



Nuestro compromiso es con el *futuro*.

React

Clase 3

Utilidades: Taller 1

*Antes de comenzar con el taller, aprenderemos
dos nuevas utilidades.*

Utilidades: Taller 1

Lo nuevo:

- **Prop “children”**
- **React.Fragment**

React

Utilidades: Prop “children”

Utilidades: Prop “children”

- *La prop “children” nos permitirá aceptar componentes dentro de las etiquetas.*

Utilidades: Prop “children”

- *La prop “children” nos permitirá aceptar componentes dentro de las etiquetas.*
- *La implementación es similar a las etiquetas HTML.*

Utilidades: Prop “children”

- *La prop “children” nos permitirá aceptar componentes dentro de las etiquetas.*
- *La implementación es similar a las etiquetas HTML.*

<Padre> Hijo </Padre>

Utilidades: Prop “children”

Padre.jsx

```
1.  import React from "react"
2.
3.  const Padre = ({ children }) => {
4.
5.      return (
6.          <div>
7.              {children}
8.          </div>
9.      )
10. }
11.
12. export default Padre
```

Utilidades: Prop “children”

Padre.jsx

```
1.  import React from "react"
2.
3.  const Padre = ({ children }) => { // Nueva prop para aceptar componentes hijos.
4.
5.      return (
6.          <div>
7.              {children}
8.          </div>
9.      )
10. }
11. export default Padre
```

Utilidades: Prop “children”

Padre.jsx

```
1. import React from "react"
2.
3. const Padre = ({ children }) => {
4.
5.   return (
6.     <div>
7.       {children} // Acá estamos colocando el componente hijo dentro del componente padre.
8.     </div>
9.   )
10. }
11.
12. export default Padre
```

React

Utilidades: `React.Fragment`

Utilidades: React.Fragment

- *En JavaScript no se puede retornar más de un elemento en una función.*

Utilidades: React.Fragment

- *En JavaScript no se puede retornar más de un elemento en una función.*
- *JSX sigue siendo JS y React nos da una utilidad para solventar ese problema.*

Utilidades: React.Fragment

- *En JavaScript no se puede retornar más de un elemento en una función.*
- *JSX sigue siendo JS y React nos da una utilidad para solventar ese problema.*

```
<React.Fragment> Hijo </ React.Fragment >  
  ◇ Hijo </>
```

Utilidades: React.Fragment

TituloYParrafo.jsx

```
1.  import React from "react"
2.
3.  const TituloYParrafo = () => {
4.      return (
5.
6.          <>
7.              <h1>Lorem, ipsum.</h1>
8.
9.              <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
10.          </>
11.      )
12.  }
13.  export default TituloYParrafo
```


Utilidades: React.Fragment

TituloYParrafo.jsx

```
1. import React from "react"
2.
3. const TituloYParrafo = () => {
4.   return (
5.     // Puede usarse también con React.Fragment, pero su forma más común es <> </>.
6.     <>
7.       <h1>Lorem, ipsum.</h1>
8.       <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
9.     </>
10.   )
11. }
12.
13. export default TituloYParrafo
```

Utilidades: Ejemplo

App.jsx

```
1.  import React from "react"
2.  import Padre from "../Padre"
3.  import TituloYParrafo from "../TituloYParrafo"
4.
5.  export const App = () => {
6.    return (
7.      <
8.        <Padre>
9.          <TituloYParrafo />
10.        </Padre>
11.
12.        <button>Continuar</button>
13.      </>
14.    )
15.  }
```

Utilidades: Ejemplo

App.jsx

```
1. import React from "react"
2. import Padre from "../Padre"
3. import TituloYParrafo from "../TituloYParrafo"
4.
5. export const App = () => {
6.   return (
7.     ◇ // Ejemplo de React.Fragment.
8.     <Padre>
9.       <TituloYParrafo />
10.    </Padre>
11.
12.    <button>Continuar</button>
13.  </>
14. )
15. }
```

Utilidades: Ejemplo

App.jsx

```
1.  import React from "react"
2.  import Padre from "../Padre"
3.  import TituloYParrafo from "../TituloYParrafo"
4.
5.  export const App = () => {
6.    return (
7.      <
8.        <Padre>
9.          <TituloYParrafo /> // Ejemplo de children.
10.        </Padre>
11.
12.        <button>Continuar</button>
13.      </>
14.    )
15.  }
```

Utilidades: Taller 1

*Ahora sí,
¡continuemos con el taller!*

React

Temas: Taller 1

Temas: Taller 1

¿Qué vamos a hacer?:

- **Componente Card**
- **Componente Producto**
- **Componente User/Password**

Temas: Taller 1

¿Qué vamos a necesitar?:

- **Proyecto “levantado” con Vite**
- **Conocimiento básico de React**

Temas: Taller 1

¿Qué vamos a necesitar?:

- **Proyecto “levantado” con Vite**
- **Conocimiento básico de React**

Extras:

- **HTML: <button />, <input />, <label />, **
- **CSS: flexbox o grid**
- **JS: condicionales ternarios, callbacks**

React

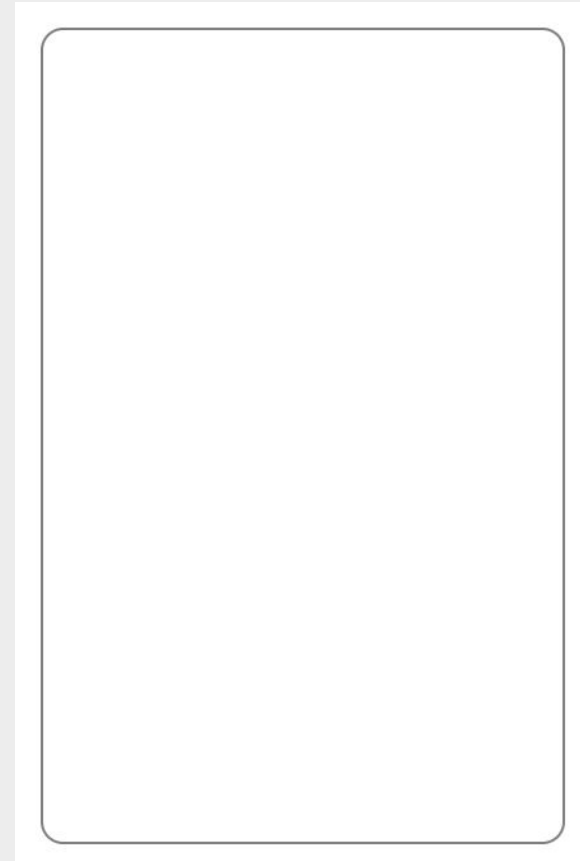
Taller 1: Componente Card

Taller 1: Componente Card

React se basa en la creación de componentes, la propuesta es crear un componente Card básico el cual va a poder ser llenado con un pequeño formulario y luego con la información de un producto.

Taller 1: Componente Card

El Componente Card vacío deberá verse así:



Taller 1: Componente Card

Requisitos:

- Archivo de estilos.
- Atributo `className` para manejar los estilos.
- Atributo que cambie la orientación (`horizontal/vertical`).
- Atributo que cambie el tamaño (`grande/pequeño`).
- Prop `“children”` que acepte otro componente React.

React

Taller 1: Componente Product

Taller 1: Componente Producto

Este componente también va a utilizar el componente Card como componente padre. Deberá mostrar precio e imagen de un producto.

** El botón “comprar” no requiere funcionalidad por ahora.*

Taller 1: Componente Producto

El Componente Producto deberá verse así:



Taller 1: Componente Producto

Requisitos:

- Prop con src para la imagen del producto.
- Se deberá usar el componente Card en posición Vertical.
- Botón comprar.

** IMPORTANTE: No darle tanta importancia detalles visuales.*

*** Para la imagen puede usarle esta url “<https://picsum.photos/300/300>”.*

React

Taller 1: User/Password


Taller 1: User/Password

Este componente deberá utilizar el componente Card anteriormente creado y deberá ser insertado por medio de la prop “children” previamente creada.

** No requiere funcionalidad de formulario por ahora.*

Taller 1: User/Password

El Componente User/Password deberá verse así:



A wireframe diagram of a User/Password form. The form is contained within a light gray rounded rectangle. Inside this rectangle is a pink rounded rectangle representing the form area. At the top of the pink area is the label 'Usuario' in black text. Below it is a white rectangular input field with a thin gray border. Underneath the input field is the label 'Contraseña' in black text. Below that is another white rectangular input field with a thin gray border. At the bottom of the pink area is a blue rounded button with the white text 'Aceptar'.

Taller 1: User/Password

Requisitos:

- Usar dos componentes Card con distinto tamaño.
- Input de usuario.
- Input de contraseña.
- Botón continuar.

** IMPORTANTE: No darle tanta importancia detalles visuales.*

React

Taller 1: Extra

Taller 1: Extra

Enviar el botón de los componentes User/Password y Producto al componente Card.

Hacer un botón genérico en el componente Card que acepte un callback.

¡Éxitos!

Clase 3: Taller 1

¡Muchas gracias!



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