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**
2. **Descripción de la solución**

* **Consulta Personajes**
* **Carga Personajes**
* **Carga Personaje Modal**
* **Carga de Favoritos**
* **Paginación**

**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.