

Objetivo del Proyecto

Desarrollar una versión simplificada del clásico juego **Monopoly** que funcione en el navegador web, empleando únicamente **HTML**, **CSS** y **JavaScript puro**, aplicando conceptos de manipulación del DOM, asincronía (fetch hacia un backend), estilos dinámicos y lógica de programación orientada a objetos. El backend del proyecto puede ser descargado desde el siguiente enlace: <https://github.com/felipebuitragocarmona/ms-monopoly>



 <p>FACULTAD DE INTELIGENCIA ARTIFICIAL E INGENIERÍAS</p>	<h2>Primer Proyecto</h2> <h3>Desarrollo Frontend</h3>	FECHA ELABORACIÓN DEL DOCUMENTO: 21/Agosto/2025 VERSIÓN: 1
---	---	---

El juego permitirá jugar de **2 a 4 jugadores** en un tablero interactivo estilo Monopoly. Toda la información de propiedades (nombre, color, precio, renta, etc.), así como las cartas de **Sorpresa** y **Caja de Comunidad**, provendrá del **backend**.

Los jugadores avanzan en el tablero mediante el lanzamiento de dos dados virtuales. El sistema soporta compra/venta de propiedades, construcción de casas y hoteles, hipotecas, préstamos y finalización de partida con envío de resultados para ranking global.

Requisitos NO Funcionales

- La aplicación debe contar con una interfaz responsive, adaptable a dispositivos móviles, tabletas y pantallas de escritorio.**

Requisitos Funcionales

1. Configuración inicial

- Selección del número de jugadores (2, 3 o 4).
 - Cada jugador inicia con **\$1500**.
 - Cada jugador tiene:
 - Una ficha con un color único.
 - El sistema debe solicitar antes de iniciar el juego los datos de “nickname” para identificarlo dentro del juego y el país al que pertenece. (Para esto puede llamar la siguiente api (<http://127.0.0.1:5000/countries>)
-

2. Tablero de juego

- El tablero se renderiza con todas las casillas. Para esto se puede consultar el siguiente endpoint <http://127.0.0.1:5000/board>
- En cada casilla de propiedad se muestra:
 - Nombre** de la propiedad
 - Color** característico en la parte superior.
 - Estado encima de la casilla:
 - "Disponible" en blanco si está libre.
 - Color de ficha si pertenece a un jugador.
 - Número de casas u hotel en caso de existir.

 <p>FACULTAD DE INTELIGENCIA ARTIFICIAL E INGENIERÍAS</p>	<h2>Primer Proyecto</h2> <h3>Desarrollo Frontend</h3>	FECHA ELABORACIÓN DEL DOCUMENTO: 21/Agosto/2025 VERSIÓN: 1
---	---	---

3. Lanzamiento de dados

- Botón para lanzar dos dados.
- Se generan valores aleatorios entre **1 y 6** o puede existir una caja de texto para ingresar **los valores arbitrarios que deseé el usuario** (esto es con el fin de hacer pruebas rápidas en la sustentación). La ficha del jugador actual se mueve según la suma.

4. Acciones al caer en una casilla

- **Propiedad libre** → Opción de comprarla (descontando dinero).
- **Propiedad de otro jugador** → Pago de renta según nivel de construcción.
- **Propiedad hipotecada** → No genera renta.
- **Propiedad del propio jugador** → Posibilidad de construir casas/hoteles si cumple condiciones.
- **Sorpresa o Caja de Comunidad** → Se obtiene carta aleatoria que suma o resta dinero.
- **Impuestos** → Se descuenta dinero según lo indique la casilla.
- **Cárcel** → El jugador pierde turnos o paga para salir según reglas de Monopoly.

5. Construcción de casas y hoteles

- Para comprar una **casa** en una propiedad:
 - El jugador debe poseer **todas las propiedades del mismo color**.
- Cada propiedad puede tener **máximo 4 casas**.
- Al llegar a 4 casas, el siguiente paso es **comprar un hotel**, que reemplaza a las 4 casas.
- Los valores de renta con casas y hoteles están definidos en el JSON del backend.
- El valor de las casas es de 100 y del hotel 250

6. Hipotecas y préstamos

- Un jugador puede hipotecar una propiedad si necesita liquidez.
 - El banco el valor definido en el campo “mortgage”.
 - Mientras esté hipotecada, no se puede cobrar renta.

 <p>FACULTAD DE INTELIGENCIA ARTIFICIAL E INGENIERÍAS</p>	<h2>Primer Proyecto</h2> <h3>Desarrollo Frontend</h3>	FECHA ELABORACIÓN DEL DOCUMENTO: 21/Agosto/2025 VERSIÓN: 1
---	---	---

- Para deshipotecar, se paga el valor recibido más un 10% de interés, esto se puede hacer cuando le toque el turno al jugador en cuestión
 - Si al final de la partida no devuelve el préstamo esa propiedad no será tenida en cuenta en el puntaje final
-

7. Visualización de información

Cada jugador tendrá un panel que muestra:

- Iniciales y bandera del país.
 - Color de su ficha.
 - Dinero disponible.
 - Lista de propiedades adquiridas.
 - Estado de hipotecas y préstamos activos.
-

8. Finalización del juego

El juego puede finalizar de manera manual pulsando el botón "**Finalizar Juego**".

- Se calcula el **valor total del patrimonio** de cada jugador:
 - Sumar el dinero disponible + Valor de propiedades (considerando casas (100) y hoteles (200)).
 - Restar el valor de las propiedades hipotecadas.
- El jugador con **mayor patrimonio neto** es el ganador.
- Una vez terminada la partida los puntos son enviados al backend a la ruta <http://127.0.0.1/score-recorder> con el método POST según el siguiente body:

```
{
  "nick_name": "fbc",
  "score":2500,
  "country_code":"co"
}
```
- Dentro del sistema debe existir una opción para mostrar el ranking de los jugadores, esta es consultada al api <http://127.0.0.1/ranking> con el método GET. En el ranking deben aparecer las banderas de los países a los que pertenecen los jugadores. Para esto puede revisar la siguiente documentación: <https://flagsapi.com/#sizes>

 <p>FACULTAD DE INTELIGENCIA ARTIFICIAL E INGENIERÍAS</p>	<p>Primer Proyecto Desarrollo Frontend</p>	<p>FECHA ELABORACIÓN DEL DOCUMENTO: 21/Agosto/2025</p>
		<p>VERSIÓN: 1</p>

Aprendizajes Esperados

- Construcción de un tablero dinámico con **HTML + CSS**.
- Manipulación avanzada del **DOM con JavaScript**.
- Trabajo con **objetos y arrays** para representar propiedades y jugadores.
- Consumo de datos desde un **backend simulado en JSON**.
- Aplicación de lógica condicional y ciclos en un entorno real (reglas del Monopoly).
- Control de estado del juego y turnos entre jugadores.