



Editar en Github

Empezar proyecto

## </> Blog de Starwars

Solución

### Lista de Lectura del Blog de Starwars

*La fuerza es fuerte en este ejercicio...*

Vamos a construir una versión minimalista del [Banco de datos de Star Wars](#) con una funcionalidad de "Read later" o "favoritos".

¡Aquí hay un Demo!

Character	Description	Action Buttons
Luke Skywalker	Gender: male Hair Color: blond Eye-Color: blue	[Learn more!] [Heart]
C-3PO	Gender: n/a Hair Color: n/a Eye-Color: yellow	[Learn more!] [Heart]
R2-D2	Gender: n/a Hair Color: n/a Eye-Color: yellow	[Learn more!] [Heart]

### 👉 Cómo comenzar este proyecto

No clones este repositorio porque vamos a usar una plantilla diferente.

Recomendamos abrir el `react flux boilerplate` usando un entorno de desarrollo como [Codespaces](#) (recomendado) o [Gitpod](#). Alternativamente, puedes clonarlo en tu computadora local usando el comando `git clone`.

Este es el repositorio que necesitas abrir o clonar:

1 <https://github.com/4GeeksAcademy/react-hello-webapp>

👉 Por favor sigue estos pasos sobre [cómo comenzar un proyecto de programación](#).

💡 Importante: Recuerda guardar y subir tu código a GitHub creando un nuevo repositorio, actualizando el remoto (`git remote set-url origin <your new url>`) y subiendo el código a tu nuevo repositorio usando los comandos `add`, `commit` y `push` desde la terminal de git.



## Instrucciones

1. Usa componentes de Bootstrap, **casi** no necesitas CSS personalizado.
2. Tómate un tiempo para comprender la API [SWAPI.tech](#), esta será nuestra fuente de información, estaremos consumiendo esta API.
3. Haremos *fetch* de personas, vehículos y planetas de la SWAPI para mostrarlos en tu aplicación.
4. Declara un array de `favoritos` en tu *central store* y permite que el usuario agregue o elimine favoritos.

### Construyendo la vista principal

- Crear una aplicación web React que enumera entidades de *personas*, *vehículos* y *planetas* proporcionados por la [SWAPI](#).

Nota: por favor utiliza [swapi.tech](#) y no [swapi.dev](#) porque la segunda está dando problemas últimamente.

The screenshot shows a dark-themed user interface for browsing a database. On the left, a sidebar lists categories: BROWSE, ALL, CHARACTERS (which is selected and highlighted in orange), CREATURES, DROIDS, LOCATIONS, ORGANIZATIONS, SPECIES, VEHICLES, WEAPONS+TECH, and MORE. The main area displays a grid of cards, each representing a character from Star Wars. Each card features a small thumbnail image, the character's name, and the word 'DATABANK' below it. The characters shown in the visible grid are 4-LOM, 4D-M1N, 8D8, AAYLA SECURA, ADI GALLIA, ADMIRAL ACKBAR, ADMIRAL COBURN, ADMIRAL GARRICK, ADMIRAL KASSIUS, ADMIRAL KILIAN, ADMIRAL MOTT, and ADMIRAL OZZEL. At the bottom of the grid, there is a 'SHOW MORE' button.

### Construyendo la vista detallada

- Cada entidad debe tener una breve descripción (Tarjeta Bootstrap) y una vista de detalles (Componentes Bootstrap):

The screenshot shows a detailed view of Luke Skywalker's profile. On the left is a large, close-up portrait of Luke wearing his iconic brown hooded robe. To the right of the portrait is a section titled "LUKE SKYWALKER" containing a biography. Below the portrait and bio is a table with various details about Luke Skywalker, including his appearances, affiliations, locations, gender, dimensions, species, vehicles, weapons, and tools. The table rows are as follows:

APPEARANCES	AFFILIATIONS	LOCATIONS	GENDER	DIMENSIONS	SPECIES	VEHICLES	WEAPONS	TOOL
Star Wars: Episode III Revenge of the Sith, Star Wars: Episode IV A New Hope, Star Wars: Episode V The Empire Strikes Back, Star Wars: Episode VI Return of the Jedi, Star Wars: Episode VII The Force Awakens, Star Wars: Episode VIII The Last Jedi, Star Wars Rebels	Rebel Alliance, Jedi Order	Polis Massa, Lars Moisture Farm, Tatooine	Male	Height: 1.72m	Human	T-16 Skyhopper, X-34 Landspeeder, X-wing Starfighter, Snowspeeder	Lightsaber, Blaster, Pistol, Luke Skywalker's Lightsaber (Green Blade), Anakin, Luke, and Rey's Lightsaber	Bacta Tank, Moisture Vaporator

**Importante:** La SWAPI no proporciona las imágenes, pero puedes usar <https://starwars-visualguide.com> para obtener las imágenes. El enfoque de este ejercicio es practicar *fetch*, *router* y *context*. También puedes enfocarte en una paleta de colores y diseño simple para que se vea bien.

**Importante 2:** no te preocupes si los datos que obtienes de la SWAPI no coinciden con los datos que ves en starwars.com.

Usa toda la información que proporciona la SWAPI (verifica la documentación y/o las respuestas JSON).

## Funcionalidad "Read later" o "Favoritos"

Implementa una funcionalidad de "Read later", es decir, un botón que permita al usuario "guardar" el elemento (personaje, vehículo o planeta) en una lista especial. La ubicación de esta lista es a elección, mientras se muestre correctamente (en nuestra demo es un botón en la navbar); esta lista se asemeja a la lista principal, pero solo muestra los elementos "guardados".

## Uso de Context

Para asegurarse que el usuario pueda "guardar" el elemento, debes implementar una acción a la que se pueda acceder desde cualquier lugar dentro de la aplicación.

## 😎 ¿Te sientes seguro?

Las siguientes funciones no son necesarias para la solución final, pero puedes desarrollarlas si te sientes lo suficientemente seguro:

- +1 Evita que el sitio web haga Fetch a la API de Starwars nuevamente si se actualiza la página (puedes usar el almacenamiento local (`localStorage`) para guardar la *store* en el navegador local).
- +3 Implementa una barra de búsqueda con "autocompletear" para los personajes, planetas y vehículos. Cuando haces clic en autocompletear, debería llevarte a la página detallada del elemento.

Este y otros proyectos son usados para [aprender a programar](#) por parte de los alumnos de 4Geeks Academy [Coding Bootcamp](#) realizado por [Alejandro Sánchez](#) y muchos otros contribuyentes. Conoce más sobre nuestros [Cursos de Programación](#) para convertirte en [Full Stack Developer](#), o nuestro [Data Science Bootcamp](#).

---

[ENVIAR PROYECTO](#)

[← Anterior](#)      [Siguiente →](#)