## Línea horizontal



Documentación práctica desarrollo web React-Angular

17/10/23

**─**

Diana Carolina Galeano Carmona

ADSO

# Descripción general

Para esta práctica de desarrollo web se optó por realizar el desarrollo con el stack MERN por lo que se usará React como framework.

en esta práctica se tendrá que realizar crud para una aplicación de notas, donde se podrán realizar creación de notas, usuarios, obtener los datos de las notas, obtener datos del usuario, eliminar notas y eliminar usuarios

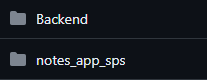
# Proceso de creación e instalación de la práctica

Creación de carpetas

Se crean las carpetas principales con la siguiente estructura: stack\_React

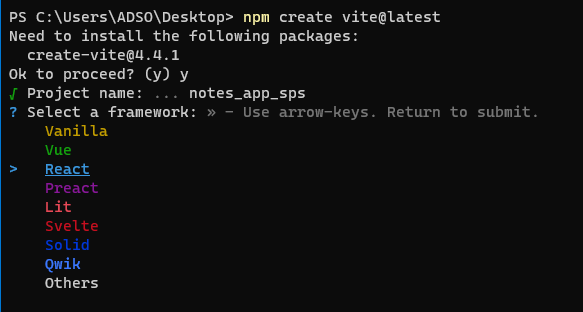


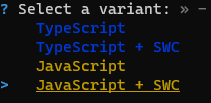
Dentro de esta carpeta crearemos 2 más una para el cliente que se llamará “notes\_app\_sps” y otra para el Backend.



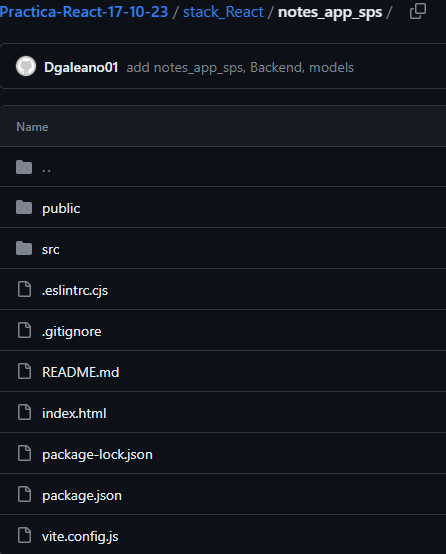
la carpeta “notes\_app\_sps” no la crearemos manualmente, en esta carpeta crearemos el proyecto con el framework elegido y será realizado con vite.

Ejecutamos el siguiente comando en una consola “npm create vite@latest” esto nos creará un proyecto vite, nos pedirá que le indiquemos el nombre del proyecto y el framework de elección, luego nos preguntará si typescript o javascript y las opciones con SWC, como queremos desplegar la aplicación vamos a seleccionar Javascript con SWC para mayor facilidad en el despliegue.





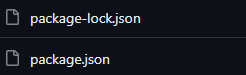
Y luego se nos creará el proyecto con el nombre que le asignamos y con la tecnología escogida



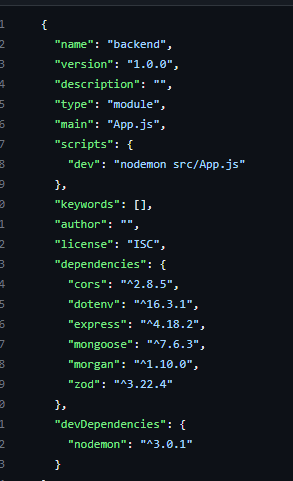
Instalación de dependencias

Ahora vamos al Backend e instalaremos los archivos básicos.

dentro de la carpeta en la consola ejecutaremos el comando “npm init” esto nos instalará los packages .json y con estos ya podremos descargar las dependencias que usaremos para este proyecto.



Para este proyecto instalaremos las siguientes dependencias “nodemon, cors, express, dotenv, mongoose, morgan, zod” para instalar estas dependencias ejecutaremos el siguiente comando “npm install cors express dotenv mongoose morgan zod” así instalaremos todas esas en una sola instalación, y para nodemon ejecutaremos “npm install nodemon -D” y así tendríamos todas las dependencias necesarias hasta el momento



añadimos unos detalles, como el “type”, lo pondremos como “module” para usar export e import, también el “main” cambiaremos el archivo dependiendo del que se cree puede ser “App.js” o “index.js” y por último en “script” añadimos la siguiente línea “”dev”:”nodemon src/App.js”” para poder ejecutar el proyecto.

# Estructura básica del Backend

Creación del servidor y conexión del puerto de ejecución

Ahora vamos a crear una carpeta que se llamara “src” y dentro de esta carpeta vamos a crear el archivo “App.js” también es conocido como “index.js”





dentro del App.js vamos a iniciar creando el servidor para el proyecto, para esto vamos a importar el express



luego creamos una variable para guardar y crear el servidor



Además de que necesitamos que lo que nos manden del servidor nos llegue como json para así poder manejar la información con más facilidad



ya con el servidor creado, procederemos a conectar con el puerto de ejecución, creamos una variable donde guardaremos el puerto que usaremos



Luego creamos una función con listen, la cual va a escuchar constantemente donde está corriendo el puerto para ejecutarlo



y como algo adicional añadiremos morgan para que cuando ejecutemos el programa nos muestre las diferentes peticiones y darnos cuenta si hay algún error y saber de qué tipo



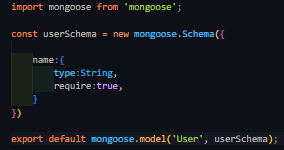
Creando models

Procederemos a crear una carpeta models en la carpeta src, aca crearemos los archivos note.model.js y user.model.js.



en estos archivos indicaremos que atributos tendrá cada modelo y que cosas son requeridas en su modelo





Creando rutas de entorno

Luego procedí a crear la variable de entorno, la cual conectara la base de datos con el servidor

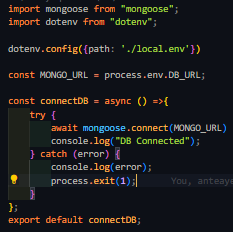
Creamos un archivo por fuera del src y este archivo lo llamé local.env y dentro de este crearemos una variable donde le asignaremos la dirección de la base de datos con su usuario y contraseña, y el nombre de la base de datos



luego en src crearemos una carpeta llamada config donde crearemos un archivo que haga referencia a la base de datos, por ejemplo “db.js”



dentro de este archivo crearemos toda la configuración de la configuracion para la base de datos



le daremos la ubicación de donde se encuentra la variable de entorno, guardaremos esta información en una variable la cual usaremos para realizar la conexión con la base de datos, luego de esto exportaremos la base de datos y la importamos en el app para que se ejecute al mismo tiempo que el servidor

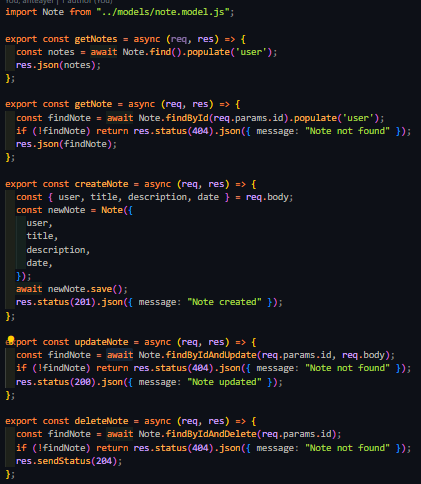


Creando Controllers

Ahora procedemos a crear una carpeta en “src” llamada “controllers”, donde crearemos los 2 archivos controladores para los modelos que creamos anteriormente



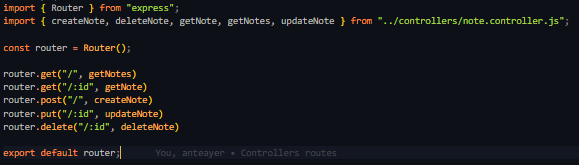
En estos controladores podremos crear las funciones de crear, editar, mostrar y eliminar las notas o usuarios

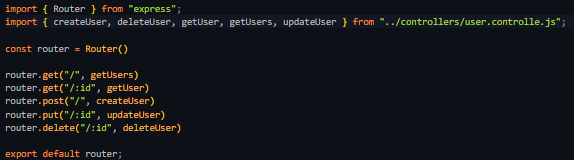




Creando Routes

Luego en “src” crearemos una nueva carpeta llamada “routes” en esta crearemos las rutas para cada verbo http e importaremos los controllers que creamos anteriormente para cada una de las funciones





Luego iremos al “App.js” y crearemos la ruta principal que será la que nos indica el puerto donde se aloja el servidor



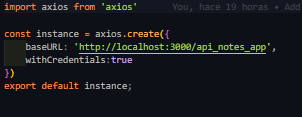
y añadiremos las rutas de cada modelo anterior para así poder visualizar sus resultados al entrar a cada ruta

# Desarrollo de Frontend

Ahora para desarrollar el Frontend iremos a la carpeta de “notes\_app\_sps”, adentro encontraremos el proyecto creado con Vite, y realizaremos algunas instalaciones antes de proceder con la creación de carpetas y componentes.

Primero descargaremos “npm i react-router-dom” para poder utilizar rutas en react, luego instalaremos axios con “npm install axios” para hacer las peticiones, luego instalaremos “npm install react-hook-form” esto lo usaremos más adelante para guardar la informacion que enviemos, y ya por último instalaremos Tailwind que será el framework de estilos que se escogió para este proyecto, seguimos todas las instrucciones que podemos encontrar en la pagina oficial de Tailwind para instalarlo con Vite, y procederemos a traer la información del backend al frontend.

Creamos una carpeta llamada “api”, dentro de esta carpeta crearemos un archivo llamado “axios.js” dentro de este archivo importaremos axios y crearemos la conexión indicando la URL base y dándole credenciales para poder enviar y recibir información del backend al frontend.

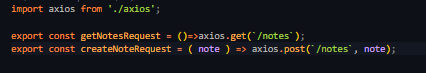


También debemos hacer una vinculacion desde el backend para indicarle que nuestra URL de frontend tiene permitido enviar y recibir información.

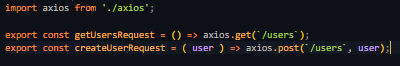


Esto irá dentro del “App.js” en el backend

Luego volvemos al proyecto de Vite y creamos las rutas para solicitar la información, crearemos los siguientes archivos “users.js” y “notes.js” en ambos realizaremos casi lo mismo.

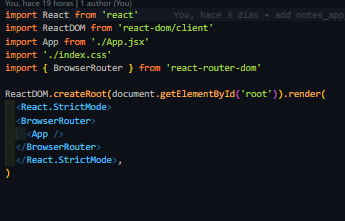


Creamos las funciones y le indicamos las rutas de como se usara la información, y esto mismo haremos con “users.js”



Ahora vamos a dejar esa carpeta por un rato, he iremos a crear las rutas.

Vamos al main y añadimos lo siguiente

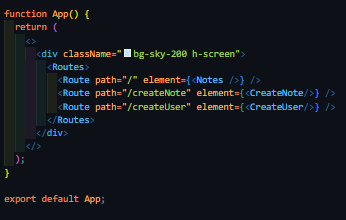


Vamos a importar el “BrowserRouter” de react-router-dom y añadiremos esta etiqueta por fuera del “App” component y por dentro del “React.StrinctMode”.

Ahora iremos a “App.jsx” y borraremos todo lo que está dentro del componente, e importaremos lo siguiente



y dentro del componente vamos a añadir lo siguiente



Aca estamos creando las rutas dandoles sus respectivas direcciones url y añadiendo el componente correspondiente, aunque aun no los hemos creado.

Ahora crearemos en la carpeta “src” la carpeta “pages” dentro de esta crearemos todos los componentes y las vistas que necesitaremos para el proyecto, iniciaremos creando el componente de “NavBar.jsx” que tendra la siguiente estructura.



Dentro del componente estamos creando todos los estilos y también importamos “Link” para crear vínculos para las rutas y poder viajar entre los componentes con el NavBar.

Ahora crearemos el componente principal que será la primera pestaña al iniciar, en este caso la llamaremos “Notes.jsx” ya que sera donde se renderizaran las notas que creemos.



Importamos el “NavBar” para renderizar al mismo tiempo que el componente de las notas.

ahora para dejar listo la pagina principal vamos a crear la plantilla de las notas, creamos un nuevo componente llamado “Cards.jsx” en la misma carpeta de “pages”



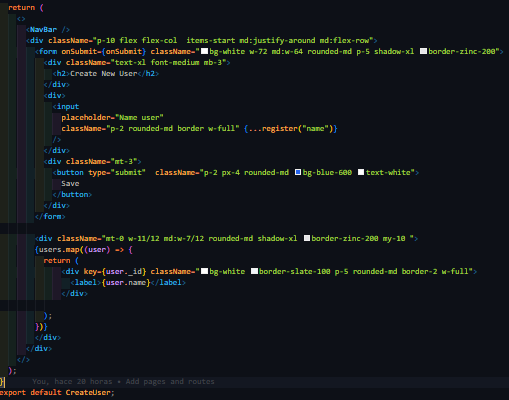
Acá estamos creando el estilo de la tarjeta que será renderizada, y tenemos un .map que recorrerá cada elemento de la tarjeta mostrando la información que traeremos del backend.



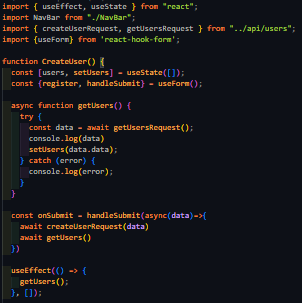
Acá estamos creando el proceso de obtener la información que tengamos guardada en la base de datos y la conexión con el backend.

Luego crearemos un el componente de “CreateUsers.jsx” en la carpeta “pages”, ahora haremos algo muy similar con los componentes anteriores.

En este crearemos un formulario para añadir el nombre y una tabla para mostrar los nombres creados.

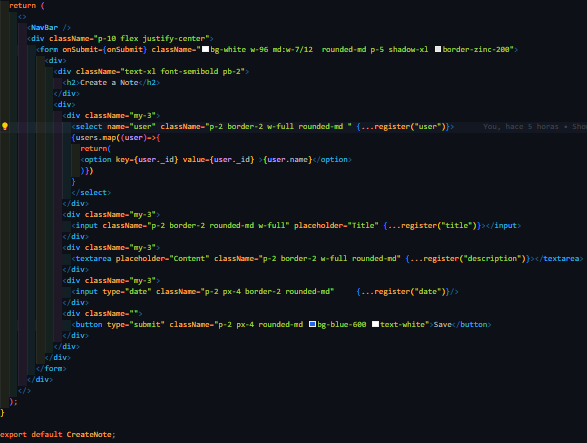


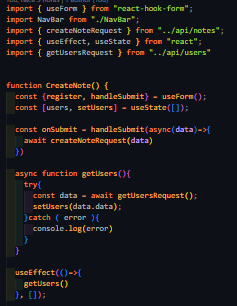
aca estamos creando el estilo del formulario y la tabla que renderizar los nombres,



Acá estamos obteniendo la información y usamos el “useForm” que proviene del “react-hook-form” para registrar la información que estamos ingresando en el formulario y esta se guarde y pueda ser mostrada en la tabla de los nombres.

Ahora crearemos el componente de “CreateNotes.jsx”, acá vamos a crear las notas que se mostraran en la página principal, y es muy parecido al componente anterior de “CreateUsers.jsx”, Crearemos un formulario donde registraremos el título, seleccionaremos el usuario responsable de esta nota, añadiremos una descripción a la nota y le daremos una fecha, esta información la guardaremos y será mostrada en el componente de “Cards.jsx” que hicimos anteriormente.





con esto hemos terminado de crear la parte de Frontend, con esto podemos mostrar y crear información en nuestra aplicación.