

«Talento Tech»

React JS

Clase 01



Clase N° 1 | Primeros Pasos con ReactJS

Índice:

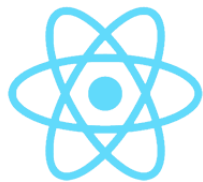
- Repaso HTML, CSS y JavaScript.
 - Instalación con Vite y Node.js.
 - Estructura básica del proyecto React.
 - Introducción a JSX y creación de componentes funcionales.
-

Objetivos de la Clase:

- Revisar los fundamentos de HTML, CSS y JavaScript.
- Comprender el propósito de ReactJS y su lugar en el desarrollo web moderno.
- Configurar un proyecto inicial de React con Vite.
- Crear los primeros componentes funcionales usando JSX.

Bienvenida y presentación del curso

¡Bienvenidos al curso de **React**!



React

En este curso, exploraremos **React**, una de las bibliotecas más populares para construir interfaces de usuario en aplicaciones web. Aprenderemos desde los conceptos básicos, como componentes

y estados, hasta temas más avanzados como el manejo de ciclos de vida, hooks, y gestión del estado global.

Nuestro objetivo es ayudarlos a dominar React para que puedan desarrollar aplicaciones interactivas, modernas y escalables. Este curso está diseñado para que avancen paso a paso, con ejemplos prácticos y ejercicios dinámicos.

Repaso de HTML, CSS y JavaScript: : ¿Qué son y cómo funcionan?

HTML, CSS y JavaScript son la base del desarrollo web y trabajan juntos para dar vida a las páginas que navegamos todos los días.

HTML



HTML (HyperText Markup Language) es el lenguaje que utilizamos para estructurar el contenido de una página web. Piensa en HTML como el esqueleto de la página: define qué elementos aparecen, como títulos, párrafos, imágenes o botones. Por ejemplo, si queremos mostrar un encabezado y un botón, escribiríamos:

Ejemplo práctico:

1. HTML Básico

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

<button>Haz clic aquí</button>
```



CSS (Cascading Style Sheets) es lo que usamos para estilizar y dar formato visual al contenido definido con HTML. Con CSS podemos cambiar colores, tamaños, tipografías, márgenes y mucho más, logrando que nuestras páginas sean atractivas y fáciles de usar. **Por ejemplo, para estilizar un botón, podríamos usar:**

2. CSS para diseño:

```
button {  
  
  background-color: blue;  
  
  color: white;  
  
}
```



JavaScript es el lenguaje que añade interactividad a nuestras páginas. Es lo que hace que los botones respondan a los clics, que aparezcan mensajes o que cambien dinámicamente los datos en pantalla. JavaScript permite que las páginas sean dinámicas y funcionales, como en **este ejemplo donde mostramos una alerta al hacer clic en un botón:**

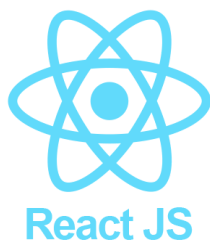
3. JavaScript interactivo:

```
document.querySelector('button').addEventListener('click', () => {  
  
  alert('¡Botón clickeado!');  
  
});
```

Además de estas funciones básicas, JavaScript también nos permite trabajar con estructuras más avanzadas como **arrays** y **objetos**, que son fundamentales para almacenar y manipular datos. **Por ejemplo:**

```
const numeros = [1, 2, 3, 4];  
const dobles = numeros.map(num => num * 2); // [2, 4, 6, 8]  
  
const persona = { nombre: "Ana", edad: 25 };  
console.log(persona.nombre); // Ana
```

¿Qué es ReactJS? Introducción y beneficios



ReactJS es una biblioteca de JavaScript desarrollada por Meta (antes Facebook) que nos permite **construir interfaces de usuario dinámicas de una forma sencilla y estructurada**. Su principal característica es la capacidad de dividir la interfaz en pequeños bloques reutilizables llamados componentes, lo que mejora la organización y mantenimiento

del código.

Con React, dejamos atrás la necesidad de manipular directamente el DOM (Document Object Model) mediante JavaScript puro. En su lugar, usamos el **Virtual DOM**, una representación virtual que optimiza las actualizaciones en la interfaz, haciendo que nuestras aplicaciones sean más rápidas.

React también facilita la gestión del estado de nuestra aplicación mediante un flujo de datos unidireccional, lo que significa que los datos fluyen de arriba hacia abajo, manteniendo un control claro y predecible de la lógica.

Entre sus ventajas destacan:

- **Componentización:** Permite dividir interfaces complejas en piezas más pequeñas y reutilizables.
- **Desempeño optimizado:** Gracias al Virtual DOM, las actualizaciones son más eficientes.
- **Gran ecosistema:** Documentación, herramientas, y una comunidad amplia que facilita el aprendizaje y la resolución de problemas.

Un ejemplo sencillo de React sería un componente que muestra un saludo:

```
import React from 'react'; function Saludo() {  
  
  return <h1>¡Hola desde React!</h1>;  
  
} export default Saludo;
```

Configuración del entorno con Vite y Node.js



Antes de empezar con React, necesitamos un entorno de desarrollo configurado. **Node.js** es una herramienta esencial porque nos permite usar npm (Node Package Manager) para instalar paquetes y ejecutar comandos como `npm create vite@latest`.

1. Descarga Node.js desde su [sitio oficial](#) e instálalo.
2. Abre una terminal y ejecuta el siguiente comando para crear un proyecto React:

npm create vite@latest

Selecciona React como framework y configura el proyecto.

3. Instala las dependencias del proyecto
npm install
4. Inicia el servidor local para ver tu aplicación:
npm run dev

Introducción a JSX y creación de componentes funcionales



JSX (JavaScript XML) es una extensión de JavaScript que nos permite escribir código similar a HTML dentro de nuestros archivos JavaScript. Esto facilita mucho la creación de interfaces en React.

Por ejemplo, en lugar de escribir:

```
React.createElement('h1', {}, 'Hola Mundo');
```

podemos escribir directamente:

```
<h1>Hola Mundo</h1>;
```

Los componentes funcionales son bloques reutilizables que devuelven JSX para renderizarse en la interfaz. **Aquí tienes un ejemplo:**

```
function Bienvenida() {  
  return <h1>¡Bienvenidos al curso de React!</h1>;  
}
```

Para usar este componente en nuestra aplicación:

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

Reflexión Final

¡Muy bien! Ya calentamos motores con un repaso técnico y conocimos un poco de qué trata React. Ahora contás con la experiencia para crear proyectos React y poder trabajar sobre un proyecto creado con Vite. A partir de acá será nuestra tarea ir construyendo en necesidad un producto digital que luzca moderno y sencillo para el uso en el mercado y fácil para los usuarios.

Ejercicios prácticos:

1. Crea una página HTML que incluya un botón estilizado con CSS.
2. Añade interactividad al botón usando JavaScript para mostrar un mensaje al hacer clic.
3. Configura un proyecto React con Vite e implementa un componente funcional que muestre un saludo personalizado.

Materiales y Recursos Adicionales:

- ★ [Documentación oficial de React](#)
- ★ [Guía de instalación de Node.js](#)
- ★ [Guía oficial de Vite](#)

Preguntas para Reflexionar:

- ¿Qué relación tienen HTML, CSS y JavaScript en un proyecto?
- ¿Qué ventajas crees que aporta React frente a JavaScript puro?
- ¿Cómo podrías mejorar la reutilización de código en un proyecto?

Próximos Pasos:

- JSX y creación de componentes más complejos.
- Propiedades de los componentes (props)
- Introducción a TalentoLab.



Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad