1. **Las capas de la aplicación (por ejemplo capa de persistencia, vista, de aplicación, etc) y qué clases u objetos pertenecen a cual.**

Capa de persistencia y de lógica: PHP, clase: calcular

Capa de cliente: PHP, HTML5 y Javascript

1. **La responsabilidad de cada clase u objeto creada**

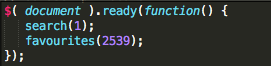
**Funciones**

* **Search:** Se encarga de la carga inicial de personajes, a su vez pasandole como parametro el nombre del personaje permite que se realice el filtro.
* **Modal:**  Permite la carga de uno de los comics del personaje asociado.
* **Favourites:** Permite la carga del comic en la seccion de favoritos.

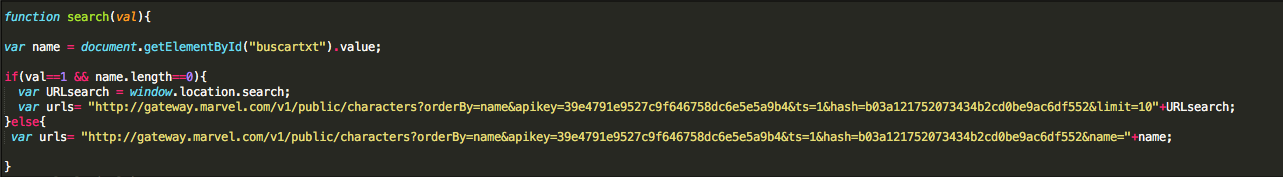
1. **Descripción de la solución**

* **Carga Personajes:**

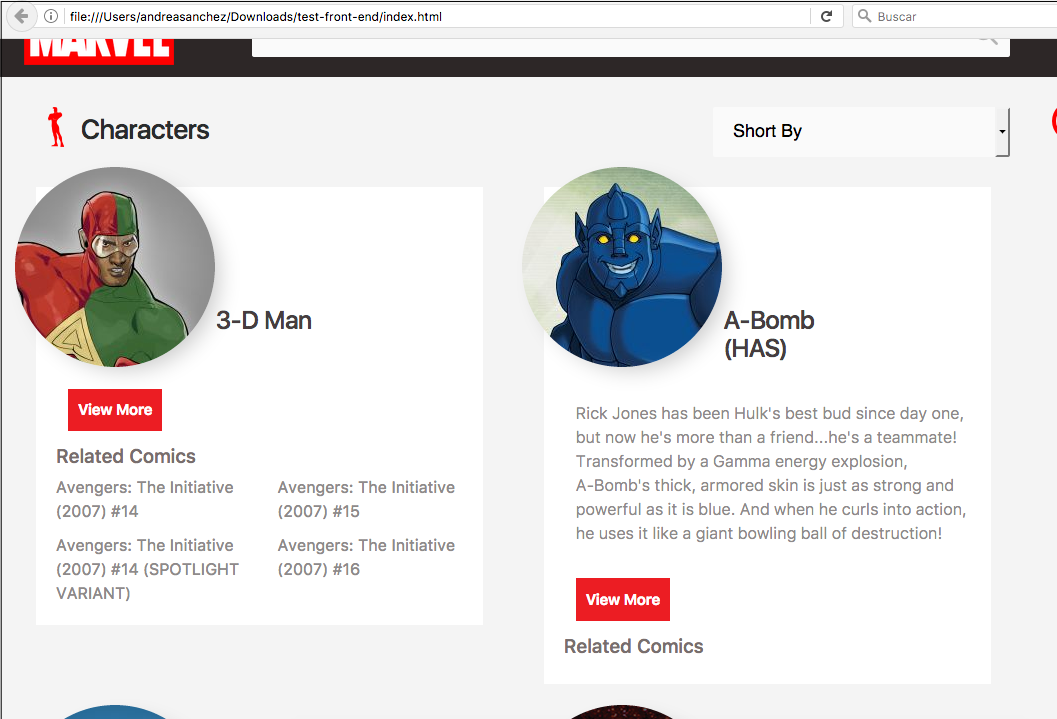
La carga inicial se realiza a traves de la funcion:



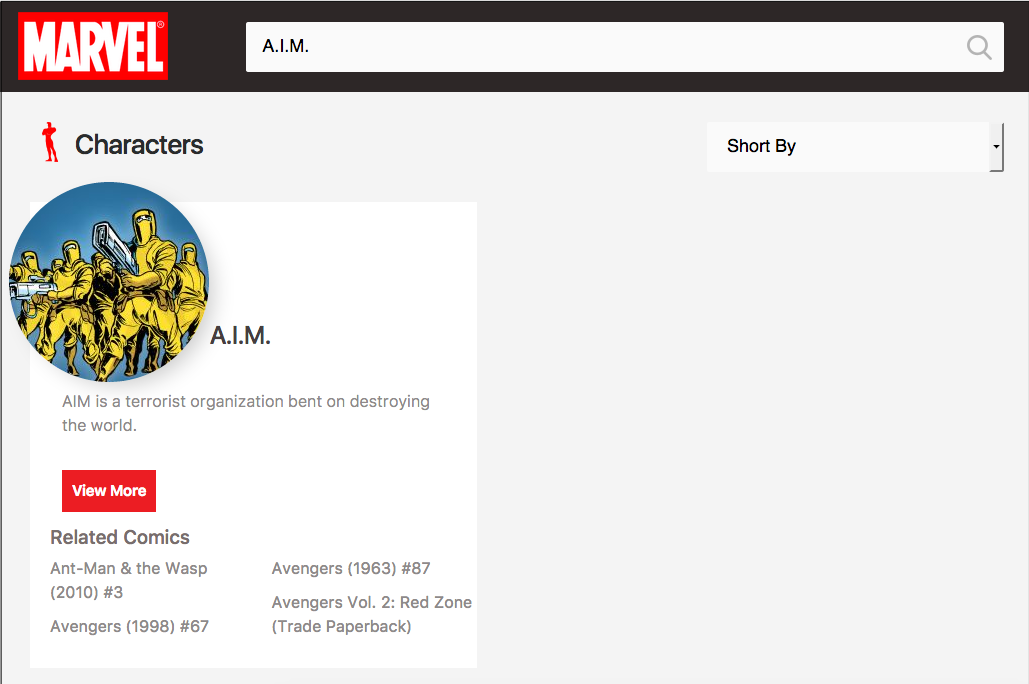
Esta a su vez entra a la funcion search en la cual se le pasa un parametro con el fin de saber si debe traer todos los personajes o uno en especifico.



dependiedo el parametro consulta la información.



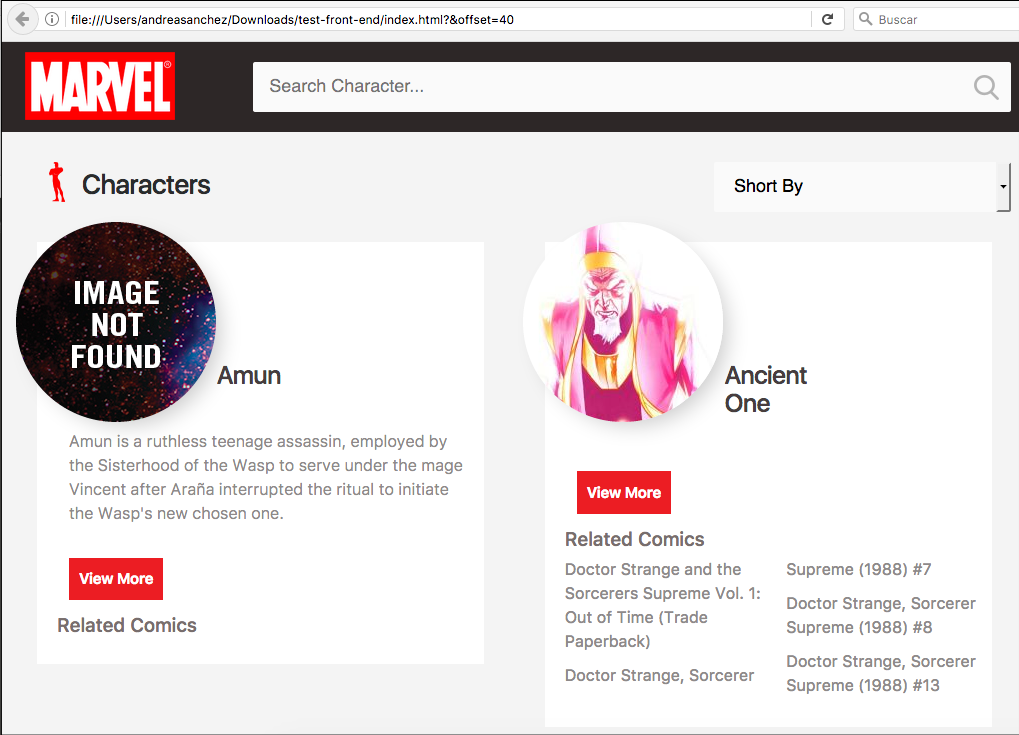
Si se realiza busqueda por personaje carga la informacion del personaje solicitado



al borrar el nombre del personaje y dar clic en la lupa cargara los personajes nuevamente.

* **Paginación**

Se implemento para que pagine hasta la opcion de 50 resultados de la consulta.

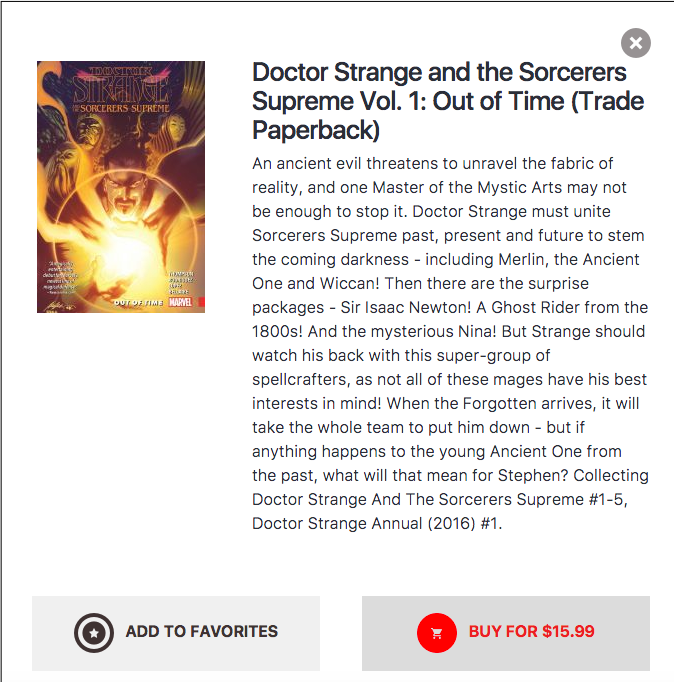


* **Carga Comic Modal:**

En esta opcion se consumio la ruta:

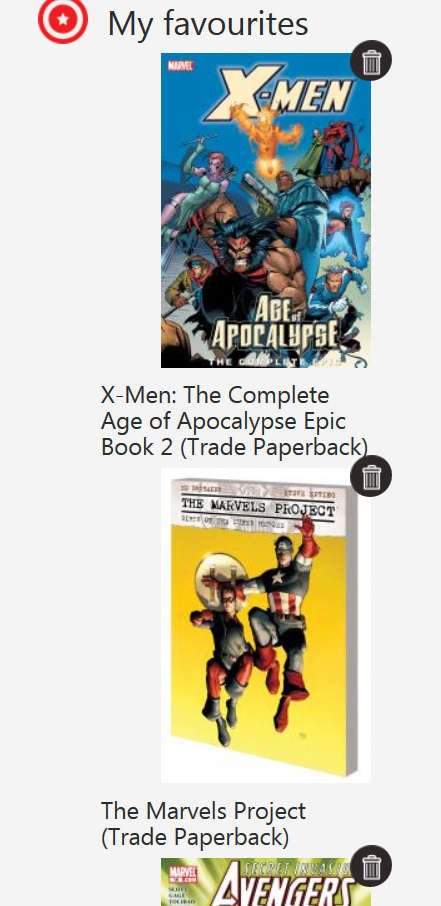


donde se trae la informacion del primer comic que encuentra.



* **Carga de Favoritos**

Se realizó cargando 3 por defecto, luego se modificó implementando localstorage, está pendiente validaciones.

****

**4. Code Refactoring**

* **Las malas prácticas de programación que en su criterio son evidenciadas en el código**

- código que no se usa comentariado

- sobre carga de métodos

* **Cómo su refactorización supera las malas prácticas**

- Separación de las clases en métodos

- Colocar comentarios en las funciones principales

- Crear una clase que envié el mensaje

**5. Preguntas**

**1. ¿En qué consiste el principio de responsabilidad única ? ¿Cual es su propósito?**

Cada método o clase creada debe tener una función específica, tener un código limpio de tal manera que sea más fácil su mantenimiento, con el propósito de tener una mayor escalabilidad y que sea entendible por otros desarrolladores.

**2. ¿Qué características tiene según su opinión “buen” código o código limpio?**

- Identación, comentarios indicando la funcionalidad de cada clase, en procesos colocar un try y catch con el fin de encontrar excepciones más fácilmente, nombre de las variables con camellcase, herencia de clases.