



# Parte 2: Dándole estilos a las aplicaciones React

UD5: Construcción de aplicaciones web con React

# Después de este documento...

- Conectarás lo aprendido hasta ahora de React con aspectos de diseño del módulo de DIW:
  - Habrás aplicados estilos importando directamente el **archivo css**
  - Verás que también se puede usar **Bootstrap** realizando pequeñas modificaciones
  - Incorporarás estilos css en el código JavaScript usando ***styled components***
  - Podrás aplicar estilos Bootstrap en las aplicaciones más directamente con **React Bootstrap**
  - Usarás componentes de terceros con **mui**

# Forma “tradicional”

- Aplicar una hoja de estilos en un componente:
  - `import './index.css'`
  - Usar atributos **className** y **style**
- Usar Bootstrap usando los enlaces CDN en el **index.html**
  - Si copiamos un componente de Bootstrap, hay que adaptar el HTML a JSX
    - Cambiar class por className
    - Las etiquetas tienen que cerrar (input, hr...)
    - ...

# *Styled components*

- Instalar **dependencias**

```
npm install --save @emotion/styled
```

- Importar "***styled***"

```
import styled from '@emotion/styled'
```

- Declarar el ***styled component***, que luego se usa como un componente "normal"

```
const Texto = styled.div`  
  background-color: #B7322C;  
`
```

# Ejemplo

- Declarar

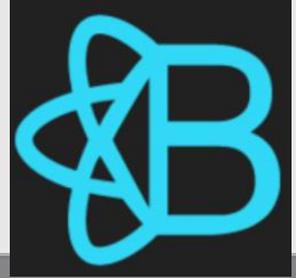
```
const Texto = styled.div`  
  background-color: #B7322C;  
  color: #FFF;  
  padding: 15px;  
  font-size: 22px;  
  text-transform: uppercase;  
  text-align: center;  
`
```

- Usar

```
const Error = ({children}) =>  
{  
  return (  
    <Texto>  
      {children}  
    </Texto>  
  )  
}
```

**TODOS LOS CAMPOS SON  
OBLIGATORIOS**

# React Bootstrap



- Es la versión de Bootstrap reconstruida para **React** ([documentación](#))
  - Ya es código JSX
  - Para usarlo:
    - Se instalan dependencias:

```
npm install --save react-bootstrap bootstrap
```

- Se importa el css en main o app:

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

- Es necesario importar cada componente que se use:

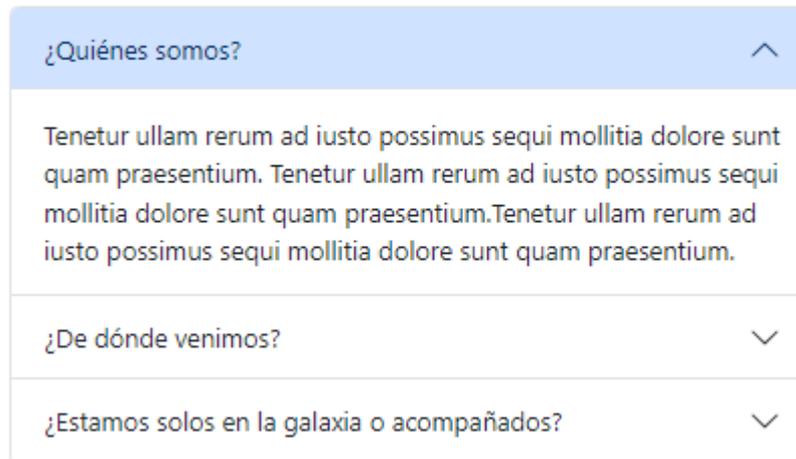
```
import Button from 'react-bootstrap/Button';
```

# EJERCICIO PROPUESTO I

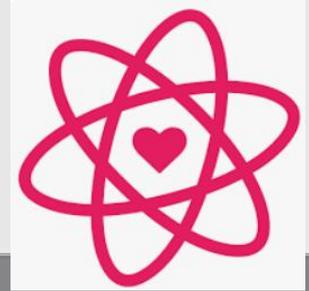
## (1/2)

- Crea una nueva versión del componente acordeón, esta vez usando **Accordion** de React Bootstrap:

### FAQ (Preguntas más frecuentes)



# React icons



- Un módulo muy usado es **React-icons** (<https://react-icons.github.io/react-icons/>):
- Se instala con: `npm install react-icons --save`
- Para usarlo importamos el icono concreto de la ruta de iconos:

```
import { VscBug, VscGlobe } from "react-icons/vsc";  
...  
root.render(  
  <>  
    <VscBug/>  
    <VscGlobe/>  
  </>  
);
```



# EJERCICIO PROPUESTO I

## (2/2)

- Modifícalo para que incorpore iconos

### FAQ (Preguntas más frecuentes)

¿Quiénes somos?	▼
¿De dónde venimos?	▼
¿Estamos solos en la galaxia o acompañados?	▼

# MUI



- Implementación React de **Material Design**
  - Surge Material UI en 2014
  - Desde la **versión 5** (2019) recibe el nombre de MUI – todos los paquetes cambian de @material-ui a @mui
- Componentes gratuitos de código abierto muy utilizados para acelerar el desarrollo de aplicaciones web ([documentación](#), [mui-treasury](#))
  - Menús, calendarios, etc.

# Instalación y uso

- Instalar dependencias de MUI

```
npm install @mui/material @emotion/react @emotion/styled
```

- Si vas a usar iconos

```
npm install @mui/icons-material
```

- La fuente que usa por defecto es **Roboto** y también es necesario instalarla o importarla

```
npm install @fontsource/roboto
```

```
{} package.json > {} dependencies > @emotion/react
11 | "dependencies": {
12 |   "@emotion/react": "^11.10.5",
13 |   "@emotion/styled": "^11.10.5",
14 |   "@mui/icons-material": "^5.11.0",
15 |   "@mui/material": "^5.11.8",
```

# Ejemplos

- Para usar un **botón**:

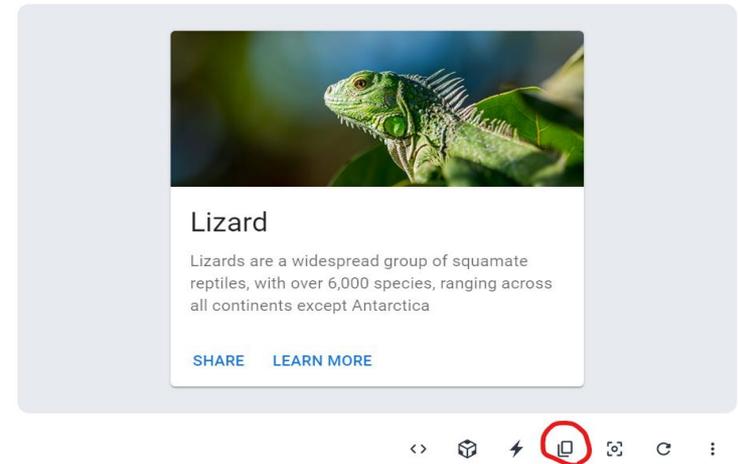
```
import { Button } from "@mui/material";  
<Button variant="contained">Hello World</Button>
```

- Se le puede añadir un icono:

```
startIcon={<AirplanemodeActiveIcon />}
```

- Para usar una **Media Card**

- Copiar el código
- Ajustar contenido (imágenes, texto...)



# Elementos básicos de mui

- **Propiedad sx:** atajo para definir estilos personalizados que tienen acceso al tema
- **System props:** componente compatible con todos
  - color, mb, pb, align...
- **Componentes muy usados:**
  - **Container** (maxWidth...)
  - **Typography** (variant: h1, h2, h2..)
  - **Grid** (basado en 12 columnas como Bootstrap)
  - **Box...**

```
<Container
  maxWidth="xs"
  sx={{
    border: 2,
    boxShadow: 1,
    pb: 2, }} >
</Container>
<Typography
  variant="h1"
  mb={2}
  color="primary"
  align="center" >
Soy un h1
</Typography>
```

# Usando temas con MUI

- Crearemos el tema con `createTheme`, y lo ponemos disponible en nuestra app con `<ThemeProvider>`

```
import { createTheme, ThemeProvider } from "@mui/material/styles";

const theme = createTheme({
  palette: {
    primary: { main: "#000e35", ... },
    secondary: { main: "#f50057", ... },
  },
});
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <App />
    </ThemeProvider>
  </React.StrictMode>);
```