



Agencia de Habilidades  
para el Futuro

<Talento  
Tech />

# ReactJS

---

**Clase 01 | Primeros Pasos con ReactJS**

# ¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase.



# ¿Ya viste la “Introducción al programa” disponible en el campus virtual?

- ⓘ La visualización y resolución de un breve cuestionario es de carácter obligatorio para desbloquear los contenidos de las primeras 2 clases

✨ Conozcamos al Staff ✨



---

## ⭐ Staff Académico ⭐

---



Nombre y Apellido



Nombre y Apellido

Rol: XXXXX

Rol: XXXXX

Descripción del rol: XXXXX

Descripción del rol: XXXXX

Contacto: XXXXX

Contacto: XXXXXX



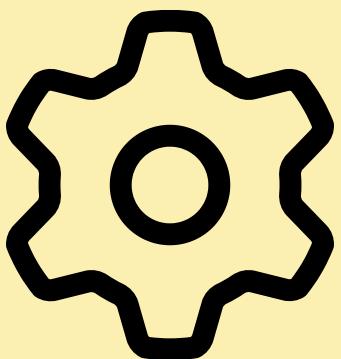
# Metodología de cursada

---

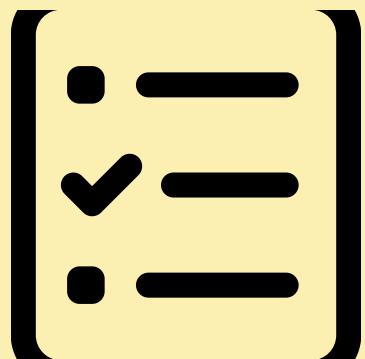
# Metodología de Evaluación

---

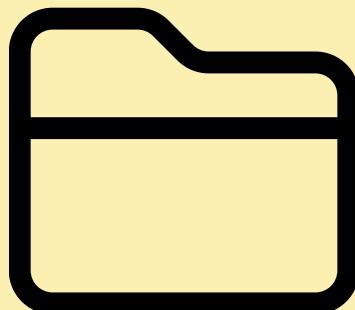
Nuestro objetivo es prepararte para enfrentar los desafíos del siglo XXI y facilitar tu inserción en el mercado laboral. Para lograrlo, hemos desarrollado un programa que enfatiza la ejercitación constante y el seguimiento continuo. A continuación, te explicamos cómo serás evaluado a lo largo de la cursada:



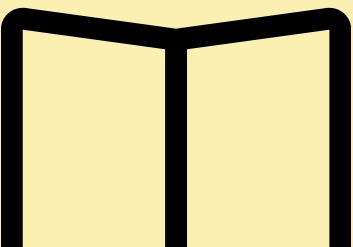
Ejercicios Prácticos



Cuestionario



Pre-Entrega





## Proyecto Final

- ⓘ La información detallada de cada evaluación está disponible en el apartado “Introducción” dentro del Campus Virtual.



# Proyecto Integrador Final

Obligatorio |  
Entregable

---

## ¿Cómo se construye el proyecto integrador?

Al final de la cursada, serás evaluado mediante la entrega de un **Proyecto Integrador**, que es fundamental para completar el curso y cumplir con los requisitos de egreso. Este proyecto se construirá de manera progresiva, combinando la resolución de **Ejercicios Prácticos** y la **Pre-Entrega de Proyecto** presentes a lo largo de la cursada.

Las consignas a evaluar en este proyecto final integrador estarán constituido por la resolución de los "**Ejercicios Prácticos**" presentes a lo largo de la cursada en función del proyecto integrador.

# After Class

---

El espacio "After Class" está diseñado para ofrecerte apoyo adicional y facilitar tu progreso durante la cursada. Aunque es opcional, te recomendamos que utilices este espacio para optimizar tu aprendizaje y el desarrollo de tu proyecto integrador.

## 💡 Beneficios de asistir:

- **Consultas y Asesoría:** Aprovecha este tiempo para resolver cualquier duda o consulta que tengas sobre el contenido de las clases, ejercicios prácticos, o cualquier aspecto relacionado con tu proyecto integrador. Podrás recibir orientación más personalizada de los/as instructores y obtener aclaraciones que te ayudarán a comprender mejor los conceptos y mejorar tu desempeño.



**Frecuencia:** Una vez por semana en un día distinto y en la franja horaria de la cursada regular.

# Enfoque inmersivo

---

En este curso utilizaremos un enfoque inmersivo como herramienta fundamental para que puedan experimentar los roles más solicitados en el mercado. Este curso va más allá de la enseñanza de programación; su objetivo es prepararlos para enfrentarse a problemas del mundo real en entornos laborales.

Este enfoque nos ayuda a:

**1 Conectar con situaciones reales.**

**2 Desarrollar pensamiento crítico.**

**3 Hacer el aprendizaje más significativo.**

**4 Vivir la experiencia de un profesional.**

- ⓘ Cada ejercicio en el curso será como un capítulo de una historia. Ustedes serán los protagonistas, y su misión será utilizar las herramientas brindadas en las clases para superar los retos que se les presenten.

# Índice

1

2

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

- Repaso de JSX y componentes funcionales.
- Creación de componentes más complejos.
- Bienvenida a TalentoLab
- Situación Inicial en TalentoLab

# Objetivos de la Clase

1

Revisar los fundamentos de HTML, CSS y JavaScript.

- 2 Comprender el propósito de ReactJS y su lugar en el desarrollo web moderno.
- 3 Configurar un proyecto inicial de React con Vite.
- 4 Crear los primeros componentes funcionales usando JSX.

# REACTJS

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.

## HTML: La Estructura Web

---

# HTML: La Estructura Web

---

## Definición

HTML es el lenguaje para estructurar el contenido de una página web.

## Función

Define qué elementos aparecen, como títulos, párrafos, imágenes o botones.

## Analogía

HTML es el esqueleto de la página web.





# Ejemplo Práctico de HTML



```
<h1>Hola Mundo</h1>
<button>Haz clic aquí</button>
```

Este código HTML crea un encabezado y un botón simple en la página web.

# CSS: El Estilo Web

---

## Definición

CSS es el lenguaje para estilizar y dar formato visual al contenido HTML.

## Función

Cambia colores, tamaños, tipografías, márgenes y más elementos visuales.

## Beneficio

Logra que las páginas sean atractivas y fáciles de usar.



# Ejemplo Práctico de CSS



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

Este código CSS estiliza un botón, dándole un fondo azul y texto blanco.

# **JavaScript**

---

**JavaScript: La Interactividad Web**

---

## **Definición**

JavaScript añade interactividad a las páginas web.

## **Función**

Hace que los elementos respondan a acciones del usuario.

## **Beneficio**

Permite que las páginas sean dinámicas y funcionales.

**JS**



# Ejemplo Práctico de JavaScript

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

Este código JavaScript muestra una alerta al hacer clic en un botón.



# JavaScript Avanzado: Arrays y Objetos

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:**

---

## Arrays

```
const numeros = [1, 2, 3, 4];const dobles  
= numeros.map(num => num * 2);
```

## Objetos

```
const persona = { nombre: "Ana", edad: 25  
};console.log(persona.nombre);
```

# **React JS**

---

**Introducción a ReactJS**

---

## **Definición**

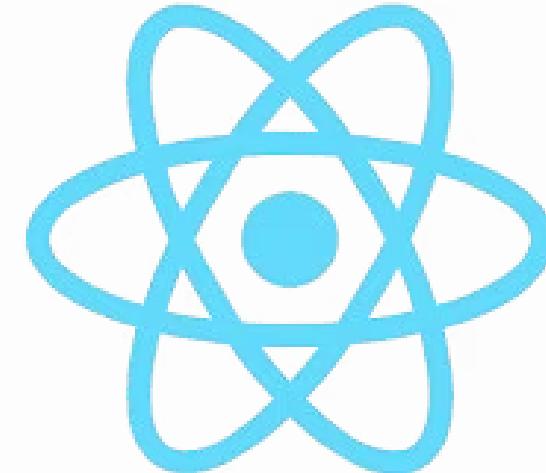
Biblioteca de JavaScript para construir interfaces de usuario dinámicas.

## **Desarrollador**

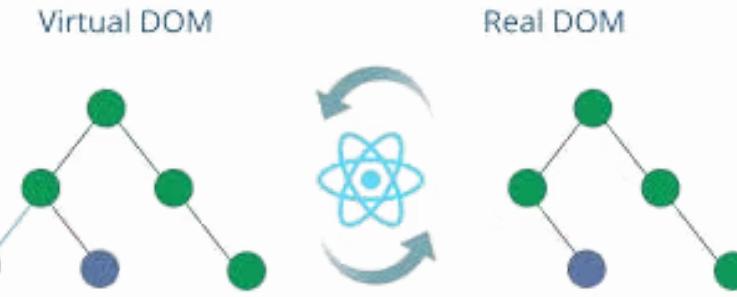
Creada por Meta (antes Facebook).

## **Característica Principal**

División de la interfaz en componentes reutilizables.



# **Virtual DOM en React**



### Concepto

Representación virtual del DOM que optimiza las actualizaciones de la interfaz.

### Ventaja

Mejora el rendimiento de las aplicaciones React.

### Funcionamiento

Actualiza solo los elementos necesarios, evitando recargar toda la página.

# Flujo de Datos en React

1

## Unidireccional

Los datos fluyen de arriba hacia abajo en la jerarquía de componentes.

2

## Predecible

Facilita el seguimiento y depuración del estado de la aplicación.

3

## Eficiente

Mejora el control de la lógica y el rendimiento de la aplicación.



# Ventajas de React

---



## Componentización

División de interfaces complejas en piezas más pequeñas y reutilizables.



## Desempeño Optimizado

Actualizaciones eficientes gracias al Virtual DOM.



## Ejemplo Básico de React

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

Este componente React simple muestra un saludo en la pantalla.

# Configuración del Entorno

---

# Configuración del Entorno: Node.js

## 1 Descarga

Visita el sitio oficial de [Node.js](#) y descarga la versión adecuada.

## 2 Instalación

Sigue las instrucciones del instalador para tu sistema operativo.

## 3 Verificación

Abre una terminal y ejecuta: `node --version` para confirmar la instalación.



# Creación de Proyecto React con Vite

## 1 Iniciar Proyecto

Ejecuta: `npm create vite@latest` en la terminal.

## 2 Configurar

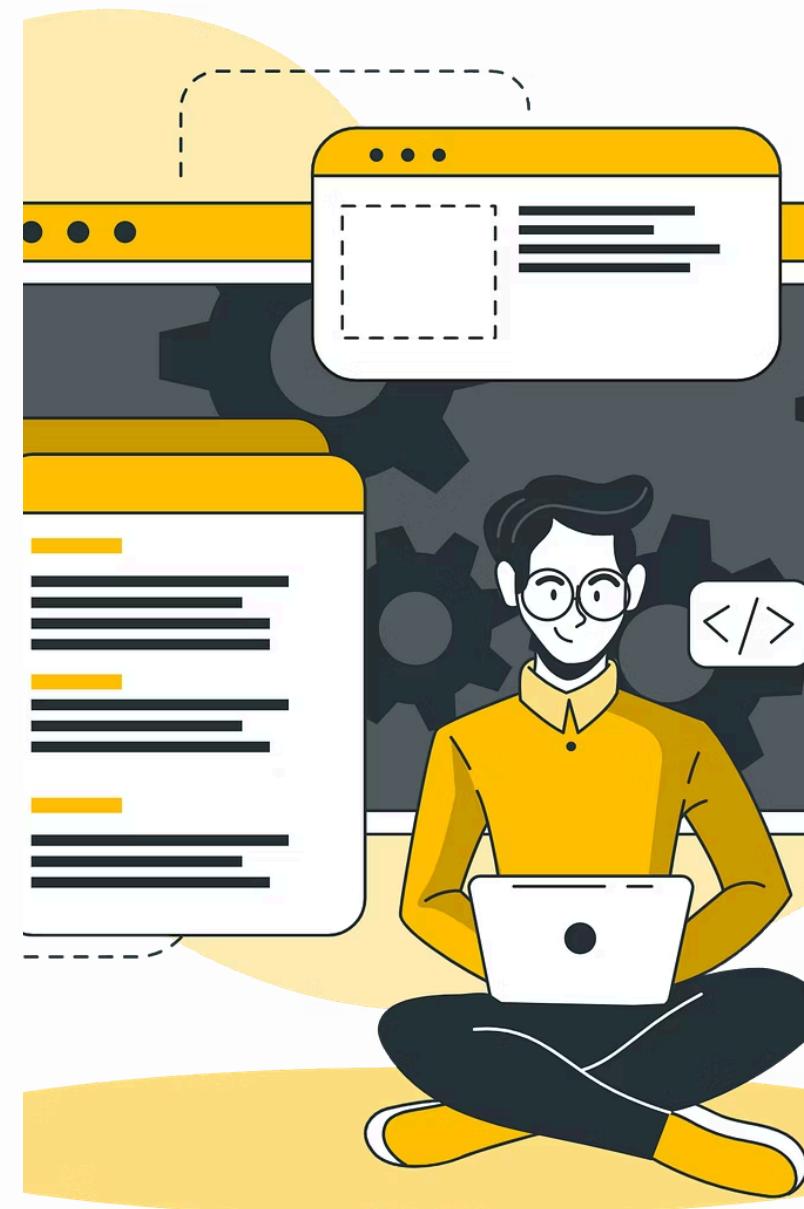
Selecciona **React** como framework y configura el proyecto.

## 3 Instalar Dependencias

Ejecuta: `npm install` en el directorio del proyecto.

## 4 Iniciar Servidor

Ejecuta: `npm run dev` para ver tu aplicación en local.



# **JSX**

---

**Introducción a JSX**

## **Definición**

JSX es una extensión de JavaScript que permite escribir código similar a HTML.

## **Ventaja**

Facilita la creación de interfaces en React de manera más intuitiva.

## **Sintaxis**

Combina JavaScript con elementos similares a HTML en un mismo archivo.

**JS**



# Ejemplo de JSX **vs** JavaScript Puro

---

## JSX

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

## JavaScript Puro

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



JSX simplifica la escritura de elementos React comparado con JavaScript puro.

## Componentes Funcionales en React

### **Definición**

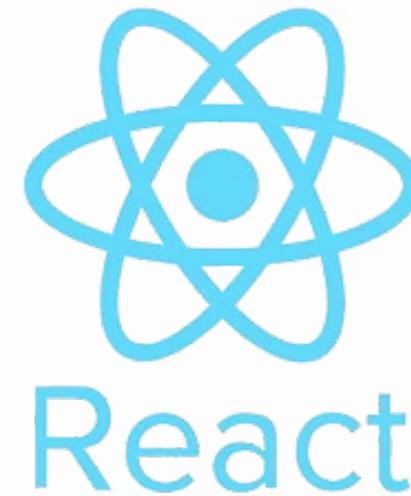
Bloques reutilizables que devuelven JSX para renderizar en la interfaz.

### **Sintaxis**

Funciones JavaScript que retornan elementos JSX.

### **Ventaja**

Facilitan la organización y reutilización del código.



# Ejemplo de Componente Funcional



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

Este componente simple muestra un mensaje de bienvenida en la interfaz.

# Uso de Componentes en la 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 />);
```

Este código renderiza el componente Bienvenida en el elemento raíz de la aplicación.



ⓘ ¡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.

# Preguntas para reflexionar



## 1 Relación Web

¿Cómo se relacionan HTML, CSS y JavaScript en un proyecto web?

## 2 Ventajas de React

¿Qué beneficios aporta React frente al uso de JavaScript puro?

## 3 Reutilización

¿Cómo podrías mejorar la reutilización de código en un proyecto?

# Próximos Pasos: Componentes Avanzados

1

**JSX y creación de componentes más complejos.**

2

**Propiedades de los componentes (props)**

3

**Introducción a TechLab**

# Recursos Adicionales



---

1

## [Documentación React](#)

Visita la documentación oficial de React para profundizar en los conceptos.

2

## [Node.js](#)

Consulta la guía de instalación de Node.js para más detalles.

3

## [Vite](#)

Explora la guía oficial de Vite para optimizar tu entorno de desarrollo.



# Ejercicios Prácticos

---



## Ejercicio Práctico

Obligatorio

## Primeros Pasos:

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.