

# Proyecto Final

---

Curso	Programación Avanzada en Web
Código	SC-701
Profesor	Luis Andrés Rojas Matey

## Introducción

El juego de mesa **Connect4** se podría decir que es una evolución del **Tic-tac-toe** (conocido popularmente como "Gato"). Este es para 2 jugadores y consiste en "conectar" cuatro fichas de un mismo color en una cuadrícula.

## Objetivo

Aplicar los conocimientos adquiridos para desarrollar una aplicación web del juego **Connect4**.

## Especificaciones funcionales

Al ejecutar la aplicación web, se le deben proveer cuatro opciones al usuario:

1. Crear un jugador nuevo.
2. Crear una partida nueva.
3. Cargar una partida existente.
4. Mostrar el escalafón de jugadores.

### Crear un jugador nuevo

Esta sección permite crear un jugador a partir de dos datos que se deben solicitar al usuario:

- **Identificación**. Es un número entero positivo único (por ejemplo, un número de cédula).
- **Nombre**. Es un *string* con el nombre del jugador.

Cada jugador tendrá asociado su propio **Marcador**, el cual será la suma de los resultados de las partidas, asignados de la siguiente forma:

- Una unidad positiva (+1) por cada partida ganada.

- Una unidad negativa (-1) por cada partida perdida.
- Una unidad nula (0) por cada partida empatada.

Este **Marcador** puede ser negativo en caso que el jugador haya perdido más partidas de las que ha ganado.

## Crear una partida nueva

Cuando el usuario selecciona esta opción, se le solicitará que escoja ambos jugadores (de los creados previamente), primero el **Jugador #1** y luego el **Jugador #2**. No es permitido escoger el mismo jugador (un jugador no puede jugar contra sí mismo).

Una vez escogido los jugadores, se debe desplegar una cuadrícula de 7 columnas y 6 filas (es decir, con 42 celdas) donde cada columna va nombrada con letras mayúsculas en orden de izquierda a derecha empezando por la letra **A** (es decir: **A, B, C, D, E, F, G**). Por ejemplo:

A	B	C	D	E	F	G
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_

El área de juego (la cuadrícula) deberá aparecer con las celdas vacías y se debe indicar que le toca jugar al **Jugador #1** (mostrando su nombre). El **Jugador #1** empieza el juego escogiendo una columna por su letra, por lo que consecuentemente su ficha "caerá" en la celda disponible (vacía) inferior de dicha columna. Luego de esto, se debe indicar que el turno para jugar es para el **Jugador #2** (por su nombre), el cual igualmente debe escoger una columna y su ficha debe "caer" en la celda vacía "de más abajo". Un jugador no puede escoger una columna que ya no tenga celdas disponibles (vacías).

El juego finaliza cuando se da alguno de los dos escenarios siguientes:

1. Una victoria: cuando un jugador logra poner cuatro de sus fichas seguidas en línea en cualquier dirección.
2. Un empate: cuando ambos jugadores hayan puesto la totalidad de sus fichas (21 fichas por cada jugador).

En cualquier caso (cuando se finaliza la partida), la página debe indicar el resultado:

- Victoria del **Jugador #1**.
- Victoria del **Jugador #2**.
- Empate.

Las fichas deben tener forma redonda (en celdas cuadradas -o rectangulares- del área de juego -cuadrícula-).

En cualquier momento de la partida se debe tener la opción de "reiniciar", esto es, crear una nueva partida con los mismos jugadores de la partida actual (no sobreescribirla).

## Cargar una partida existente

Cuando el usuario escoge esta opción, se debe mostrar la lista de todas de las partidas creadas, sin importar si están finalizadas o no (aunque sí se debe visualizar su estado para facilitar la selección). Esta lista debe contener las fechas y horas de las partidas creadas, así como los nombres de ambos jugadores, y estar ordenada en forma descendente por fecha y hora de creación. El usuario entonces puede escoger alguna de las partidas. Si la que escoge aún no ha finalizado, entonces se puede seguir jugando tal como una partida nueva, excepto que los nombres de los jugadores no se deben solicitar (porque la partida ya los tiene) y la cuadrícula debe aparecer con las fichas en el área de juego (cuadrícula) correspondientes a dicha partida.

## Mostrar escalafón de jugadores

Esta es una tabla que muestra la lista de jugadores (**Identificación** y **Nombre**), ordenados por su **Marcador** de forma descendente. Además debe incluir la cantidad de partidas ganadas, perdidas y empatadas. Por ejemplo:

Identificación	Nombre	Marcador		Ganadas	Perdidas	Empatadas
123456789	Ana	10		15	5	3
987654321	Braulio	4		10	6	1
304560765	Carlos	-2		6	8	20

## Ejemplo

Para este ejemplo, las fichas del **Jugador #1** se identifican como "●" y del **Jugador #2** como "○".

El usuario del juego crea 2 jugadores con sus respectivas identificaciones y nombres: Alicia y Bernabé.

Al escoger una nueva partida, se seleccionan ambos jugadores creados anteriormente (Alicia y Bernabé) y la cuadrícula debe desplegarse con todas las celdas vacías.

Se debe indicar que es el turno del **Jugador #1** (con su respectivo nombre: Alicia). Solo se debe mostrar el nombre del jugador, no así su identificación.

Suponiendo que el **Jugador #1** (Alicia) escoge la letra **D**, su ficha "caerá" hasta la fila inferior (porque todas las celdas de dicha columna están vacías). Esto significa que su ficha se posicionará en la fila inferior de dicha columna, así:

A	B	C	D	E	F	G
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	•	_	_	_

Posteriormente, se debe indicar que es el turno del **Jugador #2** (Bernabé) y al escoger este también la columna **D**, su ficha "caerá" hasta la celda vacía inferior ("sobre" la ficha puesta inicialmente por el **Jugador #1**):

A	B	C	D	E	F	G
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	◊	_	_	_
_	_	_	•	_	_	_

La partida continúa con el **Jugador #1** escogiendo la columna **E** y el **Jugador #2** escogiendo la **F**:

A	B	C	D	E	F	G
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	_	◊	_	_	_
_	_	_	•	•	◊	_

La partida sigue así:

- El **Jugador #1** escoge la columna **C**.
- El **Jugador #2** escoge la columna **B**.
- El **Jugador #1** vuelve a escoger la columna **C**.

- El **Jugador #2** escoge la columna **E**.
- El **Jugador #1** vuelve a escoger la columna **C**.
- El **Jugador #2** escoge la columna **C**.
- El **Jugador #1** escoge la columna **A**.
- El **Jugador #2** escoge la columna **D**.

En este punto, el **Jugador #2** ganó la partida (consiguió poner 4 de sus fichas seguidas en diagonal). La posición final de la cuadrícula quedaría así:

A	B	C	D	E	F	G
_	_	_	_	_	_	_
_	_	_	_	_	_	_
_	_	○	_	_	_	_
_	_	●	○	_	_	_
_	_	●	○	○	_	_
●	○	●	●	●	○	_

Aquí, la partida finaliza y ya ningún jugador puede escoger más columnas, indicando que ganó Bernabé.

## Especificaciones técnicas

Esta es una aplicación web, por lo que debe ser posible ejecutarse en un navegador web.

Se puede utilizar cualquier *framework* de arquitectura **SPA** (*Single-Page Application*, como **Mithril** o **Svelte**) o **MPA** (*Multi-Page Application*, como **ASP.NET Core** o **PHP**).

Para la persistencia de los datos, se debe utilizar una base de datos. Esta base de datos puede ser de tipo **SQL** (como **MySQL** o **PostgreSQL**) o **NoSQL** (como **MongoDB** o **Firestore**), debe estar guardada (o ejecutándose) en el "lado del servidor" (incluso un **DBaaS** o *Database as a Service*), y puede ser de tipo "un solo archivo" (como **SQLite** o **DuckDB**) o bien como parte de un sistema de gestión de bases de datos (como **MariaDB** o **SQL Server**). Es posible también utilizar cualquier conector, intermediario, biblioteca, paquete o herramienta para la comunicación con la base de datos (como por ejemplo **Entity Framework**).

Así mismo, es posible utilizar cualquier *framework* de CSS disponible, o bien, se pueden usar estilos propios.

Para la página web, se puede utilizar cualquier elemento HTML para la representación de la cuadrícula (como **<table>**). También, es posible utilizar cualquier elemento para la escogencia de la columna por parte del jugador, pero esta escogencia debe ser con clics; es decir, los usuarios solo deben escribir las identificaciones y nombres de los jugadores, todo lo demás debe hacerse con solo clics (por ejemplo, con **<button>** o **<a>**, etc.).

## Entregables

Debido a que este proyecto se debe hacer según los grupos establecidos previamente, el único entregable (es decir, lo único que se debe subir al **Campus Virtual**) es el vínculo (*link*) al repositorio en línea de **Git** (como por ejemplo, a **GitHub** o **GitLab**). Este vínculo debe ser subido por solo uno de los miembros del grupo. Este repositorio puede ser privado pero deberá ser público al momento que les llegue el turno de la exposición del proyecto (ya sea en la semana 14 o 15), para que el profesor pueda tener acceso al mismo.

En el repositorio debe estar lo siguiente:

- Todo el código fuente del proyecto, excepto los archivos compilados.
- Un archivo **README.md** (escrito en **Markdown**) en la raíz del proyecto, que contenga lo siguiente:
  - Los nombres y carnés de los integrantes del grupo. Estos serán los únicos que serán tomados en cuenta para la calificación.
  - El nombre de usuario y correo de **Git** de cada integrante.
  - Una lista de los *frameworks* utilizados, así como herramientas (por ejemplo, el motor de base de datos).
  - El tipo de aplicación (**SPA** o **MPA**).
  - La arquitectura utilizada (por ejemplo, **MVC**).
  - Un diagrama (puede utilizar, por ejemplo, **Mermaid**) de la definición de la base de datos (por ejemplo, una base de datos **SQL** relacional tendría tablas, columnas, tipos de campos, llaves, referencias, etc.; mientras que una **NoSQL** documental tendría colecciones, documentos, tipos de campos, etc.).
  - Un breve instructivo de instalación (por ejemplo, para proyectos hechos con **NodeJS**, la instalación de paquetes vía **npm install**), compilación o creación (siguiendo con el ejemplo de **NodeJS**, como con el comando **npm build**) y ejecución (continuando con **NodeJS**, el comando para que funcione la aplicación, como **npm run execute** o **node script.js**).
  - Todas las referencias de sitios webs con *snippets* de código utilizados, así como los *prompts* (tanto de entrada como salida) de los agentes de AI, que se hayan utilizado.

## Evaluación

El proyecto será calificado según la rúbrica que se presenta en el programa del curso.