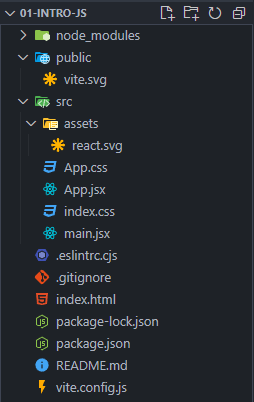
Variables y constantes

En esta sección nos vamos a sumergir en la estructura inicial del proyecto y se realizarán algunos ajustes para comenzar a trabajar con ejemplos prácticos de JavaScript. Se empieza con dos conceptos fundamentales: variables y constantes. Estos son pilares básicos en la programación y son esenciales para comprender el lenguaje.

Estructura predeterminada del proyecto

Cuando se crea un nuevo proyecto con Vite, se genera una estructura de directorios y archivos por defecto.



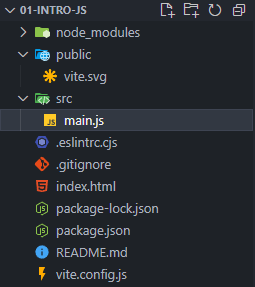
Esta estructura sirve como base para la aplicación:

* **node\_modules:** Contiene todas las librerías y dependencias instaladas, generado después de ejecutar **npm install**.
* **public:** Aquí se encuentran los recursos estáticos como imágenes y archivos JavaScript personalizados.
* **src:** Contiene todo el código de la aplicación React escrito en JavaScript.
* **index.html:** Actúa como el punto de entrada principal para la aplicación React.
* **package.json:** Contiene la lista de dependencias del proyecto y otra información relevante.
* **vite.config.js:** Contiene configuraciones de Vite como plugins, alias y configuraciones de servidor de desarrollo.

Limpiando el proyecto

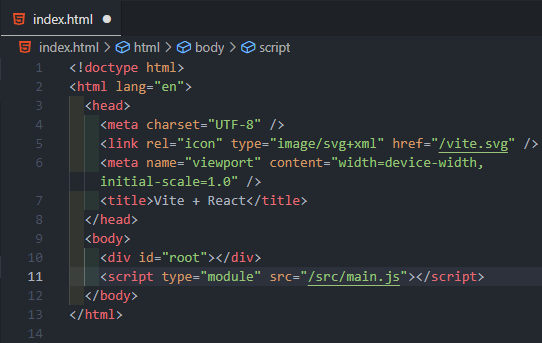
Antes de comenzar a trabajar, se procede a limpiar el proyecto:

1. Elimina los siguientes archivos: **app.css**, **app.jsx**, **index.css** y la carpeta **assets**. Puedes seleccionar varios archivos manteniendo pulsada la tecla **SHIFT** y luego haz clic derecho en uno de los archivos seleccionados y selecciona **Eliminar**.
2. Cambia el nombre del archivo **main.jsx** a **main.js**. Simplemente haz clic derecho en el archivo y selecciona **Cambiar Nombre**.



Estableciendo la ruta del archivo principal

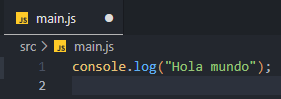
1. Abre el archivo **main.js** y elimina todo el código fuente, ya que este archivo actúa como la puerta de entrada a la aplicación de React. Luego, abre el archivo **index.html** y modifica el atributo **src** de la etiqueta **<script>** para que apunte a **main.js** en lugar de **main.jsx**.



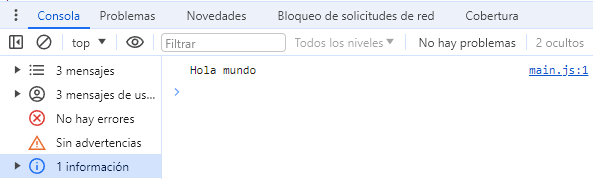
1. Guarda todos los cambios realizados pulsando la combinación de teclas **CTRL + S** en ambos archivos (**main.js** e **index.html**).
2. Al ejecutar el proyecto, escribiendo el comando **npm run dev** en la terminal, verás una pantalla en blanco en el navegador, lo que indica que el proyecto está limpio y listo para comenzar a escribir el código de la aplicación React en el archivo **main.js**.

Imprimir mensaje en la consola

La función **console.log** se utiliza para imprimir mensajes, variables o cualquier tipo de datos en la consola del navegador.



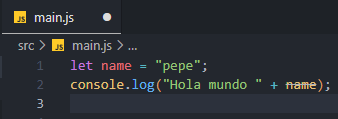
Puedes ver el resultado en la consola del navegador. Asegúrate de que el proyecto esté en ejecución con **npm run dev** y luego accede a la dirección [**http://localhost:5173/**](http://localhost:5173/). Abre la consola en el navegador con la tecla F12.

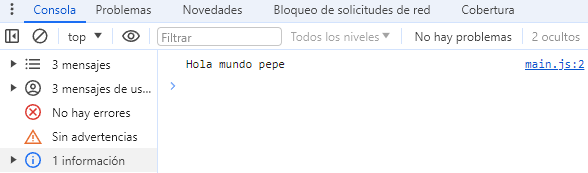


**Nota:** Es importante tener en cuenta que cada vez que se guarden los cambios en el código fuente, estos cambios se verán reflejados automáticamente en el navegador sin necesidad de volver a cargar la página.

Variables

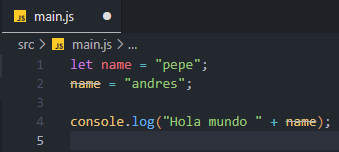
Antiguamente, se utilizaba el término **var** para declarar variables en JavaScript. Sin embargo, actualmente se recomienda utilizar **let** para declarar una variable.

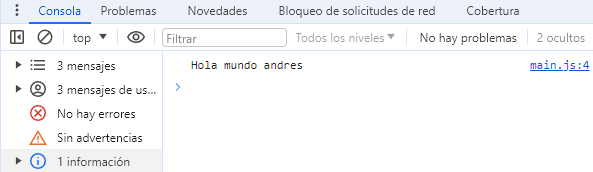




Reasignación de valores

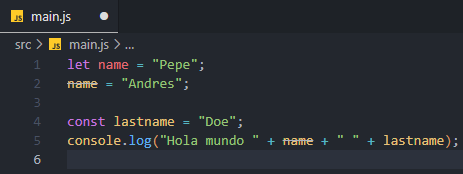
Es posible asignar un nuevo valor a una variable declarada con **let**. Es importante tener en cuenta que, si una variable tiene varios valores asignados en diferentes momentos, se utilizará el último valor asignado.

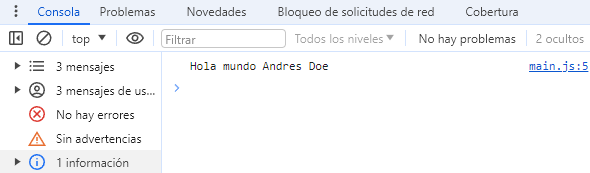




Constantes

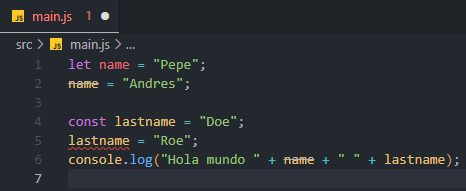
Para declarar una constante en JavaScript, se utiliza la palabra clave **const**.

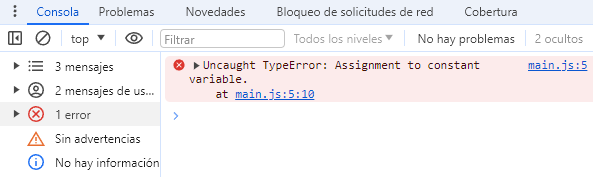




Diferencia entre variables y constantes

La principal diferencia entre **let** y **const** es que **const** se utiliza para declarar variables cuyos valores no cambiarán a lo largo del tiempo. Aunque el valor de una constante no puede ser reasignado, si la constante es un objeto o un array, sus atributos o elementos sí pueden ser modificados.





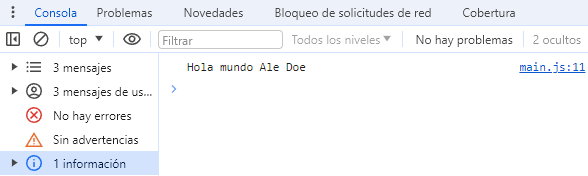
El error mostrado en la consola indica que no se puede reasignar un valor a la constante.

Definición de una variable dentro de un condicional

Normalmente el condicional **if** se utiliza para ejecutar un bloque de código si una condición especificada es verdadera. Si se declara una variable dentro de una condición que va a ser verdadera, entonces la variable va a tomar el valor que se encuentra dentro del condicional.

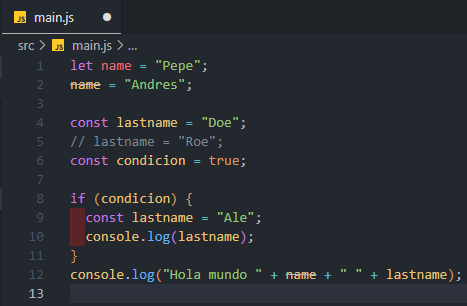


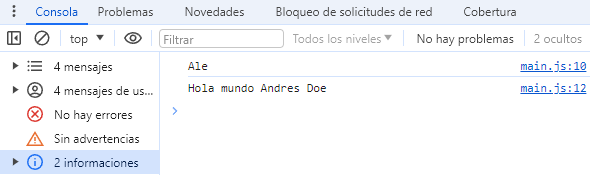




Definiciones de constantes en diferentes contextos

Una característica interesante de JavaScript es que puedes definir una constante dentro del contexto de un **if**. Esto crea un nuevo ámbito (scope) para esa constante, lo que significa que la constante definida dentro del **if** no afectará a la constante con el mismo nombre en el ámbito exterior.





En este ejemplo, la constante **lastname** definida dentro del **if** tiene un ámbito limitado a ese bloque, por lo que no afecta a la constante **lastname** definida en el ámbito global.