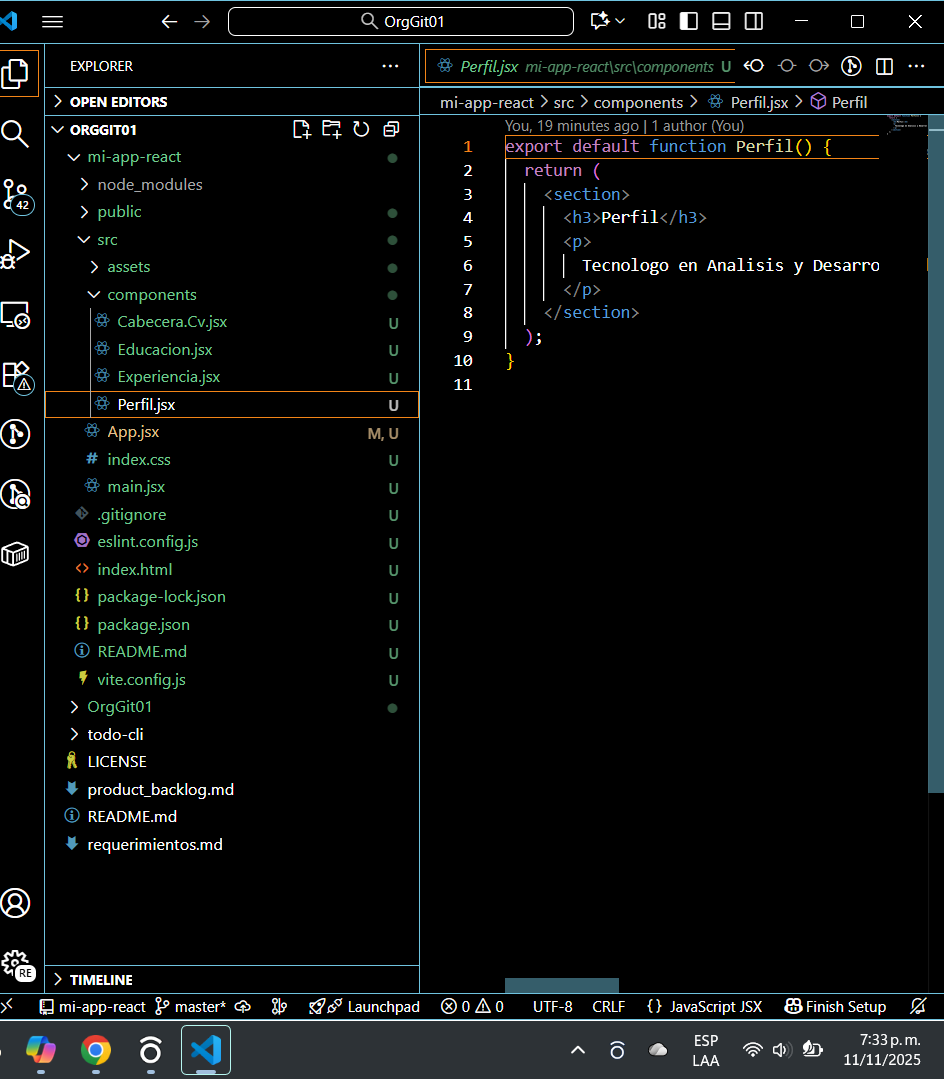


**Creación y anidación de componentes en React**

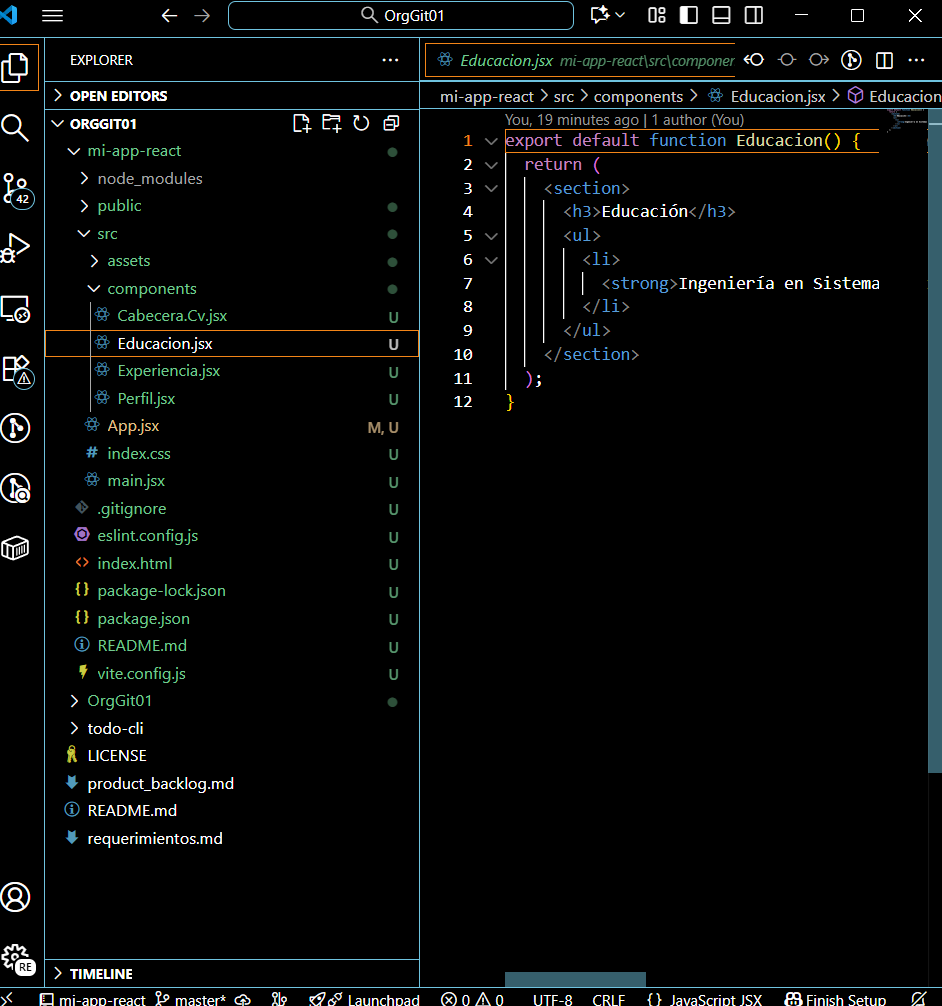


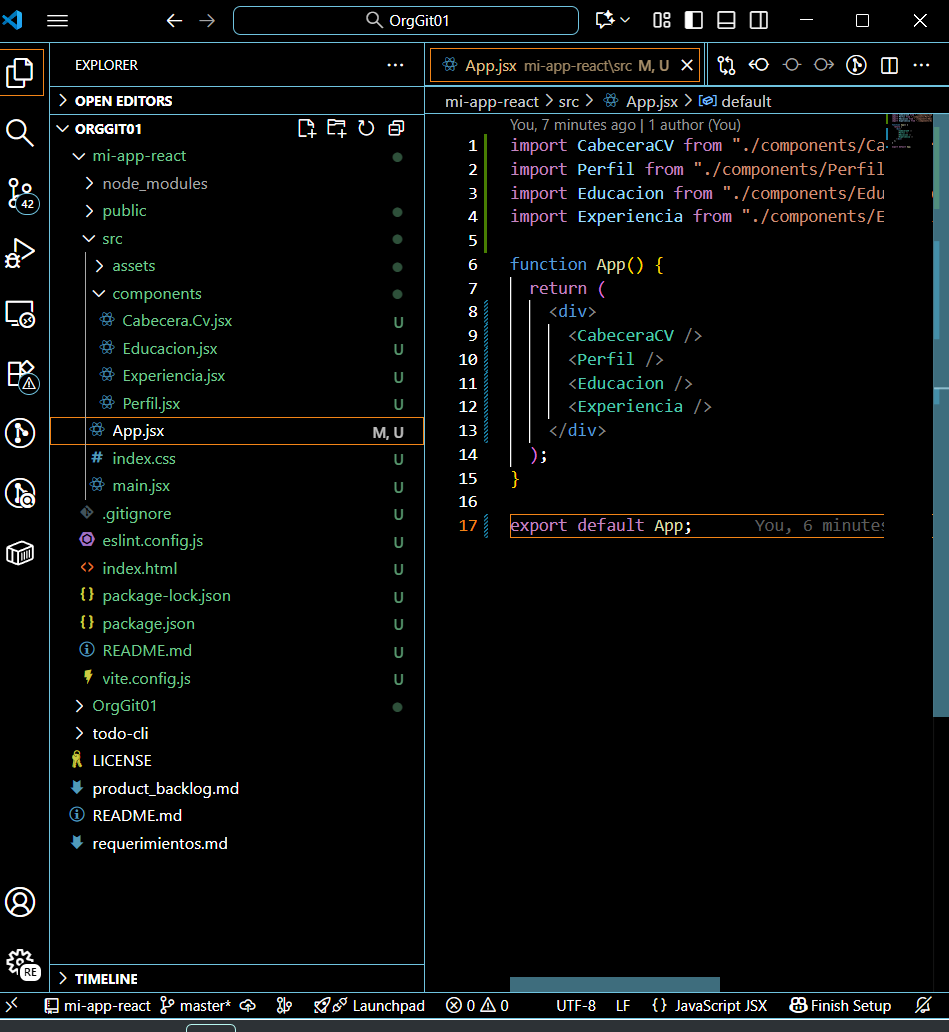


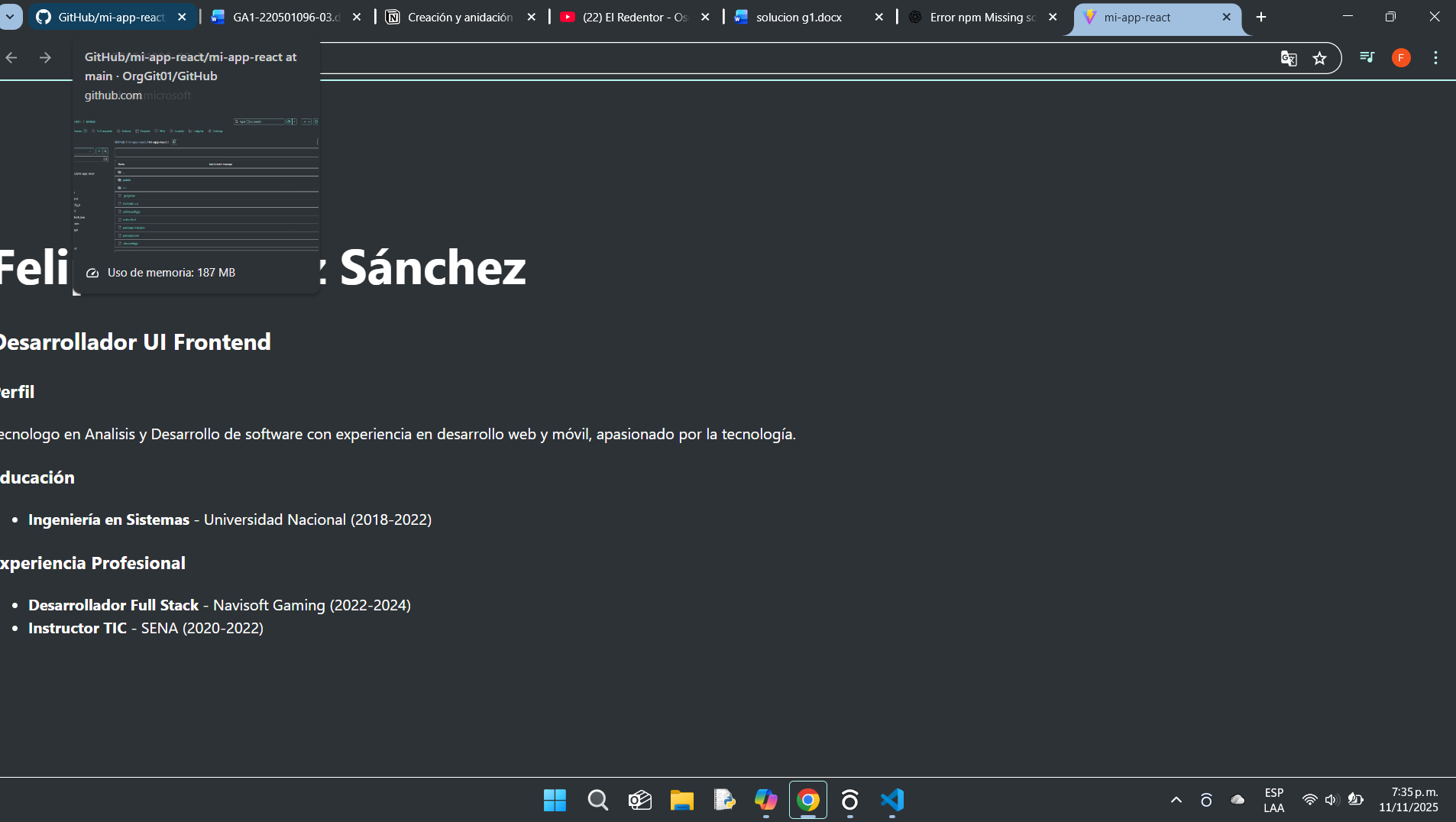




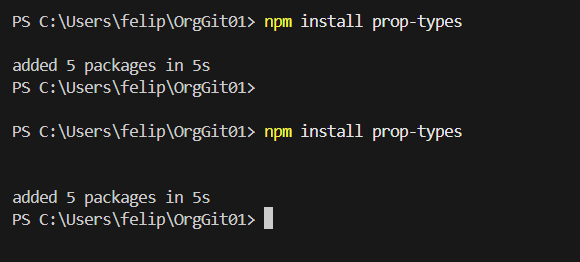
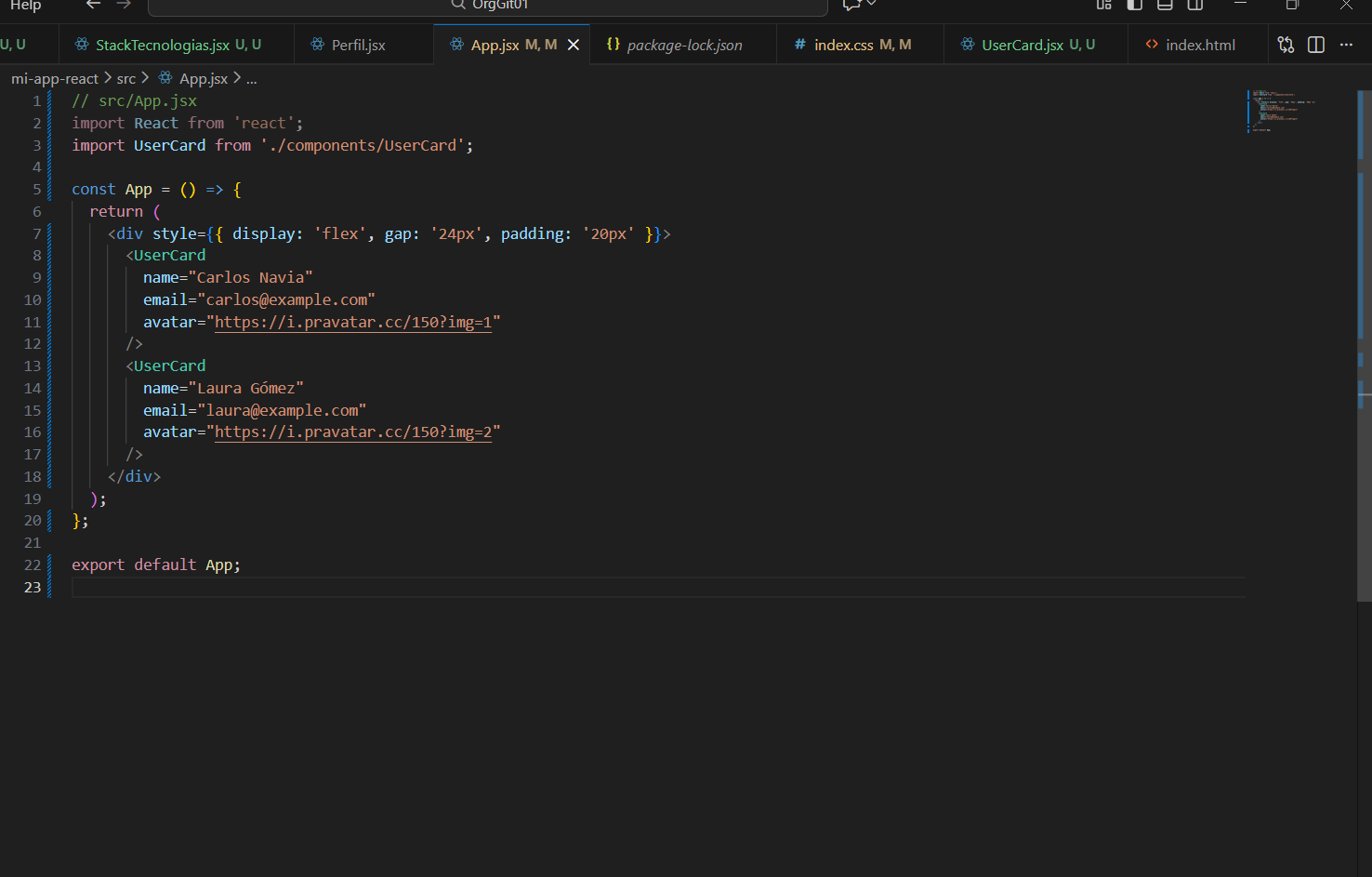
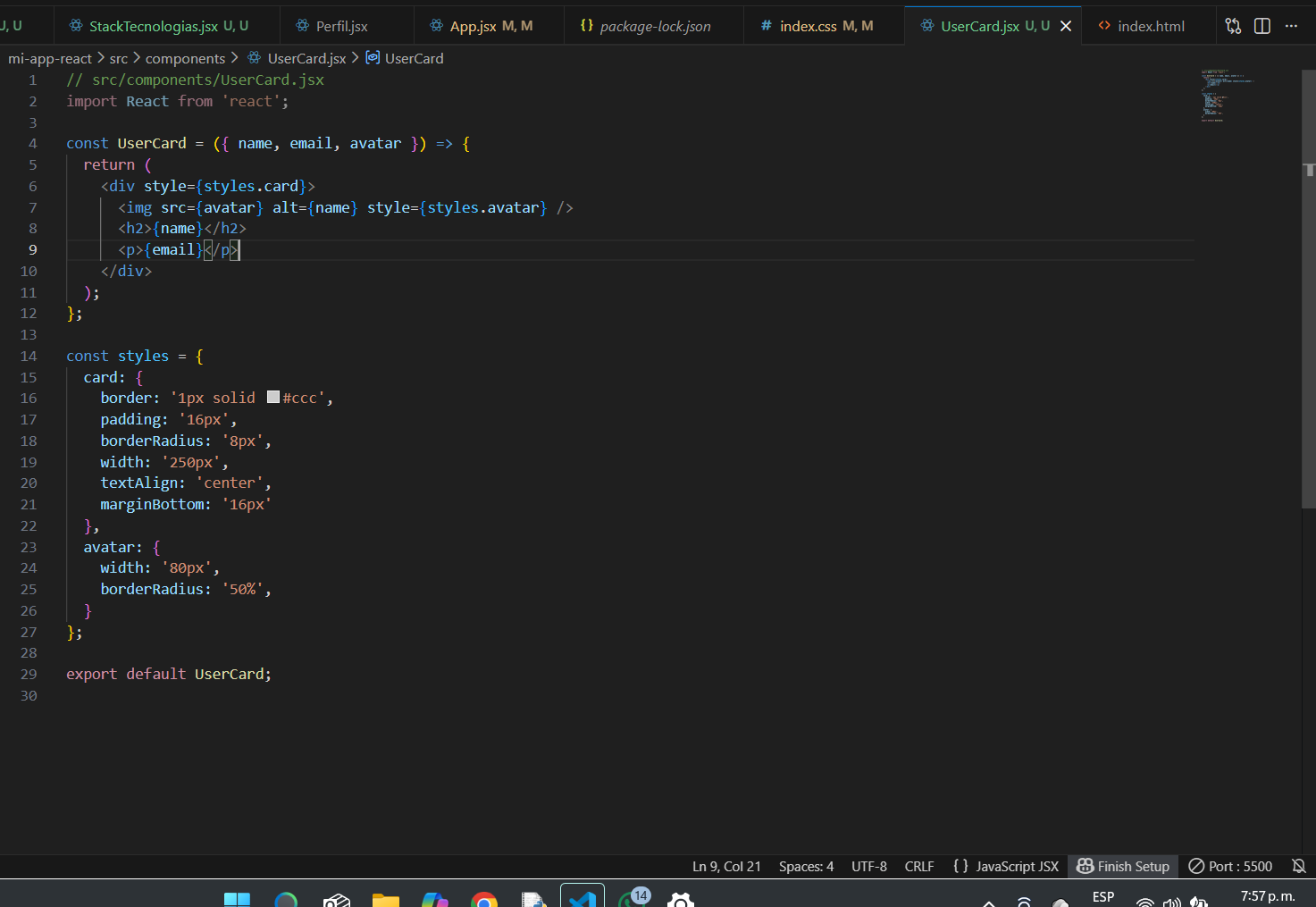








# **Componentes Reutilizables con Props en React**



# Ejercicio Práctico: Ampliando tu CV Dinámico con Props y Desestructuración.

# Ejercicio Práctico Integrador: “CV Dinámico Interactivo con Eventos y Estado Local en React”

