



Nuestro compromiso es con el *futuro*.

React

Clase 1

Temas: Clase 1

Conceptos básicos:

- **Librería VS Framework**
- **Programación Funcional (FP)**
- **Métodos Funcionales**
- **¿Framework o librería?**
- **React Puro**
- **Props**

React

Intro: Librería VS Framework

Intro: Librería VS Framework

Ejemplos de librería:

- lodash
- axios
- styled-components
- jQuery

Intro: Librería VS Framework

Ejemplos de Framework:

- Angular
- Bootstrap
- Express
- Next.js

React

Intro: Programación Funcional

Intro: Programación Funcional

Características:

- Inmutabilidad
- Composición de funciones
- Higher order functions (HoF)
- Paradigma declarativo
- Funciones Puras (una entrada siempre devuelve la misma salida)

Intro: Programación Funcional

Métodos Funcionales:

- `.map()`
- `.filter()`
- `.reduce()`
- `.find()`

*Aclaración: el método `.forEach()` no es considerado un método funcional

React

Intro: ¿Framework o librería?

Intro: ¿Framework o librería?

Criterios para analizar:

Pro-Librería	Pro-Framwork
No hay inversión de control	Hay inversión de control
Libre de opinión	Fuertemente opinado
Colección de rutinas descentralizadas	Flujo de datos centralizado
Cubre con una sola tarea en el desarrollo	Cubre la gran mayoría de las tareas de desarrollo

React

Intro: React Puro

Intro: React Puro

Código paso a paso:

Intro: React Puro

Código paso a paso:

La propuesta es crear la página más sencilla que se puede hacer con React. La idea es mostrar que React es una librería cómo cualquier otra.

Intro: React Puro

Código paso a paso:

La propuesta es crear la página más sencilla que se puede hacer con React. La idea es mostrar que React es una librería cómo cualquier otra.

Los bundlers (como CRA, Next.js o Vite) son simplemente un entorno de desarrollo, pero la librería funciona como funcionaría cualquier otra librería.

Intro: React Puro

index.html

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <meta charset="UTF-8">
5.   <link rel="stylesheet" href="./style.css">
6.   <title>React Puro</title>
7. </head>
8. <body>
9.   <div id="root"></div>
10.  <script src="https://unpkg.com/react@17.0.2/umd/react.development.js"></script>
11.  <script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.development.js"></script>
12.  <script src="./index.js"></script>
13. </body>
14. </html>
```


Intro: React Puro

Explicación: Links (index.html)

```
10. <script src="https://unpkg.com/react@17.0.2/umd/react.development.js"></script>  
11. <script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.development.js"></script>
```

Links a las librerías v17.0.2 (react y react-dom)

Intro: React Puro

Explicación: Links (index.html)

```
10. <script src="https://unpkg.com/react@17.0.2/umd/react.development.js"></script>  
11. <script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.development.js"></script>
```

Links a las librerías v17.0.2 (react y react-dom)

```
12. <script src="./index.js"></script>
```

Conexión con el archivo index.js

Intro: React Puro

index.js

```
1.  
2. const App = () => {  
3.  
4.     return React.createElement(  
5.         "div",  
6.         {},  
7.  
8.         React.createElement("h1", {}, "Hola Mundo")  
9.     )  
10. }  
11.  
12. const rootDiv = document.getElementById("root")  
13.  
14. ReactDOM.render(React.createElement(App), rootDiv)
```

Intro: React Puro

Explicación: Componente App (index.js)

```
1. // Componente Principal
2. const App = () => {
3.     // Componente contenedor del título
4.     return React.createElement(
5.         "div",
6.         {},
7.         // Componente del título
8.         React.createElement("h1", {}, "Hola Mundo")
9.     )
10. }
11. // Vanilla JS
12. const rootDiv = document.getElementById("root")
13. // Seteo del componente principal en nuestra página
14. ReactDOM.render(React.createElement(App), rootDiv)
```

Intro: React Puro

Explicación: método createElement()

```
1. React.createElement(  
2.     // String con el nombre de la etiqueta html  
3.     // ej: "div"  
4.     // → <div></div>  
5.     // ej: "custom-web-component"  
6.     // → <custom-web-component></custom-web-component>  
7.     "h1",  
8.     // Objeto con los atributos de la etiqueta  
9.     // ej: React.createElement("div", { id: "titulo" }, "")  
10.    // → <div id="titulo"></div>  
11.    {},  
12.    // Hijo del componente creado  
13.    // ej: React.createElement("div", {}, "foo") → <div>foo</div>  
14.    "Hola Mundo"  
15. )
```

Intro: React Puro

Alumno.js

```
1. // Componente Alumno
2. const Alumno = () => {
3.     // El tercer parámetro puede contener muchos componentes hijos dentro de un array [] o a través de un spread ...
4.     return React.createElement("div", {}, [
5.         React.createElement("h2", {}, `Nombre: `),
6.         React.createElement("h3", {}, `Lenguaje: ❤️`),
7.         React.createElement("h3", {}, `Intereses: `),
8.     ])
9. }
10.
11. export default Alumno
```

Intro: React Puro

Actualización: index.js

```
1. import Alumno from "../Alumno"
2.
3. const App = () => {
4.   return React.createElement(
5.     "div",
6.     {},
7.     React.createElement("h1", {}, "Hola Mundo"),
8.     React.createElement(Alumno),
9.     React.createElement(Alumno),
10.    React.createElement(Alumno)
11.  )
12. }
13.
14. const rootDiv = document.getElementById("root")
15. ReactDOM.render(React.createElement(App), rootDiv)
```

React

Intro: Props

Intro: Props

- *El conocimiento de JavaScript es fundamental en React.*

Intro: Props

- *El conocimiento de JavaScript es fundamental en React.*
- *La utilización de los argumentos en las funciones nos permiten una comunicación directa entre componentes.*

Intro: Props

- *El conocimiento de JavaScript es fundamental en React.*
- *La utilización de los argumentos en las funciones nos permiten una comunicación directa entre componentes.*
- *Comunicación: Padre → Hijo*

Intro: Props

Ejemplo

```
1. const Ejemplo = props => React.createElement("h1", {}, props.titulo)
2.
3. const App = () => {
4.   return React.createElement(Ejemplo, { titulo: "Hola Mundo" })
5. }
```

Intro: Props

Ejemplo

```
1. const Ejemplo = props => React.createElement("h1", {}, props.titulo)
2.
3. const App = () => {
4.   return React.createElement(Ejemplo, { titulo: "Hola Mundo" })
5. }
```

La línea 1 es la definición de la función, la invocación de la misma está en la línea 3 por medio de un callback y las props son “pasadas” a través del segundo parámetro del método `React.createElement()`

Intro: Props

Actualización: Alumno.js

```
1. const Alumno = props => {  
2.  
3.   const interesesFormateados = props.intereses.join(", ")  
4.  
5.   return React.createElement("div", {}, [  
6.  
7.     React.createElement("h2", {}, `Nombre: ${props.nombre}`),  
8.     React.createElement("h3", {}, `Lenguaje: ${props.lenguaje}❤️`),  
9.     React.createElement("h3", {}, `Intereses: ${interesesFormateados}.`),  
10.   ])  
11. }  
12.  
13. export default Alumno  
14.
```

Intro: Props

Actualización: Alumno.js

```
1. const Alumno = props => {
2.     // Implementación de métodos de JS
3.     const interesesFormateados = props.intereses.join(", ")
4.
5.     return React.createElement("div", {}, [
6.         // Props dentro de template strings
7.         React.createElement("h2", {}, `Nombre: ${props.nombre}`),
8.         React.createElement("h3", {}, `Lenguaje: ${props.lenguaje}❤️`),
9.         React.createElement("h3", {}, `Intereses: ${interesesFormateados}.`),
10.    ])
11. }
12.
13. export default Alumno
14.
```

Intro: Props

```
1.  const App = () => { // Nuevo index.js
2.      return React.createElement(
3.          "div",
4.          {},
5.          React.createElement("h1", {}, "Hola Mundo"),
6.          React.createElement(Alumno, {
7.              nombre: "Martín",
8.              lenguaje: "JS",
9.              intereses: ["Helado", "Fútbol", "Danza"],
10.         }),
11.         React.createElement(Alumno, {
12.             nombre: "Alejandra",
13.             lenguaje: "JS",
14.             intereses: ["Helado", "Fútbol", "Viajes", "Música"],
15.         }),
16.         React.createElement(Alumno, {
17.             nombre: "Lucrecia",
18.             lenguaje: "JS",
19.             intereses: ["Helado", "Danza", "Viajes"],
20.         })
21.     )
22. }
```


¡Vamos al código!

Clase 1: Intro

¡Muchas gracias!



ICARO Asociación Civil
CUIT 30716564815
info@icaro.org.ar
www.icaro.org.ar