

Scape Room React

1 Crear proyecto React

- **Paso:** Abre una terminal en la carpeta donde quieras crear el proyecto y ejecuta el siguiente comando para iniciarlo:

Shell

```
pnpm create vite .
```

- **Detalles adicionales:**
 - Selecciona **React** y luego **JavaScript** cuando te pregunte.
- **Objetivo:** Crear un proyecto base de React con Vite.
- **Pista:** Revisa que el proyecto tenga una estructura con una carpeta **src** y un archivo **App.jsx**. No te olvides los comandos:

Shell

```
pnpm install  
pnpm run dev
```

2 Instalar Axios con pnpm (o se puede usar fetch)

- **Paso:** Instala Axios para hacer solicitudes HTTP a la API usando el siguiente comando:

Shell

```
pnpm install axios
```

- **Objetivo:** Tener Axios instalado para realizar las peticiones.
- **Pista:** Confirma que Axios está en las **dependencias** en **package.json**

3 Limpiar los archivos iniciales

- **Paso:** Ve a la carpeta `src` y limpia los archivos, eliminando los que no necesites (como `App.css` o `logo.svg`).
- **Objetivo:** Simplificar el proyecto para que tengamos solo `App.jsx`, `main.jsx`, e `index.css`
- **Pista:** Limpia `App.jsx` dejando solo una estructura mínima:

```
JavaScript
const App = () => {
  return (
    <>
      { /* Aquí irá el contenido */ }
    </>
  );
};

export default App;
```

4 Crear carpeta `pages`

- **Paso:** Dentro de `src`, crea una carpeta llamada `pages` donde pondremos nuestros componentes de página.
- **Objetivo:** Mantener la organización del proyecto para hacerlo más escalable.
- **Pista:** La carpeta `pages` debe estar dentro de `src`

5 Crear componente `ShowCharacter` en `pages`

- **Paso:** En la carpeta `pages`, crea un archivo llamado `ShowCharacter.jsx`
- **Objetivo:** Este archivo será el componente encargado de mostrar los personajes.
- **Pista:** Crea el componente `ShowCharacter` como una función de flecha:

JavaScript

```
const ShowCharacter = () => {  
  return (  
    <>  
      { /* Aquí irá el contenido */ }  
    <>  
  );  
};  
  
export default ShowCharacter;
```

6. Mostrar el componente **ShowCharacter** en **App.jsx**

- **Paso:** Importa y muestra el componente **ShowCharacter** dentro de **App.jsx**. En el componente **ShowCharacter** escribe en un **<h1>** la frase “Hola, soy el componente ShowCharacter”.
- **Objetivo:** Verificar que **ShowCharacter** aparece en la página principal.
- **Pista:** Modifica **App.jsx** para que quede así:

JavaScript

```
import ShowCharacter from './pages/ShowCharacter';  
  
const App = () => {  
  return (  
    <>  
      <ShowCharacter />  
    <>  
  );  
};  
  
export default App;
```

- **Prueba:** Inicia el servidor de desarrollo con:

Shell

```
pnpm run dev
```

- Si ves el texto **"Hola soy el componente ShowCharacters"**, ¡lograste este paso!

7. Encontrar la API de personajes de *Rick y Morty*

- **Paso:** Busca en internet la API de *Rick y Morty* que muestre los personajes.
- **Objetivo:** Asegurar que tienes la URL correcta.
- **Pista:** Busca la palabra *character* en la URL. Explora la documentación de la API de Rick y Morty para ver qué datos puedes obtener.

8. Crear la función para obtener datos de la API y mostrar en consola

- **Paso:** Usa `axios`, `useEffect` y `useState` para hacer la solicitud a la API cuando se monte el componente `ShowCharacter` y mostrar los datos en consola.
- **Objetivo:** Llamar a la API y mostrar el resultado en consola.
- **Pista:**
 - Crea una variable de estado y la función que la modifica usando `useState`. La variable puede llamarse por ejemplo `characters` y la función que modifica su estado puede llamarse `setCharacter`. No te olvides poner el estado inicial de la variable, que debe ser un array vacío.
 - Crea una función que obtenga los datos, por ejemplo que se llame `getAllCharacters`.
 - Implementa la función `getAllCharacters` haciendo una llamada de tipo `get` a la API (usando `axios` o `fetch`) y que muestre los datos en consola.
 - Crea un `useEffect` y mueve la función `getAllCharacters` para dentro del `useEffect` (acuerdate, con dependencia vacía para que se ejecute solo una vez, cuando se monta el componente o cuando se renderiza).
- **Prueba:** Si ves en la consola una lista de 20 nombres de personajes de *Rick y Morty*, ¡completaste este paso!

9. Mostrar solo los campos: imagen, nombre, especie y estado en la página

- **Paso:** Modifica el componente `ShowCharacter` para mostrar solo estos cuatro datos: **imagen**, **nombre**, **especie** y **estado**, haciendo uso de la función ***map*** para mostrar cada personaje en la página.
- **Objetivo:** Mostrar los detalles clave de cada personaje en la página.
- **Pista:** Puedes estructurarlo así:

JavaScript

```
<div key={character.id}>
  <img src={character.image} alt={character.name} />
  <p>Nombre: {character.name}</p>
  <p>Especie: {character.species}</p>
  <p>Estado: {character.status}</p>
</div>
```

- **Prueba:** Ahora, deberías ver una lista de personajes de *Rick y Morty*, donde cada personaje muestra solo la **imagen**, **nombre**, **especie** y **estado** en la página.

10. ¡Escapaste! 🚀

¡Felicitaciones! Has completado con éxito este proyecto en React usando Vite, `pnpm`, Axios y la API de *Rick y Morty*.