

## Prueba - Rick and Morty, Info de personajes

- Para realizar esta prueba debes haber estudiado previamente todo el material disponibilizado correspondiente al módulo.
- Una vez terminada la prueba, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Puntaje total: 10 puntos.
- Desarrollo prueba:
  - La prueba se debe desarrollar de manera Individual.
  - Para la realización de la prueba necesitarás apoyarte del archivo Apoyo Prueba - Rick and Morty, Info de personajes.

### Descripción

Una parte muy importante del desarrollo web en la actualidad, es el consumir datos desde una API y hacer algo con esos datos. Por ende, en esta oportunidad utilizarás la siguiente [URL](#) de “Rick and Morty”, la cual nos devolverá la información que trabajaremos en esta prueba. Específicamente, obtendremos todos los personajes y el detalle de cada una de las siguientes direcciones URL de la API: [Personajes](#) y [Detalles](#). Por lo que el propósito de esta prueba es crear una función global autoejecutable (IIFE), que realice la conexión mediante fundaciones privadas a la API, instancia un nuevo objeto perteneciente a una clase hija (ver imagen 3 - diagrama de clases) y desde la una función pública se muestre la información en la página web dinámicamente.

En la imagen número uno (1) mostrada posteriormente, podrás ver cómo quedará el proyecto final, es decir, una lista de todos los personajes que nos devuelve la API (los primeros 20 personajes), y al pulsar sobre la imagen (hacer un clic) de cada uno de ellos se mostrará una ventana modal con toda la información del personaje (id, nombre, status, especie, género, creación, origen, cantidad de episodios) como se aprecia en la imagen número dos (2). Mientras que al lado de la imagen solo se mostrará el id y la especie a modo de lista.

Así mismo, en el material de apoyo denominado “Apoyo Prueba - Rick and Morty, Info de personajes.”, encontrarás todos los archivos y carpetas necesarias para desarrollar todo el proyecto. Es decir, ya se encuentran disponibles los archivos “index.html, style.css” con toda la configuración básica, así como las imágenes iniciales del sitio web. Por lo que solo debes preocuparte por trabajar directamente en la programación con JavaScript para poder consultar la API y mostrar la información dinámicamente de acuerdo a los requerimientos indicados al final de este archivo.

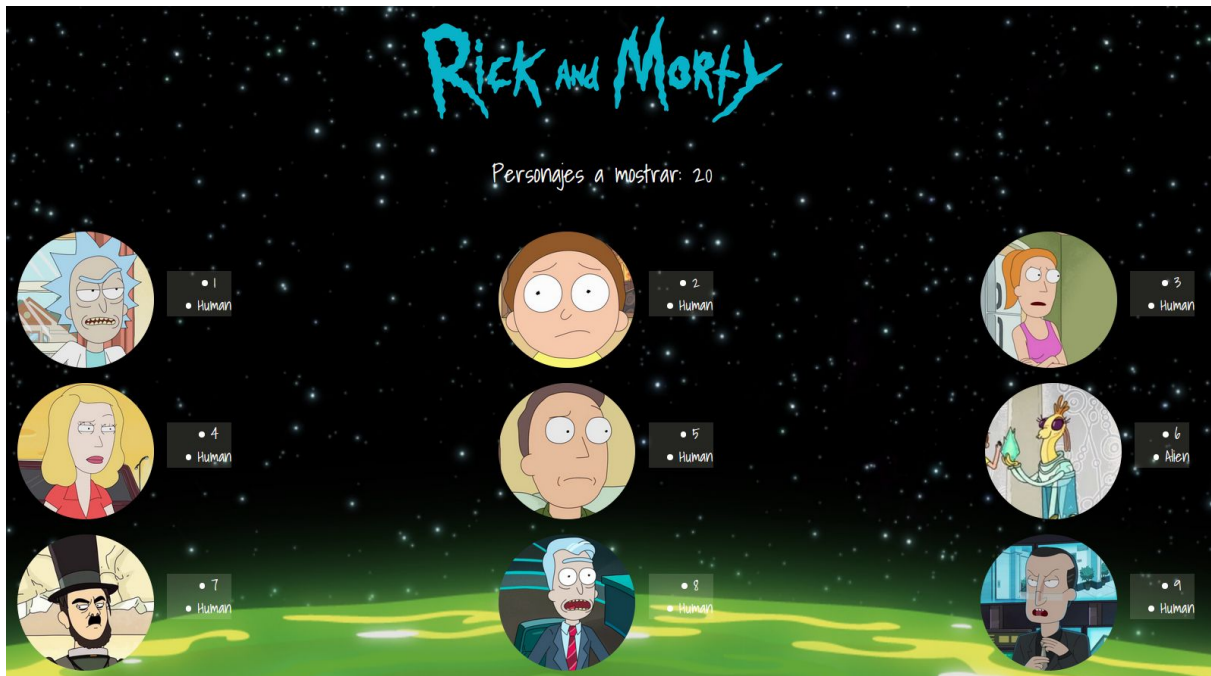


Imagen 1. Vista principal del proyecto.

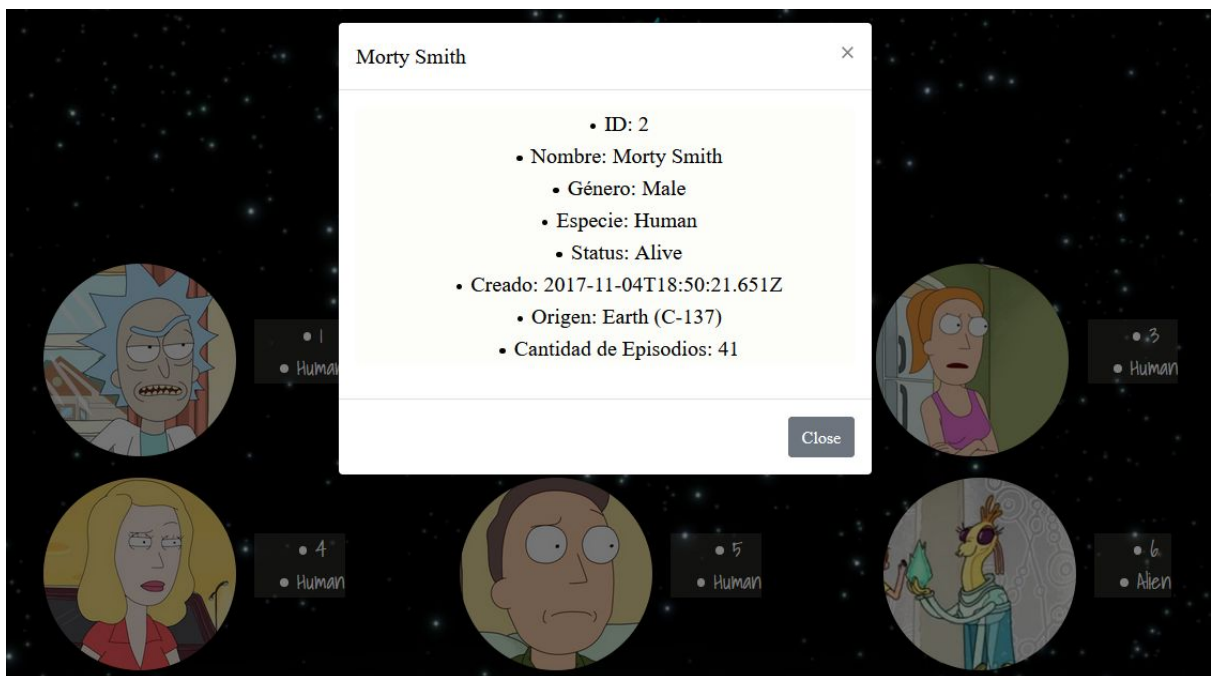


Imagen 2. Vista de ventana modal al hacer un clic sobre una imagen.

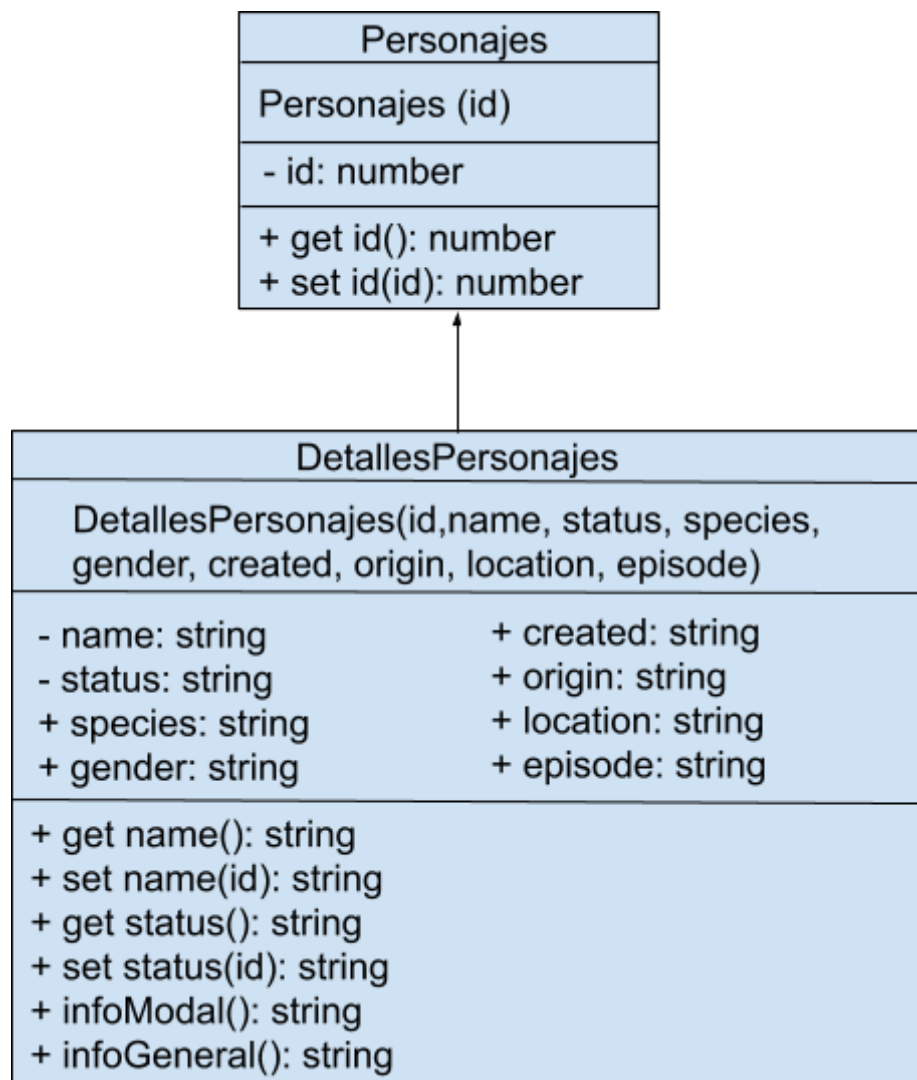


Imagen 3. Diagrama de Clases.

## Requerimientos

1. Crear un archivo con el nombre de "personajes.js" que exporte una clase padre con el nombre de "Personajes" que reciba en su constructor un id de personaje. Agregando métodos get y set para obtener el atributo además de closures para proteger el atributo id. **(1 Punto)**
2. Crear un archivo con el nombre de "detallesPersonajes.js" que exporte una clase hija con el nombre de "DetallesPersonajes" que reciba en su constructor las propiedades indicadas en el diagrama de clases. Agregando métodos get y set para obtener los atributos name, status, species y gender. Utilizando closures solo para los atributos name y status **(1 Punto)**

3. Crear un archivo "main.js" que importe la clase hija y realice una función IIFE con el nombre de "llamadoPersonajes". (1 Punto)
4. Agregar tres variables privadas a la función IIFE, que contenga la URL de la API "<https://rickandmortyapi.com/api/character/>", así como la captura del elemento donde se mostrarán los resultados y una variable sin valor alguno para almacenar el resultado que retorne el llamado de la API. (1 Punto)
5. Agregar dos funciones asíncronas privadas implementando async/await y el bloque try/catch, para conectarse a las dos URL de la API. (1 Punto)
6. La primera función privada debe conectarse mediante el método fetch a la URL de la API "<https://rickandmortyapi.com/api/character/>", pasando el resultado a la variable privada creada sin ningún valor y retornado el resultado de la promesa en la función. (1 Punto)
7. La segunda función privada, deberá recibir el id a consultar del personaje para la URL de la API "<https://rickandmortyapi.com/api/character/id>". Luego enviar los datos recibidos, específicamente "id,name,status,species,gender,created,origin,location,episode", para crear una nueva instancia de la clase hija "DetallesPersonajes", para así hacer el llamado al método "infoGeneral" que mostrará al lado de la imagen la información del personaje, como el número del id y la especie. (1 Puntos)
8. Crear dos funciones públicas dentro del retorno de la IIFE que permitan realizar las consultas a la API y mostrar la información. (1 Puntos)
9. La primera función pública será asíncrona, consultado a la primera función privada por los datos generales de todos los personajes de la serie. Luego, mediante la inyección de código HTML desde JavaScript con el DOM, mostrar la imagen y al mismo tiempo hacer el llamado a la segunda función privada para pasar el id de cada personaje y mostrar la información al lado de la imagen. (1 Puntos)
10. La segunda función pública, será llamada fuera de la IIFE después de transcurrir dos segundos (utilizar setTimeout), para así borrar el elemento circular o spinner y consultar a la variable privada que contiene los datos de todos los caracteres para que muestre al lado del título de personajes la cantidad de personajes que se muestran en pantalla. (1 Puntos)

## Requerimientos Opcionales

1. Agregar closures a los atributos species y gender de la clase hija para evitar alteraciones no deseadas.
2. Agregar métodos get a todos los atributos de la clase hija.
3. Agregar un método "infoModal" que permita retornar más información del personaje, como el id, nombre, género, especie, estatus, creación, origen y cantidad de

episodios en una lista. Este método será llamado para retornar los atributos indicados dentro de una modal.

4. A la primera función pública, agregar la cabecera para poder activar una ventana modal cuando el usuario realice un clic sobre cualquiera de las imagen disponibles en pantalla.
5. En la segunda función privada, crear la estructura HTML que complete la ventana modal, haciendo el llamado al método de la clase hija denominado "infoModal".