**Semana 1 — Diseño, estructura y base del juego**

**Objetivo:** Establecer la estructura general del proyecto, definir los endpoints del Back-End y preparar la lógica inicial de generación del juego.

**Back-End**

**Tareas:**

1. **Definir la arquitectura del proyecto PHP**
   * Crear la estructura de carpetas (backend/, frontend/, data/).
   * Preparar los archivos base: StartGame.php, SaveScore.php, GetScores.php, ScoreFile.php.
   * Configurar el archivo scores.json inicial (vacío).
   * *Responsable:* Backend 1
   * *Criterio de aceptación:* La estructura está lista y accesible en el repositorio.
2. **Implementar la clase ScoreFile.php**
   * Métodos: read() y write($data).
   * Control de errores si el archivo no existe.
   * *Responsable:* Backend 2
   * *Criterio de aceptación:* Se puede leer y escribir correctamente en scores.json.
3. **Diseñar la lógica de generación de flota (pseudocódigo y test inicial)**
   * Definir tamaños de barcos, posiciones válidas, control de solapamientos.
   * Preparar la estructura JSON de salida.
   * *Responsable:* Backend 1
   * *Criterio de aceptación:* El algoritmo genera una flota válida en estructura JSON.

**Semana 2 — Lógica de juego y comunicación Front/Back**

**Objetivo:** Conectar el cliente con el servidor y gestionar la interacción básica del juego.

**Back-End**

**Tareas:**

1. **Implementar la clase StartGame.php**
   * Métodos: generateFleet(), placeShip(), validatePosition(), toJSON().
   * Comprobación de solapamientos y límites del tablero.
   * *Responsable:* Backend 1
   * *Criterio de aceptación:* Devuelve un JSON con posiciones válidas al llamar por GET.
2. **Implementar GetScores.php**
   * Lectura y devolución de ranking en formato JSON.
   * *Responsable:* Backend 2
   * *Criterio de aceptación:* Devuelve el ranking existente desde scores.json.
3. **Endpoints funcionales**
   * Probar los tres endpoints (StartGame, SaveScore, GetScores) con Postman o test.http.
   * *Responsables:* Backend 1 y 2
   * *Criterio de aceptación:* Las tres rutas devuelven respuestas JSON válidas, con datos coherentes y formato correcto.

**Semana 3 — Puntuaciones, ranking y refinamiento**

**Objetivo:** Guardar puntuaciones, mostrar ranking y realizar pruebas de integración final.

**Back-End**

**Tareas:**

1. **Implementar SaveScore.php**
   * Métodos: receivePOST(), updateScores(), sortScores(), saveToFile().
   * Validación de datos (jugador y puntuación).
   * *Responsable:* Backend 2
   * *Criterio de aceptación:* Se guardan y ordenan correctamente las puntuaciones en scores.json.
2. **Pruebas de integración entre módulos**
   * Comprobar el flujo completo: nueva partida → guardar puntuación → obtener ranking.
   * *Responsables:* Backend 1 y 2
   * *Criterio de aceptación:* Flujo completo funcional y estable.

**Entregable final**

* Juego funcional “Hundir la Flota” accesible desde navegador.
* Backend completamente operativo (PHP) con endpoints:
  + StartGame.php
  + SaveScore.php
  + GetScores.php
* Archivo scores.json con puntuaciones persistentes.
* Documentación técnica básica para el repositorio.