Introduction

In this project you must create your own website for the famous television series **Rick and Morty**. All the information to display on the page must be consumed by a **third-party API**.

What are the main objectives in this project?

- Obtain information from **third-party API's**
- Improve your knowledge of **HTML**, **CSS** and **Javascript**
- Learn more about how to integrate **APIS** and its interaction with the client
- Improve your skills in obtaining **asynchronous information by using AJAX**
- Structure a WEB project in a clear and orderly manner

1. General analysis

You must create a **website** capable of displaying all the information in the series (**episodes**, **characters**, **locations**) in an orderly manner so that you can **navigate through the episodes and characters**.

1.1. Rick and Morty API

The **third party API** that you must use to make the requests is Rick and Morty.

In this API you will see the **HATEOAS** and **pagination** concept applied in its structure. The main objective is to **show the Rick and Morty episodes** and navigate through the information it provides us.

The **information** that **you should obtain** from the **API** of each section is as follows:

Episodes

In this section you will obtain information about the chapters of Rick and Morty:

- Name
- Air date
- Episode code
    - Ex: “S01E01”
- Created date

Characters

In this section you will obtain information about the characters of Rick and Morty:

- Name
- Status
- Species
- Gender
- Image

Locations

In this section you will obtain information about the locations of Rick and Morty:

- Name
- Type
- Dimension

Depending on which part you show the data in, you should get only part of the information shown above.

1.2. Web structure

The initial information to show in this project will be the chapters, from which you must navigate within each one of them to show all the information related to them.

The page will consist of three main sections:

Header

The header section should show at least the following information:

- Project title

Sidebar

The sidebar section should show at least the following information:

- List of the Rick & Morty episodes of the first API pagination
- Button that will load the next page of episodes below the previous one

- The user will be able to scroll in the sidebar to see the entire list of chapters

Main container

The section of the main container will be the one that contains the detail of the elements.

1.3. Steps for development

The steps to follow for the development of this project are detailed below:

Step 1

To start you will have to **create the web structure** following the one mentioned above.

Once the web structure has been implemented, in the **sidebar** you should show an **episode list**, allowing the user the option to **show more episodes** through a button located after the last item in the list using the **pagination of the API**. You have to take into account that **the user will be able to scroll** in this list if it exceeds the height of the screen.

Step 2

When the **user clicks** on an item in the **episode** list, you should display the information of the episode in question in the **main container**.

Information you must show for the selected **episode**:

- Name
- Air date
- Episode code
    - Ex: “S01E01”

Step 3

Once the main information of the **episode is shown** in the **main container**, you should also **show the following information** for each of the characters that appear in that episode:

- Character name
- Character status
- Character specie
- Character image

Optional steps:

Once you have finished **step 3**, the following steps indicated below are **optional**.

Step 4

In this first optional step, if the **user clicks on a character**, you should show the following **information about the character** in the main container:

- Name
- Status
- Specie
- Gender
- Image
- Origin name
    - Ex: Earth (C-137)

Step 5

Once the previous step of the characters has been implemented, you must **show a button** that will load the **detailed information of the origin location** in the **main container**:

- Location name
- Location type
- Location Dimension

- Residents list
    - If you select an item from this list, the detail of the selected character will be shown in the main container (note that the functionality of displaying the character information has already been applied previously)

Step 6

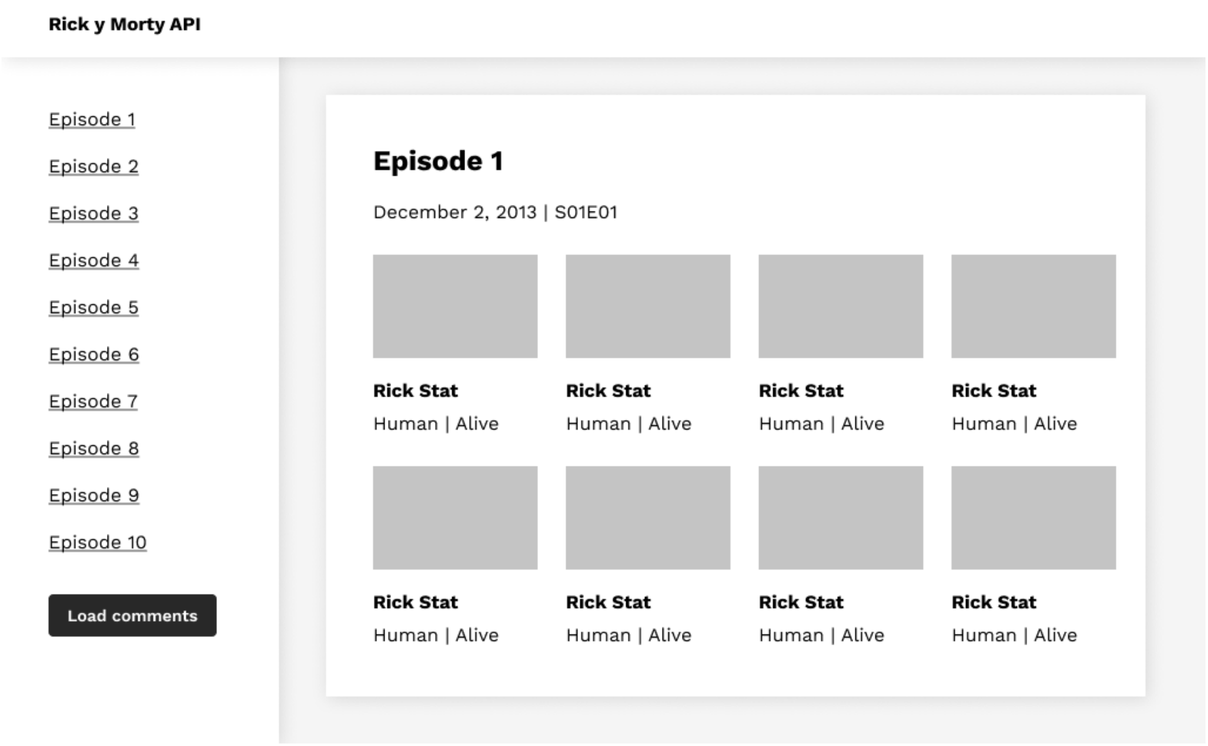
In the final step, you must **show in the character view** the **list of episodes** in which this **character appears**.

- **Episodes list**:
    - Episode name
    - Episode code

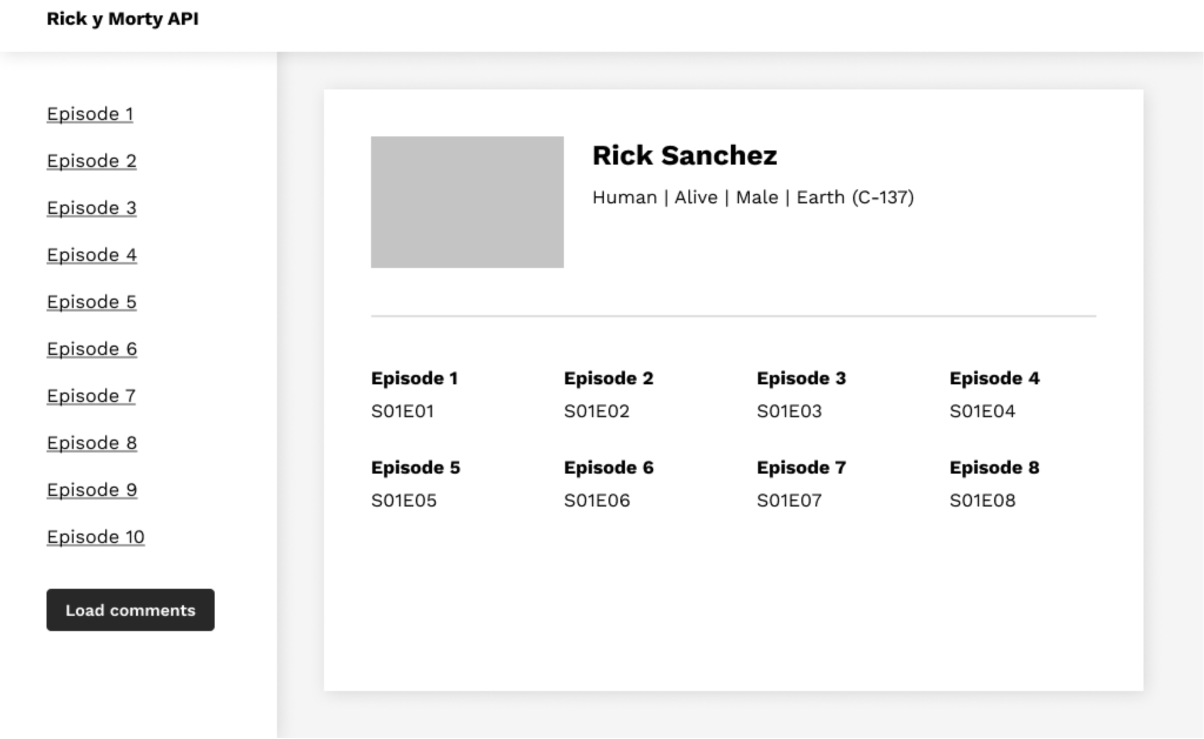
If you click on an episode in the list, the **information of that episode** will be shown in the **main container** as you have **previously implemented**.

1.4. Wireframes

Main view



Character view



Rick y Morty API

Episode 1

Episode 2

Episode 3

Episode 4

Episode 5

Episode 6

Episode 7

Episode 8

Episode 9

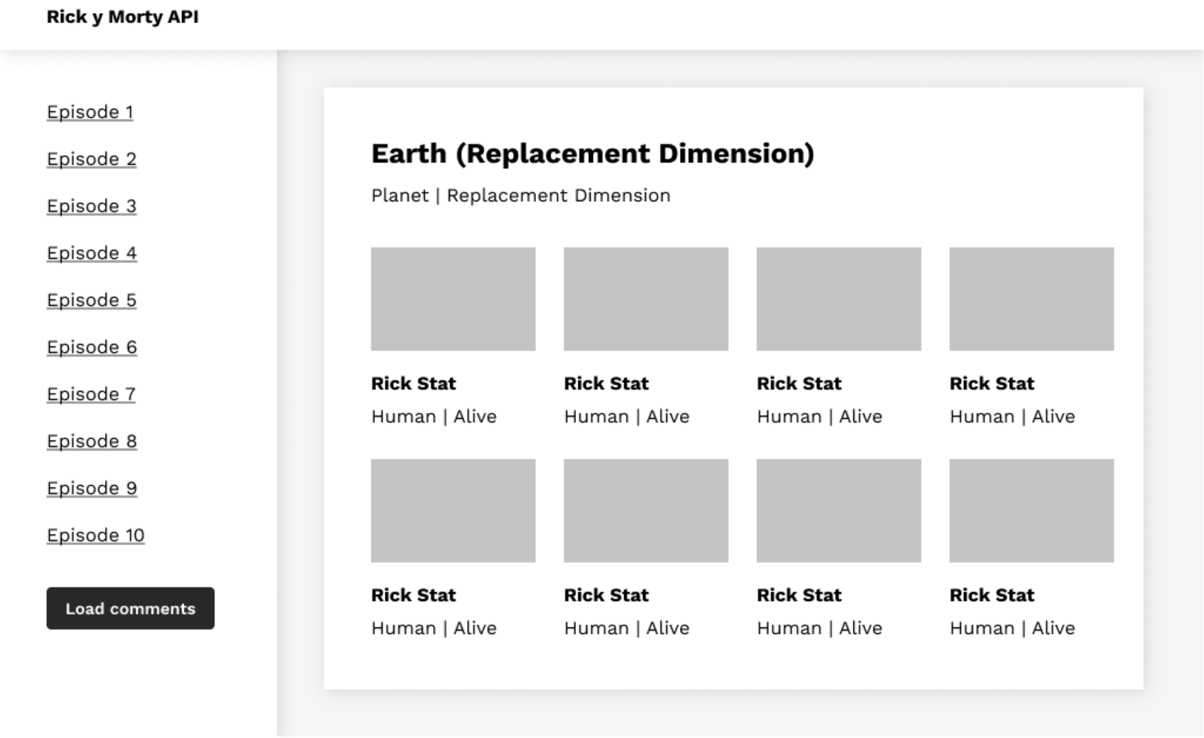
Episode 10

Rick Sanchez

Human | Alive | Male | Earth (C-137)

Episode 1 S01E01	**Episode 2** S01E02	**Episode 3** S01E03	**Episode 4** S01E04
Episode 5 S01E05	**Episode 6** S01E06	**Episode 7** S01E07	**Episode 8** S01E08

Location view



Rick y Morty API

Episode 1

Episode 2

Episode 3

Episode 4

Episode 5

Episode 6

Episode 7

Episode 8

Episode 9

Episode 10

Earth (Replacement Dimension)

Planet | Replacement Dimension

Rick Stat Human | Alive	**Rick Stat** Human | Alive	**Rick Stat** Human | Alive	**Rick Stat** Human | Alive
Rick Stat Human | Alive	**Rick Stat** Human | Alive	**Rick Stat** Human | Alive	**Rick Stat** Human | Alive

2. Project organization

Next you will have to create a document where you can **explain in detail** how the current project is **organized**. It is important that it be updated throughout the life of the project.

The document must include at least:

- Requirements documentation
- A PDF version is required within the project folder for the project documentation
- Record of incidents that were detected during project execution
- Documentation of the API used in the project
- Record of lessons learned

3. Requirements

- You must create a **Postman collection** with all the requests that you need for this project
- You must use **jQuery** to edit the DOM
- You must use **Axios**
- It is essential that your code is well documented
- All code including **comments** needs to be written in **English**
- Use a **camelCase** code style
- Never use inline styles in **HTML**
- Delete unused files
- You can use only one HTML file
- The web application must be responsive

- Organize CSS files in a clear and orderly manner

4. Development

Then you must get down to work and develop a project, it is important that you review the requirements and comply with all of them.

5. Deliverables

To evaluate the project you will need the following deliverables:

- Project files with a **.zip** compressed folder.
  - Project folder + Postman collection
- Project documentation
- A **presentation** in **Google Slides** explaining:
  - Explain what lessons you've learned during this project
  - What problems have you encountered when developing this project?

You must send the previous deliverables to the following email:
pau@assemblerschool.com

6. Resources

- AXIOS: https://github.com/axios/axios
- AXIOS CDN: https://unpkg.com/axios/dist/axios.min.js
- AJAX: https://www.w3schools.com/xml/ajax_intro.asp