

## Parte 1: Alta de Naves Espaciales

### HTML/CSS:

Crea un formulario que permita ingresar el nombre clave de la nave, subir una foto de la misma y especificar el combustible disponible en la nave y que indica la distancia máxima que puede recorrer.

Diseña una tabla o lista donde se mostrarán las naves ingresadas junto con sus respectivos datos y fotos.

### PHP:

Implementa un script PHP que procese el formulario, guarde la imagen en el servidor y almacene los datos de la nave en un archivo JSON o en MySQL (en caso de usar MySQL, compartir el Dump de la base para la evaluación y en caso de JSON, compartir el JSON final para poder realizar pruebas)

Asegúrate de validar los datos ingresados y gestionar posibles errores de carga de archivos.

### JavaScript:

Opcional: implementa validaciones del lado del cliente antes de enviar el formulario para mejorar la experiencia del usuario.

## Parte 2: Asignación de Naves a Planetas

### HTML/CSS:

Despliega una lista de planetas obtenida de un archivo JSON, mostrando el nombre y la distancia a la que se encuentra de la base.

El archivo JSON tiene esta estructura

```
{
  "planetas": [
    {
      "nombre": "AX9201",
      "distancia": 100000,
      "combustible_disponible": 100000,
    },
    {
      "nombre": "QPO12312",
      "distancia": 150000,
      "combustible_disponible": 100000,
    },
    {
      "nombre": "JK123",
      "distancia": 300000,
      "combustible_disponible": 100000,
    }
  ]
}
```

## Parte 3: Algoritmos

Deberás de implementar un algoritmo que optimice la asignación de naves espaciales a planetas basándose en la distancia y capacidad de las naves, asegurando un uso eficiente de los recursos.

Dada la información de naves y planetas provenientes de los archivos JSON:

**Naves:**

- Nombre
- Distancia máxima que puede recorrer

**Planetas:**

- Nombre
- Distancia respecto a la base
- Combustible disponible para recargar

Crear un algoritmo que determine cuál o cuáles naves deben ser asignadas a los planetas para minimizar la distancia total recorrida por todas las naves y con el mínimo combustible obtenido de los planetas, bajo las siguientes condiciones:

- Una nave solo puede ser asignada a un planeta si la distancia al planeta es menor o igual a la distancia máxima que la nave puede recorrer.
- Cada planeta debe ser visitado al menos una vez.
- Una nave solo puede visitar un planeta por viaje (no es necesario que vuelva a la base antes de viajar a otro planeta).
- La recarga de combustible es opcional pero total, quiere decir que la nave puede elegir si recarga combustible o no; pero tiene que desechar el combustible que trae y reemplazarlo por el que hay en el planeta (son tanques intercambiables)
- Sólo hay un tanque por cada planeta y el que desecha la nave no puede ser reusado por otra.

**Entrada:**

- Lista de naves con sus respectivas distancias máximas
- Lista de planetas con sus respectivas distancias.

**Salida:**

Lista de naves asignadas a los planetas minimizando la distancia total recorrida, si es posible. Si no es posible asignar naves a todos los planetas bajo las restricciones dadas, el algoritmo deberá indicarlo.

**HTML/CSS:**

Aplica estilos que hagan que la aplicación sea visualmente atractiva y temáticamente acorde (usando colores y tipografías espaciales, por ejemplo).

Asegúrate de que la UI sea clara y fácil de entender para el usuario.

En la parte de comparación de distancias, asegúrate de realizar las operaciones de una manera eficiente para no afectar el rendimiento de la aplicación.

Considera la usabilidad al mostrar los datos, asegurándote de que la información sea fácil de leer y entender.