

# Guía para clonar la práctica interactive-cv-mui

---

## 1. Ubicación para clonar el proyecto

Vamos a clonar el repositorio dentro de la siguiente ruta local en el ordenador:

```
desktop/factoria-f5/pildoras/react
```

---

## 2. Clonar y preparar el proyecto

Abrimos la terminal y ejecutamos los siguientes comandos para clonar el repositorio, instalar las dependencias y arrancar la aplicación:

```
cd ~/desktop/factoria-f5/pildoras/react
git clone https://github.com/Aday25/interactive-cv-mui.git
cd interactive-cv-mui
code .
```

Y desde Visual Studio Code:

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

---

### 3. Estructura del proyecto

Aquí tenemos el esquema de las partes de la práctica:

```
interactive-cv-mui/
|
|— public/
|   └─ vite.svg                # Logo y archivos estáticos para la
app
|
|— src/
|   └─ assets/
|       └─ certificates/      # Imágenes de certificados
|           └─ avatar.png      # Imagen principal del avatar que
aparece en la página Home
|       └─ components/        # Componentes reutilizables como
Navbar
|           └─ Navbar.jsx
|       └─ pages/              # Páginas principales del CV
interactivo
|           └─ About.jsx
|           └─ Certificates.jsx
|           └─ Home.jsx        # Página principal con el avatar y
presentación
|           └─ Skills.jsx
|       └─ theme/              # Contexto y configuración de temas
para Material UI
|           └─ ThemeContext.jsx
|           └─ theme.js
|       └─ App.jsx             # Componente raíz que conecta las
páginas y la navegación
|           └─ App.css          # Estilos globales para la app
|           └─ index.css        # Estilos base
|           └─ main.jsx         # Entrada para arrancar la app React
con Vite
|
└─ package.json                # Dependencias y scripts de ejecución
```

---

### 4. Dependencias principales

El proyecto utiliza las siguientes tecnologías y librerías:

- **React**
  - **Material UI** (`@mui/material`, `@mui/icons-material`)
  - **Framer Motion** (para animaciones)
  - **React Icons** (para iconos adicionales)
- 

## 5. Uso de la aplicación

Ejecutamos `npm run dev` para iniciar la aplicación en el navegador.

---

## 6. Cómo modificar el contenido

Para personalizar la página y adaptarla a nuestro perfil, debemos editar diferentes partes del proyecto:

- **Avatar y texto principal:**  
El avatar se encuentra en `src/assets/avatar.png`. Podemos reemplazar esta imagen por la nuestra manteniendo el mismo nombre y formato para que se actualice automáticamente.  
Para cambiar el texto que aparece junto al avatar (nombre, presentación, saludo), debemos editar el componente `Home.jsx`, donde están definidos esos textos dentro de los elementos `<Typography>`.
- **Certificados:**  
Para añadir nuevos certificados, colocamos las imágenes dentro de la carpeta `src/assets/certificates/`. Luego, abrimos el archivo `Certificates.jsx` y actualizamos el array `certificates` agregando un nuevo objeto con los campos: `title`, `image` (nombre de la imagen), y `details` (información del certificado). Esto hará que aparezcan en la lista con animación y funcionalidad para abrir la imagen en grande.

- **Habilidades:**

Las habilidades se gestionan en el archivo `Skills.jsx`, dentro del array `skills`. Para modificar las existentes o añadir nuevas, simplemente editamos o añadimos objetos con las propiedades: `name` (nombre de la habilidad), `icon` (icono representativo), y `level` (nivel en porcentaje para la barra de progreso).

- **Experiencias profesionales:**

El componente `About.jsx` contiene el array `experiences`, donde podemos añadir o modificar las experiencias, cambiando el icono, título y descripción para que reflejen nuestro recorrido profesional.

De este modo, tenemos control total sobre el contenido, imágenes y textos para que el CV interactivo refleje fielmente nuestra identidad y habilidades.

---

## 7. Extra: Personalización del tema visual

En la carpeta `src/theme/` tenemos dos archivos clave:

- `theme.js`: aquí se definen los colores, fuentes y estilos generales que Material UI aplica en toda la app.
- `ThemeContext.jsx`: un contexto React que permite cambiar entre modo claro y modo oscuro globalmente, con un toggle fácil.

Para darle un toque más personal o adaptar la apariencia a vuestro gusto, aquí es donde, cambiando estos archivos, se modifica el look completo del CV interactivo manteniendo la uniformidad visual.