

# Ingeniería Web — Proyecto Final



## API — Rick and Morty

### **Equipo 10: CodeBreakers**

Jaider Felipe Santos Garzon

Ander Etxabe Gil

Alberto Bellera Alsina

Lucas Daniel Espitia Corredor

# **ÍNDICE**

1. Introducción.....	3
2. Sobre el equipo.....	4
3. Gestión de la configuración.....	5
4. Análisis y diseño.....	6
5. Wireframes y prototipado.....	7
6. API empleada.....	9
7. Funcionalidad de la página.....	10
8. HTML.....	11
9. CSS.....	12
10. JavaScript.....	13
11. Mobile First / Responsive Design.....	14
12. Accesibilidad.....	15
13. Usabilidad.....	16
14. Implementación mediante REACT.....	19
15. SEO.....	20
16. Rendimiento.....	21
17. Despliegue.....	22

# 1. Introducción

En el marco de la asignatura y como parte del trabajo final obligatorio, se nos planteó el desarrollo de una aplicación web que permitiera aplicar de forma práctica los conocimientos adquiridos durante el curso. A partir de un conjunto de seis opciones, seleccionamos el desarrollo de una página web conectada a la API oficial de *Rick and Morty*, una propuesta que nos pareció interesante tanto a nivel técnico como visual, y que ofrecía múltiples posibilidades para trabajar con datos reales, estructuración por componentes y navegación dinámica. Nuestro grupo, CodeBreakers (Grupo 10), formado por cuatro estudiantes, asumió este proyecto con responsabilidad y el objetivo de entregar una solución funcional, clara y bien documentada.

El resultado es una página web interactiva que conecta con la API de Rick and Morty y permite al usuario explorar información relacionada con la serie. La aplicación cuenta con una página principal que sirve como introducción al sitio, una sección de personajes, una vista organizada de episodios por temporadas, una sección de ubicaciones, y una última página en la que se presentan los miembros del equipo que ha desarrollado el proyecto. Toda la estructura está orientada a la claridad visual y a la buena experiencia del usuario, además de estar ambientada en la temática de Rick y Morty.

Antes de iniciar el desarrollo, realizamos un breve proceso de prototipado utilizando Figma, en el contexto de una sesión impartida en clase sobre diseño de interfaces. A pesar de que los wireframes eran sencillos y orientativos, nos sirvieron para establecer una visión general del proyecto, definir la estructura de navegación y facilitar la toma de decisiones iniciales.

Para la construcción del sitio utilizamos diversas herramientas clave. El proyecto está desarrollado en React, apoyándonos en la biblioteca React-Bootstrap para facilitar el diseño de componentes visuales y su adaptabilidad a distintos tamaños de pantalla. La organización del proyecto sigue una estructura modular, con carpetas claramente diferenciadas para componentes, páginas, estilos, constantes y recursos multimedia. Además, empleamos herramientas como Figma para el diseño previo, GitLab para la colaboración en equipo y Node.js como entorno para gestionar dependencias y ejecutar la aplicación.

## 2. Sobre el equipo

El equipo desarrollador está formado por cuatro estudiantes que conforman el grupo CodeBreakers (Equipo 10). Hemos trabajado de manera colaborativa, combinando nuestras habilidades y conocimientos para sacar adelante el proyecto.

La coordinación se ha llevado a cabo a través de un repositorio común en GitLab, donde inicialmente se intentó una división por ramas, pero finalmente optamos por trabajar directamente sobre la rama principal (*main*), dado que nuestras tareas no se solapaban y esto facilitó un flujo de trabajo más directo y ágil. Se explica de manera más detallada en el siguiente apartado, el de la gestión de la configuración.

### 3. Gestión de la configuración

La gestión de la configuración, al tratarse de un trabajo en grupo, claramente ha sido clave en el avance de este proyecto. Hemos trabajado en la herramienta que se nos ha dado, GitLab, la cual nos ha permitido gestionar de la manera que hemos creído más conveniente todo el intercalado de trabajo entre los distintos componentes del equipo.

Sobre todo, la clave de la correcta coordinación entre todos los miembros de cara a estar todos 'en la misma página' del proyecto, a dividir el trabajo a realizar, a permitirnos el trabajar más de uno a la vez, a no pisarnos entre diferentes avances, y, en general, a tener todo claro respecto a qué podemos y debemos hacer cada uno, ha sido la comunicación.

La comunicación y coordinación nos ha permitido, en última instancia, que todos hayamos acabado más que satisfechos y orgullosos del trabajo y los esfuerzos realizados. Principalmente gracias a ver el resultado final del proyecto, pero, también, gracias a valorar todo lo que ha sido parte del proceso, desde el primer momento en el que nos costó algo más de la cuenta el comenzar a repartirnos y asignarnos las tareas, hasta el momento final en el que el trabajo está completado y finalizado.

Además, creemos que, si bien todos estábamos dispuestos a arrimar el hombro con el objetivo de que el trabajo saliese adelante de la mejor manera posible, no ha habido que hacer esfuerzos excesivamente innecesarios por parte de ninguno de los componentes del equipo, sino que todos hemos contribuido al proyecto en una cantidad, y, sobre todo, calidad ciertamente similar. En definitiva, el reparto de tareas ha sido lo suficientemente equitativo como para valorar positivamente lo complacidos que estamos con nuestra implicación personal en el proyecto.

La positiva coordinación mediante GitLab, además de en la comunicación directa entre todos los integrantes del equipo, se basa en cómo planteamos desde el primer momento el cómo íbamos a gestionar los commits de cada uno de los integrantes (algunos muy gradual y progresivamente, otros más haciendo el trabajo por su cuenta e integrándolo más adelante de una manera más 'bruta'...). Esto, unido al uso de distintas ramas de trabajo y su posterior resolución de los conflictos en algunos casos concretos, nos ha brindado la seguridad necesaria para trabajar sabiendo que en ningún momento se iba a perder el trabajo de ninguno de nosotros, ni íbamos a tener problemas en integrar y subir nuestro trabajo realizado al GitLab.

## 4. Análisis y diseño

Antes de comenzar con la programación del proyecto, mantuvimos varias conversaciones entre los miembros del grupo para alinear ideas y decidir cómo queríamos enfocar la estructura y apariencia de la página. Uno de los puntos clave que surgió fue la utilidad del formato "wiki" para presentar grandes volúmenes de información de forma clara y jerárquica. Esta idea no era nueva para nosotros: en el proyecto anterior, el minijuego basado en *South Park*, ya habíamos extraído datos de una wiki no oficial, lo que nos ayudó a ver el potencial de este tipo de estructura para contenidos similares.

Teniendo en cuenta la naturaleza de la API de Rick y Morty, que proporciona datos bien estructurados sobre personajes, ubicaciones y episodios, la idea de construir una wiki visual e interactiva nos pareció muy acertada. Así, el planteamiento inicial fue construir un sitio dividido en varias secciones clave: una landing page principal que sirviese de presentación del sitio, y páginas específicas para personajes, ubicaciones y episodios, además de un apartado dedicado al equipo desarrollador.

Para organizarnos visualmente, diseñamos unos primeros mockups que nos ayudaron a definir de manera sencilla la disposición general de los elementos y la navegación entre secciones. Estos bocetos fueron creados con la herramienta Figma durante una sesión de clase dedicada al prototipado. Aunque eran diseños básicos y orientativos, cumplieron su función al permitirnos tener una idea compartida y visual sobre la estructura que queríamos implementar.



Los mockups funcionaron como una guía general. A partir de ellos, fuimos ajustando el diseño conforme avanzaba el desarrollo técnico, teniendo en cuenta factores como la adaptabilidad, el contenido real de la API y la usabilidad. Además se decidió cambiar aspectos como la muestra de un personaje, optando por modales en vez de pantallas separadas, por recomendación del docente.

## 5. Wireframes y prototipado

Durante la fase inicial del proyecto, una vez definida la idea general y la estructura que queríamos para la web, pasamos a realizar un prototipado básico con el objetivo de visualizar cómo distribuiríamos los contenidos y qué aspecto general tendría cada una de las secciones.

Para ello, el equipo se reunió en una videollamada, donde comentamos ideas, referencias y prioridades en cuanto a diseño. Acordamos los elementos más importantes que debían aparecer en la página —como el menú de navegación, las tarjetas de personajes o la disposición por temporadas de los episodios— y debatimos sobre la experiencia de usuario que queríamos ofrecer.

Tras esta puesta en común, pasamos a diseñar un primer mockup utilizando Figma, una herramienta con la que ya habíamos trabajado previamente en clase. Dado que estábamos aún en una fase temprana del desarrollo, no dedicamos un esfuerzo excesivo al detalle visual: los wireframes que creamos fueron esquemáticos y funcionales, pensados como una guía orientativa para marcar la dirección del diseño. No se trataba de un diseño cerrado, sino de una propuesta inicial que nos ayudara a compartir una visión común y facilitara la implementación posterior.



Imagen 1. Mock-up de la landing page

Como se puede observar en la imagen 1, el prototipado y los diseños fueron muy básicos, ya que solamente los queríamos utilizar a modo de guía para entender nuestra visión y tener una puesta en común sobre la navegación de la web.



## 6. API empleada

Para el desarrollo del proyecto, uno de los primeros pasos fue elegir la fuente de datos con la que íbamos a trabajar. De entre las seis APIs propuestas por el profesorado, decidimos hacer una votación entre los miembros del grupo para elegir la que nos resultara más interesante y accesible. El resultado fue unánime: nos decantamos por la API pública de Rick and Morty.

Esta API gratuita ofrece acceso a información básica pero estructurada sobre el universo de la serie Rick and Morty. Al comenzar a explorarla, nos dimos cuenta de que su contenido estaba dividido principalmente en tres bloques de información: personajes, ubicaciones y episodios. Cada uno de estos bloques contiene datos bastante completos dentro de su categoría, y lo más interesante es que están enlazados entre sí. Por ejemplo, cada personaje tiene asociada una ubicación de origen, y los episodios a su vez contienen información sobre los personajes que aparecen en ellos.

En un primer momento, esta simplicidad nos pareció una limitación. Al tratarse de una API sin datos demasiado profundos o múltiples endpoints complejos, creímos que el margen de desarrollo se podía quedar corto. Sin embargo, a medida que fuimos avanzando en el diseño y planificación de la web, entendimos que precisamente esa estructura sencilla y bien enlazada era una ventaja. Nos permitía trabajar con la información justa y necesaria para implementar todas las funcionalidades principales sin caer en la sobrecarga de datos irrelevantes.

Gracias a esta API pudimos estructurar el sitio en torno a las tres grandes secciones que ofrece: Personajes, Ubicaciones y Episodios, combinándolas de manera visual e interactiva con React. Además, su documentación clara y su arquitectura REST nos facilitaron enormemente la integración con nuestro frontend.

### REST

Base url: <https://rickandmortyapi.com/api>

The base url contains information about all available API's resources. All requests are `GET` requests and go over `https`. All responses will return data in `json`.

```
GET https://rickandmortyapi.com/api
```

```
{
  "characters": "https://rickandmortyapi.com/api/character",
  "locations": "https://rickandmortyapi.com/api/location",
  "episodes": "https://rickandmortyapi.com/api/episode"
}
```

*Imagen 2. Web de la API pública*

## 7. Funcionalidad de la página

La página web desarrollada permite explorar de manera sencilla e interactiva el universo de Rick y Morty a través de una interfaz estructurada en distintas secciones, todas ellas basadas en los datos que ofrece la API elegida, que es la API pública de la serie. La funcionalidad principal de la web se apoya en la recepción mediante la landing page (además del menú) que permite acceder directamente a las diferentes páginas principales: Episodios, Personajes, Ubicaciones, el acerca de nosotros y el formulario de contacto.

En la sección de Episodios, los capítulos están organizados por temporadas, lo que facilita la búsqueda y navegación por el contenido. Cada episodio muestra su título, fecha de emisión y código. Esta vista agrupada por temporadas mejora la experiencia del usuario al presentar de forma lógica y jerárquica la información ofrecida por la API.

En la sección de Personajes, se muestra un listado paginado de tarjetas con información básica de cada uno (nombre, imagen y especie). Al hacer clic sobre una tarjeta, se despliega un modal con información más detallada como el género, el estado y la ubicación del personaje. Esta funcionalidad permite una visualización rápida y ordenada sin necesidad de redirigir al usuario a otra página.

Por otro lado, la sección de Ubicaciones lista, de manera ordenada alfabéticamente y por letras, los distintos lugares mencionados en la serie, ofreciendo también un modal con información extendida al seleccionar cualquiera de ellos, como el tipo de ubicación del que se trata, su dimensión, y la población con la que cuenta.

Finalmente, en el Acerca de nosotros, se presenta de forma sencilla y un tanto humorística –algo por lo que la propia serie se identifica–, a los miembros que hemos participado en el desarrollo del proyecto, incluyendo una imagen/dibujo inspirada en la serie, una pequeña frase que cada uno de nosotros hemos elegido como presentación personal y el enlace mediante icono a cada uno de nuestros perfiles de GitHub. No obstante, el toque humorístico definitivo lo aporta la sección de FAQs (preguntas frecuentes), también accesible desde lo más bajo de la web, al igual que el apartado de Contacto en el que se incluye un formulario a rellenar por los usuarios indicando nombre, correo y la sugerencia que nos quieran transmitir.

En conjunto, la web busca un equilibrio entre funcionalidad y estética aplicada a la serie, priorizando sobre todo la claridad visual, el rendimiento y la facilidad de uso. Todas las funcionalidades están implementadas mediante React, lo que permite una carga dinámica de contenidos y una navegación fluida sin recargas de página.

## 8. HTML

Aunque HTML es el lenguaje base de cualquier estructura web, en nuestro proyecto su uso no ha sido directo. Al trabajar con React, no contamos con archivos HTML independientes que definan la estructura de cada página, como se haría en un desarrollo más tradicional. En su lugar, utilizamos archivos JavaScript (con extensión .jsx) que permiten inyectar directamente elementos HTML dentro del código, combinando lógica y estructura en un mismo lugar.

Este enfoque nos permitió construir las distintas secciones de la página web utilizando una sintaxis similar a HTML, pero con la potencia adicional que ofrece JavaScript para manejar datos, componentes y eventos. Cada uno de estos elementos se renderiza dinámicamente, lo que permite generar contenido en función del estado de la aplicación o de los datos obtenidos desde la API.

A continuación se muestra un ejemplo del uso de esta estructura dentro del proyecto:

```
return (  
  <div className="personajes-section ">  
    <div  
      className="parte-titulo-personajes"  
      aria-hidden={!selectedCharacter}  
      inert={selectedCharacter ? "true" : undefined}  
    >  
      <h1>Personajes</h1>  
      <p>  
        Aquí encontrarás la información sobre todos los personajes del  
        universo de Rick y Morty.  
      </p>  
    </div>  
  
    <div  
      className="sr-only"  
      aria-live="polite"  
      aria-hidden={!selectedCharacter}  
      inert={selectedCharacter ? "true" : undefined}  
    >  
      Estás en la sección de personajes. Usa las flechas arriba o abajo para  
      moverte por los personajes. Presiona Tab para saltar a la sección de  
      páginas y poder seleccionar otras páginas.  
    </div>  
  </div>  
)
```

*Imagen 3. Inyección de html en JavaScript*

En la imagen se puede ver cómo se devuelve una estructura de HTML, con divs, headers... mientras se conservan las estructuras de objetos de JS, que es la inyección de html a la que nos referimos.

Por último, realizamos una auditoría de nuestro HTML mediante el validador HTML (<https://validator.w3.org/nu/?doc=https%3A%2F%2Fflucasespitia.github.io%2Fwiki-rick-and-morty%2F>) como una herramienta de comprobación automática. Los resultados son muy positivos, puesto que no muestra ningún error ni warning y tan sólo hay algunas infos que nos muestran algunos detalles prácticamente sin importancia.

## 9. CSS

Para el diseño visual de la web hemos utilizado hojas de estilo en CSS, organizadas dentro de una carpeta específica llamada *styles*. Esta carpeta contiene los diferentes archivos *.css* que aplican estilos a cada una de las secciones y componentes del proyecto. Esta separación nos ha permitido mantener una buena organización del código, facilitando tanto la edición como la reutilización de estilos.

Desde el principio decidimos seguir una paleta de colores coherente, inspirada en el universo visual de *Rick y Morty*, con tonos verdes, negros y morados que evocan portales, galaxias y escenarios propios de la serie. Esto nos ayudó a mantener una estética uniforme a lo largo de toda la página, haciendo que la experiencia del usuario fuese más inmersiva y visualmente atractiva.

Además del uso de colores, también hemos integrado animaciones en algunas secciones para dar dinamismo a la interfaz, como por ejemplo transiciones suaves o efectos en botones e imágenes. A nivel de maquetación, hemos combinado técnicas de Grid y Flexbox según las necesidades del diseño. Por ejemplo, en la sección de personajes utilizamos *display: grid* para organizar las tarjetas de forma ordenada y adaptable, mientras que en la paginación optamos por *display: flex* para alinear los botones de navegación de forma centrada y fluida.

Este uso variado y estratégico del CSS ha contribuido de manera importante a que el diseño de la web sea limpio, responsivo y coherente con la temática.

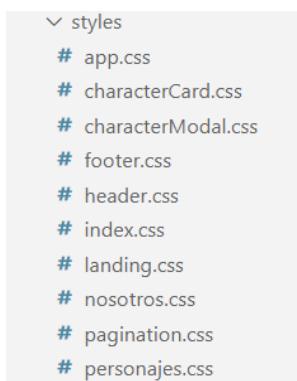


Imagen 4. Archivos css del proyecto

Por último, realizamos una auditoría de nuestro CSS mediante el validador CSS (<https://jigsaw.w3.org/css-validator/validator?uri=https%3A%2F%2Fflucasespitia.github.io%2Fwiki-rick-and-morty%2F&profile=css3svg&usermedium=all&warning=1&vextwarning=&lang=en>) como una herramienta de comprobación automática. Los resultados son bastante positivos, dado que no nos encuentra ningún error, lo cual nos valida la web como CSS level 3 + SVG.

## 10. JavaScript

JavaScript ha sido una parte fundamental en el desarrollo de nuestro proyecto, ya que, al utilizar React como biblioteca principal, toda la lógica y estructura de la aplicación se basa en este lenguaje. A través de JavaScript hemos gestionado la obtención de datos desde la API de Rick and Morty, el estado interno de la aplicación, la interacción del usuario con los distintos componentes y la navegación entre secciones.

Un recurso complementario ha sido React-Bootstrap, una integración entre Bootstrap y React que facilita la creación de componentes visuales reutilizables y adaptables. Gracias a esta librería hemos podido implementar elementos como tarjetas, botones, contenedores, filas, columnas y modales, todo ello con una integración directa con JavaScript mediante props y control de eventos.

En definitiva, JavaScript ha sido el motor detrás de toda la lógica funcional del proyecto, permitiéndonos desarrollar una web dinámica, interactiva y coherente con los objetivos establecidos.

## 11. Mobile First / Responsive Design

Desde las primeras fases de diseño del proyecto tuvimos claro que la web debía ser accesible y funcional en cualquier dispositivo, por lo que aplicamos una estrategia de diseño responsive, adaptando la interfaz a distintos tamaños de pantalla. Aunque no seguimos estrictamente una metodología Mobile First desde el inicio, sí nos aseguramos de que todos los componentes y secciones fuesen completamente adaptables a dispositivos móviles.

Para ello, hemos combinado el uso de CSS Flexbox y Grid, así como las utilidades ofrecidas por React-Bootstrap, que facilita la implementación de un diseño adaptable gracias a su sistema de rejillas y componentes responsive. Esto nos permitió construir una estructura flexible, con elementos que se reorganizan automáticamente según el ancho de pantalla.

Durante el proceso de desarrollo, realizamos múltiples pruebas en distintos dispositivos y tamaños de ventana para verificar que la web se visualiza correctamente en móviles, tablets y ordenadores de escritorio. También prestamos atención a la usabilidad táctil en pantallas pequeñas, asegurándonos de que los botones y enlaces sean fácilmente accesibles.

En resumen, la web responde adecuadamente a diferentes contextos de visualización, manteniendo su funcionalidad y diseño en cualquier dispositivo, lo cual garantiza una experiencia de usuario óptima y consistente.

## 12. Accesibilidad

A lo largo del desarrollo del proyecto, hemos procurado tener en cuenta diversos aspectos de accesibilidad web con el objetivo de que la página sea usable por el mayor número de personas posible, incluidas aquellas con alguna discapacidad. Este compromiso se ha visto reflejado tanto en el diseño visual como en la implementación técnica de los componentes.

Desde el punto de vista de las pautas de accesibilidad para el contenido web (WCAG), hemos centrado nuestros esfuerzos principalmente en tres principios fundamentales: perceptibilidad, operabilidad y comprensibilidad. Para lograrlo, hemos utilizado colores con buen contraste, una tipografía legible y tamaños de fuente adaptables, asegurando que la información sea fácilmente distinguible. También hemos mantenido una estructura clara y coherente entre las distintas páginas, facilitando la navegación y la comprensión del contenido.

En cuanto a la operabilidad, nos aseguramos de que la web pueda ser utilizada fácilmente con teclado y que los elementos interactivos como botones, enlaces y modales tengan descripciones comprensibles y un orden lógico en la navegación. Además, hemos probado la funcionalidad básica del sitio con un software lector de pantalla, para comprobar que el contenido es accesible para usuarios con discapacidad visual.

Para reforzar la evaluación técnica de la accesibilidad, realizamos una auditoría automatizada mediante Lighthouse, herramienta integrada en el navegador Chrome. Los resultados son bastante positivos en todas las secciones: 93 en la página principal, 94 en la de personajes, 92 en la de ubicaciones, 91 en la de episodios, 96 en la de acerca de nosotros y 96 en la de las FAQs. Estos valores, creemos e interpretamos, indican un buen nivel de cumplimiento de las recomendaciones actuales en accesibilidad web.

Por otro lado, realizamos otra auditoría mediante la herramienta de comprobación automática TAW ([tawdis.net](http://tawdis.net)), propuesta por Marcela Vega, la cual no nos indica ningún problema detectado y algunos warnings, si bien es cierto que en algunos apartados nos indica que no ha podido realizar la revisión automática y en otros indica un “na” que significa que no es aplicable.

La última de las auditorías realizadas es la de Achecker ([achecks.org](http://achecks.org)), también propuesta por Marcela Vega, cuya Accessibility Review, basada en las guidelines del WCAG 2.0 (level AA), no nos encuentra ningún error reconocido ni probable, y nos muestra 13 problemas potenciales que deberíamos tratar en algunos casos concretos.

En conjunto, consideramos que nuestra web cumple satisfactoriamente los requisitos fundamentales de accesibilidad, si bien somos conscientes de que siempre existen áreas de mejora en este ámbito, especialmente en lo relativo a validaciones manuales más exhaustivas y accesibilidad cognitiva.

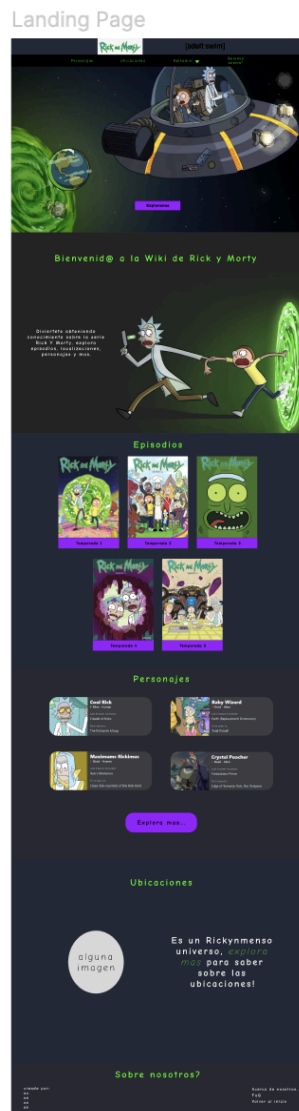
## 13. Usabilidad

Durante el desarrollo del proyecto, aplicamos distintas técnicas de análisis de usabilidad en varias fases clave: investigación, diseño y validación. En la fase inicial, realizamos una breve investigación informal entre compañeros para identificar qué tipo de estructura esperaban encontrar en una página basada en una API de contenido multimedia como la de Rick and Morty. Esta exploración nos ayudó a decidirnos por una interfaz tipo wiki, jerárquica y fácil de recorrer, que facilitase el acceso rápido a información sobre personajes, ubicaciones y episodios.

En la fase de diseño, utilizamos prototipos en Figma para visualizar la estructura general y evaluar de forma anticipada cómo sería la navegación y disposición de contenidos. Finalmente, durante la validación, realizamos pruebas de uso entre los propios miembros del equipo y observamos cómo otras personas interactuaban con la web. Estos test informales nos permitieron ajustar detalles como el uso de modales en lugar de páginas nuevas para mostrar detalles, o mejorar la ubicación de botones y filtros para hacer más intuitiva la experiencia de usuario.

Así como en el apartado de wireframes y prototipado especificábamos el prototipado básico e inicial con el que partimos del proyecto, es turno ahora de comparar el prototipado más avanzado y final a partir del cual comenzamos a diseñar la web con el resultado final de la propia web y focalizarnos un poco en las diferencias y la razón de las mismas.





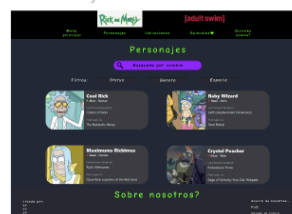
## Ubicaciones



## Episodios



## Personajes



## Acerca de Nosotros



Imagen 5. Prototipado final y actualizado de todas las páginas en Figma

La landing page creemos que ha quedado, exceptuando el footer, bastante similar y fiel a la idea original que tuvimos, que en su momento consideramos lo suficientemente atractiva, visual e intuitiva. Estamos bastante satisfechos con el trabajo realizado por esa parte.

En cuanto a las secciones más concretas, la de ubicaciones, ha quedado de una manera un tanto diferente, si bien sí mantiene la estructura de organizar las ubicaciones por letras ordenadas alfabéticamente. La de episodios creemos que ha mejorado considerablemente respecto a la idea inicial, puesto que ahora es bastante más agradable visualmente.

La sección de personajes queda algo parecida a como habíamos diseñado inicialmente, pero también contempla algunas mejoras, puesto que, debido a la cantidad de personajes que manejamos, era inviable mostrar todos sus datos de manera directa (ahora lo hacemos mediante una card que se abre clicando en el personaje en cuestión, igual que con cada ubicación y cada episodio).

Por último, la sección del acerca de nosotros la hemos diseñado bastante en la línea de lo que planteamos en una primera instancia, dándole un toque personal a la web y sin olvidarnos de ese tono humorístico que también contempla la sección de FAQs, la cual no tuvimos en cuenta desde los momentos iniciales pero acabamos incluyendo durante el proceso de desarrollo debido a una idea surgida más espontáneamente, así como la del formulario de contacto.

## 14. Implementación mediante REACT

React fue una de las tecnologías que se nos introdujo durante las sesiones prácticas de la asignatura. A lo largo de varias clases estuvimos trabajando con esta librería y aprendiendo su funcionamiento, lo que nos permitió entender bien su lógica basada en componentes y cómo se gestiona el flujo de datos. A medida que nos familiarizábamos con su estructura, fuimos comprobando que era una herramienta ideal para el tipo de proyecto que teníamos entre manos.

Una vez comenzamos el desarrollo, tuvimos claro que React era la mejor opción para crear nuestra página web. Su modularidad y eficiencia nos ofrecían todo lo necesario para construir una aplicación organizada, dinámica y escalable. En las primeras etapas del proyecto, nos centramos en definir la estructura de carpetas y directorios que íbamos a utilizar, dividiendo claramente los archivos en componentes, estilos, páginas, constantes y recursos visuales. Esta organización fue fundamental para poder trabajar de forma fluida y mantener el código limpio y entendible a lo largo del desarrollo.

Una de las grandes ventajas de trabajar con React es que permite visualizar en tiempo real los cambios que se van realizando en el código. Esta característica nos ayudó a testear de forma continua cada una de las funcionalidades y a ir corrigiendo errores de forma inmediata. Además, la posibilidad de reutilizar componentes en distintas partes del proyecto facilitó mucho la implementación de ciertas secciones comunes, como los menús o las tarjetas informativas.

También, uno de los elementos clave ha sido el uso de React Hooks, como `useState` y `useEffect`, que nos permitían manejar el ciclo de vida de los componentes, controlar la carga de datos y actualizar dinámicamente la interfaz en función de los cambios en el estado. Gracias a estos hooks, por ejemplo, se cargan los personajes de forma paginada o se actualizan los episodios al cambiar de temporada.

Además, hemos hecho uso de React Router, una librería adicional de JavaScript que nos ha permitido gestionar la navegación entre distintas rutas de la aplicación de forma dinámica y sin necesidad de recargar la página. Esto ha contribuido significativamente a mejorar la experiencia de usuario, haciendo que la navegación entre secciones sea fluida e instantánea.

En conjunto, el uso de React no sólo simplificó el proceso de desarrollo, sino que también nos permitió construir una web interactiva y rápida, con una experiencia de usuario bastante pulida.

## 15. SEO

A lo largo del diseño y desarrollo de la página web, hemos tenido en cuenta distintos aspectos relacionados con el SEO con el objetivo de mejorar la visibilidad de nuestro sitio en buscadores y facilitar su indexación. Uno de los primeros pasos fue establecer una estructura de rutas claras y amigables utilizando React Router, lo que permite que cada sección de la web tenga una URL específica y coherente con su contenido.

También le prestamos atención a aspectos como el uso de etiquetas semánticas en los componentes, la jerarquía correcta de encabezados y la inclusión de títulos de página descriptivos mediante la etiqueta title en el archivo index.html.

Además, optimizamos el peso de las imágenes y cuidamos el tiempo de carga para mejorar tanto la experiencia del usuario como los criterios que los buscadores utilizan para clasificar sitios web. Si bien el uso de React puede suponer una limitación para el SEO tradicional, intentamos compensar este impacto con una estructura bien organizada y recursos optimizados.

Realizando la auditoría del SEO mediante la herramienta Lighthouse integrada en el navegador Chrome, nos obtiene un resultado de 100, lo cual, si bien es cierto que nos lo argumenta qué audits nos ha aprobado satisfactoriamente, cuáles no son aplicables y cuáles habría que valorar manualmente, creemos que es un resultado bastante poco realista.

Es por ello que buscamos una alternativa diferente y externa a Lighthouse, y nos encontramos con Seobility (<https://freetools.seobility.net/en/seocheck/check?url=https%3A%2F%2Fflucasespitia.github.io%2Fwiki-rick-and-morty%2F&crawltype=1>), la cual sí es más crítica y nos valora en 55 nuestro SEO. Sabemos que en caso de disponer algo más de tiempo para este proyecto, y si pretendiésemos que esta web fuese competitiva en el 'mercado' al ser un proyecto más serio u oficial, tendríamos que dedicarle mucho esfuerzo a la mejora de este apartado.

No obstante, hay algunos puntos bastante positivos, como el logro del 98% de las meta specifications, un aspecto bastante importante en el SEO, o que realmente las 5 tareas importantes que nos detalla la auditoría de cara a mejorar el SEO serían aspectos, creemos, fáciles y no demasiado complejos de corregir, como por ejemplo añadir links internos para mejorar la usabilidad y que los bots puedan llegar a todas las áreas de nuestra web de una manera más rápida, o usar redirecciones para dirigir el tráfico a URLs con el mismo dominio que el nuestro.

## 16. Rendimiento

Durante el diseño y desarrollo del proyecto, procuramos tener en cuenta diversos aspectos relacionados con el rendimiento, con el objetivo de ofrecer una experiencia de usuario fluida y minimizar los tiempos de carga. Para ello, optimizamos las imágenes utilizadas, evitando archivos pesados y seleccionando formatos adecuados.

Además, estructuramos el código de forma modular para facilitar la carga parcial de componentes en lugar de renderizar todo el contenido de golpe. También aprovechamos el sistema de paginación en secciones como la de personajes para evitar la carga masiva de datos desde la API.

Para evaluar de forma objetiva el rendimiento de la página, realizamos una auditoría automatizada mediante Lighthouse, la herramienta de Chrome. Los resultados obtenidos fueron los siguientes: 83 en la página principal, 82 en la de personajes, 87 en la de ubicaciones, 88 en la de episodios, 75 en el acerca de nosotros y 82 en las FAQs. Estos valores indican un rendimiento general aceptable, aunque hay margen de mejora.

Entre los aspectos que Lighthouse nos señala como susceptibles de mejora se encuentran la carga inicial de JavaScript, el aprovechamiento de caché del navegador y el uso de formatos de imagen aún más eficientes en ciertas secciones.

Una posible línea de optimización futura sería el uso de lazy loads para cargar las imágenes tan sólo cuando se van a visualizar o emplear herramientas de minificación para reducir el tamaño del código JavaScript y CSS. También se podrían explorar soluciones de prerrenderizado o Server-Side Rendering en caso de migrar el proyecto a entornos más avanzados de producción.

## 17. Despliegue

Para finalizar, tras valorar las diferentes opciones, hemos decidido acabar desplegando el proyecto mediante GitHubPages. Al no realizar esto en última instancia y con el proyecto ya finiquitado, que por contradictorio que parezca podría haber acabado siendo un tanto más sencillo, ha habido algunos procesos algo tediosos de realizar debido a la transferencia del proyecto desde el GitLab al GitHub.

Aunque de primeras no parezca algo excesivamente complicado ni prolongado, el haber desplegado la página web sin haberla acabado supone tener más cuidado al subir los archivos actualizados al GitHub, teniendo que prestarle atención a cuáles han sido los archivos modificados y transferir sólo aquellos, puesto que, de intentar transferir todos cada vez, se convierte en un proceso de pegado de carpetas bastante largo debido al tamaño de las mismas y a la gran cantidad de archivos que estamos manejando.

En cualquier caso, hemos acabado realizando todo este proceso y consiguiendo el despliegue definitivo de manera satisfactoria. La URL directa es la siguiente:

<https://lucasespitia.github.io/wiki-rick-and-morty/>

