

Roles y Responsabilidades del Equipo

Rol	Nombre (Asignar)	Responsabilidades Clave
Product Owner (PO)	Rodrigo Pinto	Visión del Producto: Definir y priorizar el Backlog. Aceptar o rechazar los incrementos.
Scrum Master (SM)	Rodrigo	Proceso: Guiar al equipo en la metodología SCRUM. Eliminar impedimentos. Facilitar los eventos.
Desarrollador A (Core Logic)	[Nombre del Dev A] Tomas	Validación y Reglas: Responsable de la estructura de datos del tablero y la lógica de validación de movimientos (HU5, HU6, HU10).
Desarrollador B (Flujo y Estado)	[Nombre del Dev B]	Flujo y Actualizaciones: Responsable del manejo del estado del juego, la ejecución del movimiento, la gestión del turno y las condiciones de finalización (HU2, HU7, HU8, HU9).
Desarrollador C (UI y Utilidades)	[Nombre del Dev C]	Presentación: Responsable del <i>front-end</i> (HTML/CSS), la interactividad básica (selección) y los mensajes de sistema (HU1, HU3, HU4, HU10).

1. Historias de Usuario (Product Backlog)

#	Historia de Usuario (Resumen)	Prioridad	Esfuerzo (Puntos)
HU1	Al iniciar, quiero ver un tablero en forma de estrella vacío .	Alta	
HU2	Quiero que mis fichas aparezcan automáticamente en mi triángulo inicial.	Alta	3
HU3	Quiero que el sistema muestre en pantalla a quién le corresponde el turno .	Alta	1
HU4	Quiero poder seleccionar una ficha con el mouse.	Alta	1
HU5	Quiero poder mover una de mis fichas a una casilla adyacente vacía .	Alta	3
HU6	Quiero que el sistema bloquee cualquier intento de movimiento inválido .	Alta	2
HU7	Quiero que el sistema actualice la posición de mis fichas en el tablero después de cada jugada.	Alta	1
HU8	Quiero que el sistema muestre un mensaje cuando gane (todas las fichas en la meta contraria).	Media	5

HU9	Quiero tener la opción de reiniciar el juego en cualquier momento.	Media	3
HU10	Quiero que el sistema detecte si no hay más movimientos posibles y muestre un mensaje de empate.	Baja	5

2. Planificación de Sprints con Tareas de Ingeniería

Sprint 1: Configuración e Interacción Básica (Total: 8 Puntos)

- **Meta:** Tablero visualizado, fichas colocadas y selección funcional.

HU	Tareas de Ingeniería (Implementación JS/HTML/CSS)	Responsable
HU1	JS: Diseñar el objeto TABLERO_DATA (Grafo/Diccionario de 121 casillas) para el modelo de datos.	Dev A
HU1	HTML/CSS: Implementar el <i>markup</i> base (index.html) y los estilos para la forma de estrella.	Dev C
HU2	JS: Llenar TABLERO_DATA con las 20 fichas iniciales (color/posición).	Dev B
HU2	JS/DOM: Implementar dibujarFichasIniciales() y asociar las fichas al modelo de datos.	Dev B
HU3	JS: Inicializar let turnoActual = 'J1' . Implementar actualizarTurnoDisplay() en el HTML.	Dev C
HU4	JS: Asignar Event Listeners (click) a las fichas. Lógica de seleccionarFicha() y aplicar la clase CSS .resaltada .	Dev C

Sprint 2: Movimiento y Reglas Core (Total: 8 Puntos)

- **Meta:** Implementar el movimiento de paso simple y todas las validaciones necesarias para un juego jugable.

HU	Tareas de Ingeniería (Implementación JS)	Responsable
HU5	Implementar la función <code>Validar_Movimiento_Adjacente(origen, destino)</code> (Verificar vecino y casilla libre).	Dev A
HU6	Implementar la lógica de bloqueo : verificar que la casilla de destino no esté ocupada (HU6a, validación dentro de HU5).	Dev A
HU6	Implementar la lógica de bloqueo : validar que la ficha seleccionada pertenezca al jugador del <code>turnoActual</code> (HU3c/HU6c).	Dev B
HU6	Implementar la lógica de advertencia para movimientos inválidos (HU6b).	Dev C
HU7	Implementar la función principal <code>Mover_Ficha(origen, destino)</code> para actualizar <code>TABLERO_DATA</code> y el DOM.	Dev B
HU7	Implementar <code>Pasar_Turno()</code> después de un movimiento válido y actualizar el <i>display</i> (HU3b).	Dev B

Sprint 3: Finalización y Utilidades (Total: 13 Puntos)

- **Meta:** Implementar las condiciones de victoria, empate y la funcionalidad de reinicio.

HU	Tareas de Ingeniería (Implementación JS/UI)	Responsable
HU8	Desarrollar Verificar_Victoria() : Contar 10 fichas en el triángulo opuesto (Meta).	Dev B
HU8	Implementar Mostrar_Mensaje_Ganador() y deshabilitar movimientos.	Dev C
HU9	Implementar la función Reiniciar_Partida() (resetear TABLERO_DATA y turno).	Dev B
HU9	HTML/CSS/JS: Implementar el botón de reinicio y enlazarlo a la función.	Dev C
HU10	Implementar Buscar_Movimientos_Validos(jugador) : Itera sobre todas las fichas del jugador y comprueba posibles movimientos adyacentes.	Dev A
HU10	Implementar la función Verificar_Empate() : Llamar a Buscar_Movimientos_Validos para ambos y si ambos retornan 0, declarar empate.	Dev A
HU10	Implementar Mostrar_Mensaje_Empate() y deshabilitar movimientos.	Dev C

Listado Consolidado de Todas las Tasks

Sprint	HU	Tarea de Ingeniería	Responsable
1	HU1	JS: Diseñar el objeto TABLERO_DATA (Grafo/Diccionario de 121 casillas).	Dev A
1	HU1	HTML/CSS: Implementar el <i>markup</i> base y los estilos para la forma de estrella.	Dev C
1	HU2	JS: Llenar TABLERO_DATA con las 20 fichas iniciales (color/posición).	Dev B
1	HU2	JS/DOM: Implementar dibujarFichasIniciales() y asociar las fichas al modelo de datos.	Dev B
1	HU3	JS: Inicializar <code>let turnoActual = 'J1'</code> . Implementar <code>actualizarTurnoDisplay()</code> .	Dev C
1	HU4	JS: Asignar Event Listeners a las fichas. Lógica de <code>seleccionarFicha()</code> y aplicar clase CSS <code>.resaltada</code> .	Dev C
2	HU5	Implementar la función Validar_Movimiento_Adyacente(origen, destino) .	Dev A
2	HU6	Implementar la lógica de bloqueo : verificar que la casilla de destino no esté ocupada.	Dev A

2	HU6	Implementar la lógica de bloqueo : validar que la ficha seleccionada pertenezca al jugador del <code>turnoActual</code> .	Dev B
2	HU6	Implementar la lógica de advertencia para movimientos inválidos.	Dev C
2	HU7	Implementar la función principal Mover_Ficha(origen, destino) para actualizar <code>TABLERO_DATA</code> y el DOM.	Dev B
2	HU7	Implementar Pasar_Turno() después de un movimiento válido y actualizar el <i>display</i> (HU3b).	Dev B
3	HU8	Desarrollar Verificar_Victoria() : Contar 10 fichas en el triángulo opuesto (Meta).	Dev B
3	HU8	Implementar Mostrar_Mensaje_Ganador() y deshabilitar movimientos.	Dev C
3	HU9	Implementar la función Reiniciar_Partida() (resetear <code>TABLERO_DATA</code> y turno).	Dev B
3	HU9	HTML/CSS/JS : Implementar el botón de reinicio y enlazarlo a la función.	Dev C
3	HU10	Implementar Buscar_Movimientos_Validos(jugador) para detectar movimientos posibles.	Dev A

3	HU10	Implementar la función Verificar_Empate() : Llamar a Buscar_Movimientos_Validos para ambos jugadores.	Dev A
3	HU10	Implementar Mostrar_Mensaje_Empate() y deshabilitar movimientos.	Dev C