



React

Clase 10

KOICA

IGU HANDONG GLOBAL
UNIVERSITY



UNA

OBJETIVOS DE LA CLASE 10

- Manejar los conceptos de React, JSX Y Vite.
- Implementar los códigos de ejemplos propuestos en clase.



¿Qué es el DOM?

- Cuando abrís una página web en tu navegador (como Chrome), lo que ves es una interfaz gráfica generada a partir de un archivo HTML.
- Ese archivo HTML se convierte internamente en una estructura de árbol llamada:

 DOM → Document Object Model

- El DOM es una representación del HTML en memoria, que permite a JavaScript leer y modificar los elementos visuales del sitio.



Ejemplo:

- El HTML:

```
<h1>Hola</h1>
```

- Se convierte en algo así internamente:

```
document
```

```
└─ html
```

```
└─ body
```

```
└─ h1 → "Hola"
```



¿Y cómo lo modifica JavaScript?

- Con código como este:

```
document.querySelector("h1").textContent = "Hola mundo";
```

- Pero cuando hay muchos cambios en la página, escribir y mantener ese JavaScript se vuelve difícil.

¿Qué es React?

React es una biblioteca que nos ayuda a crear interfaces web de manera más ordenada, sin escribir directamente sobre el DOM.

Ventajas:

- Usa componentes reutilizables.
- Se encarga de actualizar la pantalla por nosotros.
- Es muy eficiente gracias al DOM virtual.

¿Qué es el DOM virtual?

Es una copia en memoria del DOM real.

React primero hace los cambios en ese DOM virtual, y luego compara con el DOM real para actualizar solo lo necesario.

Esto lo hace:

- ✓ Más rápido
- ✓ Más eficiente
- ✓ Menos errores visuales

¿React es lo mismo que React.js?

Sí. A veces se le llama simplemente "React", otras veces "React.js". Son lo mismo.



¿Por qué no usamos solo HTML y JS?

Porque en React usamos una sintaxis especial (JSX) y módulos, que el navegador no entiende directamente.

Vite traduce todo eso automáticamente y lo muestra al navegador.

¿Qué es Vite?

Vite (se pronuncia *vit*, como “rápido” en francés) es una herramienta para crear y ejecutar proyectos web modernos.

React no se usa solo, necesita herramientas que:

- Organicen los archivos,
- Transformen el código moderno (JSX),
- Y recarguen los cambios rápido mientras desarrollás.

Antes se usaba Create React App

— — —

Pero era lento. Vite es:

- ⚡ Súper rápido al iniciar.
- 🧪 Más fácil de configurar.
- ✅ Compatible con React.

¿Qué necesito tener?



Instalaciones necesarias::

- Node.js (versión 18 o superior)
- Un editor de código como Visual Studio Code

Crear tu primer proyecto React con Vite

Abrí la terminal o consola de tu sistema operativo y escribí:

```
npm create vite@latest
```

Te pedirá:

- Nombre del proyecto: mi-app-react
- Framework: seleccioná React
- Variant: elegí JavaScript (más adelante podés probar TypeScript)

Realizar el comando

```
cd proyecto001
```

 ¿Qué hace?

- cd significa “cambiar de carpeta” (viene del inglés change directory).
- Te metés dentro de la carpeta de tu nuevo proyecto.

 Es como decirle a la compu:

“Ahora quiero trabajar dentro de la carpeta proyecto001”.

 Después de esto, estás “dentro” del proyecto.

Realizar el comando

```
npm install
```

 ¿Qué hace?

Le dice a Node.js:

“Instalá todos los archivos que este proyecto necesita para funcionar”.

Realizar el comando

```
npm install
```

🧠 Cuando creás el proyecto, se genera un archivo llamado package.json que lista todos los paquetes necesarios. Este comando:

- Lee ese archivo
- Descarga los paquetes
- Crea una carpeta llamada node_modules con todo lo necesario

Realizar el comando

```
npm install
```



Es como llenar tu mochila con los materiales antes de ir a clase.



Este paso es necesario una sola vez, cuando creás el proyecto por primera vez.

Realizar el comando

```
npm run dev
```

 ¿Qué hace?

Le dice a Vite:

“Arrancá el servidor para que yo pueda ver mi proyecto en el navegador”.

 Vite va a mostrar la página en `http://localhost:5173` (puede variar).

Realizar el comando

```
npm run dev
```



Además:

Si cambiás el código, la página se actualiza automáticamente.



Este comando sí lo vas a usar cada vez que quieras trabajar en el proyecto.



Resumen visual

Comando	¿Qué hace?	¿Cuándo lo usás?
cd mi-app-react	Entrás a la carpeta del proyecto	Al comenzar
npm install	Instala los archivos que React necesita	Solo una vez
npm run dev	Inicia el servidor para ver tu sitio	Cada vez que quieras trabajar

Ahora abrí tu navegador en

```
http://localhost:5173
```

¡Tu primera app React está corriendo!

Entender la estructura del proyecto

— — —

Tu proyecto tendrá carpetas como estas:

proyecto001/

- |— public/ → Archivos estáticos
- |— src/ → Tu código fuente
 - | |— App.jsx → Componente principal
 - | |— main.jsx → Punto de inicio de la app
- |— index.html → HTML principal
- |— package.json → Dependencias y scripts
- |— vite.config.js → Configuración de Vite

◆ 1. public/ → Archivos estáticos

📌 Esta carpeta guarda archivos que no cambian y que se pueden usar directamente en el navegador.

Ejemplos:

- Imágenes (logo.png)
- Íconos (favicon.ico)
- Archivos PDF
- Fuentes (tipografías)
- ◆ Si ponés un archivo en public/, lo podés usar con una ruta como:

```

```

◆ 2. src/ → Tu código fuente

📌 Todo tu código de React vive acá.

Aquí ponés los componentes, hojas de estilo, funciones, etc.

Dentro de src/ tenés:

✅ App.jsx → El componente principal

🧠 Es como decir:

“Esto es lo que quiero que se vea en la pantalla”.

Vas a editar este archivo casi siempre.

```
function App() {  
  return <h1>Hola Mundo</h1>;  
}
```

```
export default App;
```




main.jsx → Punto de inicio de la app



Este archivo le dice a React:

"Usá el componente App"

"Mostralo dentro del <div id='root'> que está en el index.html"

Nunca lo vas a tocar mucho, pero es muy importante.

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import App from './App';  
  
ReactDOM.createRoot(document.getElementById('root')).render(<App />);
```

◆ 3. index.html → HTML principal

— — —

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title>Mi App</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

◆ 3. index.html → HTML principal

📌 Este es el archivo que el navegador ve.

👁️ Pero no tiene nada visible... solo un div vacío:

```
<div id="root"></div>
```

💡 Ese div es donde React va a dibujar tu app usando JavaScript.

4. package.json → Lista de dependencias y comandos

— — —

Este archivo es como un carnet de tu proyecto. Dice:

Qué librerías estás usando (React, Vite, etc.)

Qué versión tiene cada una

Qué comandos podés ejecutar (npm run dev, etc.)

Ejemplo de comandos:

— — —

```
"scripts": {  
  "dev": "vite",  
  "build": "vite build",  
  "preview": "vite preview"  
}
```

◆ 5. vite.config.js → Configuración de Vite

📌 Es un archivo para personalizar el comportamiento de Vite.

Ejemplo:

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()],
});
```

En la mayoría de los casos, no necesitás tocarlo al principio.

Modificar tu primera app – ¡Hola Mundo!

Abrí src/App.jsx y reemplazá el código por:

```
function App() {  
  return (  
    <div>  
      <h1>¡Hola Mundo desde React con Vite!</h1>  
      <p>Mi primera aplicación de React</p>  
    </div>  
  );  
}  
  
export default App;
```



Guardá y mirá el navegador: los cambios se actualizan solos sin recargar.

Problema

— — —

Mostrar nombre

```
function App() {  
  return (  
    <div>  
      <h2>Mi nombre es Ivan</h2>  
    </div>  
  );  
}  
  
export default App;
```


Problema

— — —

Mostrar la fecha de hoy

```
function App() {  
  let fecha = new Date().toLocaleDateString();  
  
  return (  
    <div>  
      <h1>Fecha de hoy: {fecha}</h1>  
    </div>  
  );  
}  
  
export default App;
```

Problema

— — —

Mostrar un número

```
function App() {  
  const numero = 42;  
  
  return (  
    <div>  
      <h2>El número es {numero}</h2>  
    </div>  
  );  
}  
  
export default App;
```

Problema

— — —

Mostrar suma de dos números

```
function App() {  
  let a = 10;  
  let b = 5;  
  let suma = a + b;  
  
  return (  
    <div>  
      <h2>{a} + {b} = {suma}</h2>  
    </div>  
  );  
}  
  
export default App;
```

EJERCICIOS ADICIONALES PROPUESTOS

— — —





**¡MUCHAS
GRACIAS!**