**Planeación para el desarrollo de la página:**

1. **SOBRE LA API**

La página va a realizar consultas a la página de: <https://rickandmortyapi.com/api>

Tiene lo siguiente:

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

1. Cada página de estas dependiendo de la elegida devuelve personajes, localizaciones y episodios, está en formato JSON y se divide por número de páginas.
2. Se pueden hacer filtros extra agregando al final de la url (ya estando en alguna de personajes/localizaciones y episodios), por ejemplo: /?name=x, siendo x el nombre del personaje/episodio/Localización y poner “&” para agregarle otra consulta como /?page=y, siendo y el número de página.
3. **SOBRE LA ESTRUCTURA DE LOS DATOS**

Los datos en la API los podemos dibujar de la siguiente manera:

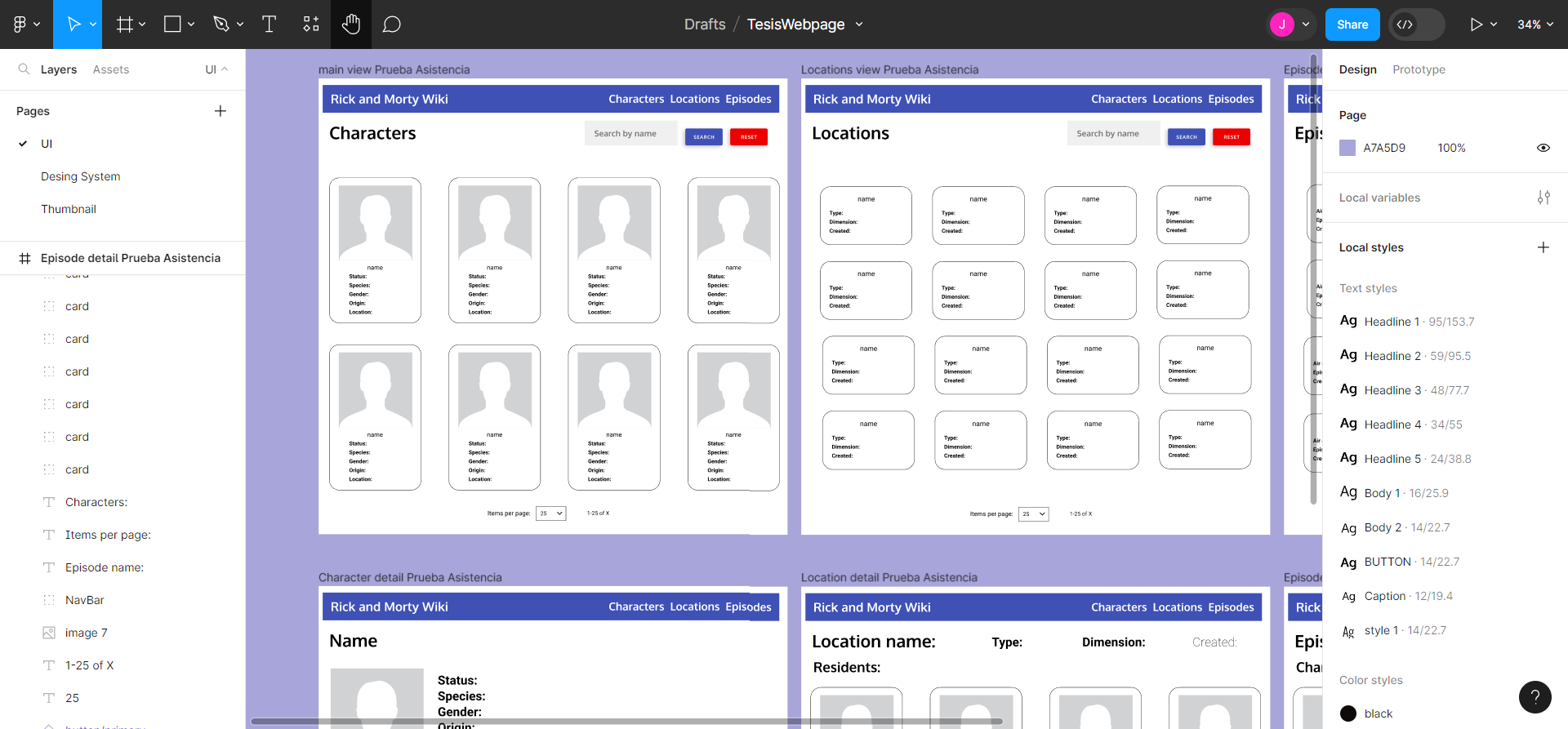
Diagrama

Descripción generada automáticamente

1. **SOBRE LA VISTA DE LA PÁGINA**

Se realizó el diseño mediante la aplicación FIGMA:

<https://www.figma.com/file/9NjAoYvxIbjEtHtq5oy2n0/TesisWebpage?type=design&node-id=1781-725&mode=design&t=zwUYGRkwkTZwpfqJ-0>



Se obtuvieron las siguientes vistas:

Main view / Characters:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Locations:

Interfaz de usuario gráfica

Descripción generada automáticamente

Episodes:

Diagrama

Descripción generada automáticamente

Character detail:

Diagrama

Descripción generada automáticamente

Location detail:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

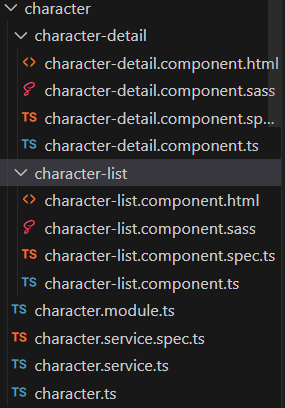
Episode detail:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. **SOBRE LA PROGRAMACIÓN DE LA PÁGINA**

Se van a tener los módulos de la NavBar, Character, Episode y Location, los últimos 3 se distribuyen de la siguiente manera:



Principalmente se basa en una estructura de módulos y componentes con una clase “character.ts” siendo esta un DTO (Data Transfer Object) para poder acomodar los datos a la página, luego se establece un servicio “character.service.ts” encargada de realizar las peticiones a la API y obtener los datos que se requieren. Finalmente, nos dividimos en el detalle (detail) encargado de manejar y dibujar la página relacionada al detalle de cada objeto obteniendo los datos del servicio y acomodándolos a su uso, y el listado (list) que se encarga de la búsqueda por parte del usuario y el dibujado de cada uno de los datos dependiendo de la clase.

Por otro lado, para mostrar la página se utilizó HTML con diseño de estilos SASS y apoyo de la librería de Angular Material.