

<b>UTN – FRMDP Mar del Plata</b> <b>TSP - Programación 1</b> <b>Trabajo Práctico - Parte 2</b>  Mayo - Junio 2025	<b>Integrantes del grupo</b>	<b>Nota</b>
---	------------------------------	-------------

## Introducción

Con el propósito principal de integrar todo lo aprendido en la materia Programación 1 hemos planteado la siguiente problemática:

- Codificar un sistema de Logueo que gestione la estructura **Jugador**.
- Administrar un archivo de Partidas, Jugadores y Partidas Por Jugadores.
- Lógica de juego (parte 1).

## Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo (se formarán grupos de 3 o 4 estudiantes) a partir del desarrollo de un proyecto de software. Para que este tipo de proyectos sea más exitoso, deben llevarse a cabo desde un enfoque que facilite alcanzar los Objetivos de Aprendizaje propuestos.

Una de las ideas centrales es desarrollar competencias profesionales y preparar al futuro programador para el mundo laboral y el trabajo en equipo.

En un ambiente de aprendizaje colaborativo, los estudiantes:

- Construyen conocimiento y en lugar de recibirlos en forma pasiva;
- Se involucran y se comprometen directamente con el descubrimiento de nuevo conocimiento;
- Se exponen a puntos de vista alternativos e ideas contrapuestas, de forma tal que pueden sacar sus propias conclusiones y así transformar conocimientos y experiencias previas y de esta manera comprender con mayor profundidad;
- Transfieren conocimientos y habilidades a nuevas situaciones o circunstancias;
- Se responsabilizan y apropian tanto de su aprendizaje continuo de contenidos curriculares, como del desarrollo propio de competencias;
- Los estudiantes colaboran para el aprendizaje del grupo y el grupo colabora en el aprendizaje individual de estos.

## Objetivos

De aprendizaje:

- Incorporar Arreglos, Matrices, Modularización, Estructuras de Datos y Archivos Binarios.
- Trabajar en forma colaborativa.

Metodológicos:

- Ser capaces de trabajar en un proyecto complejo, aplicando técnicas de desarrollo de software.
- Lograr integrar contenidos de otras asignaturas.
- El grupo deberá ir mostrando el avance sobre el trabajo en clase.

## Pautas Generales

### La función principal - Main() y menús: [ 15 puntos ]

#### IMPORTANTE: LA NAVEGABILIDAD DEL PROGRAMA.

El sistema deberá contar con una presentación amigable con el usuario, construir menús de acceso a las diferentes estructuras y funcionalidades del sistema, **y de manera directa o indirecta, permitir probar todas las funciones desarrolladas.**

El desarrollo del sistema deberá ser ordenado, identificando con comentarios cada una de las funciones realizadas, explicando brevemente lo que realizan. Se tendrá en cuenta, al momento de evaluar, la prolijidad del código y la organización de los módulos. Se recomienda agrupar los mismos por funcionalidad.

### Funcionalidad básicas: [60 puntos]

Módulo	Funcionalidad
<b>Gestión de la información durante todo el ciclo de vida de la aplicación</b>	El sistema debe trabajar en <b>memoria principal (en arreglos)</b> consumiendo las estructuras generadas para tal fin. Al inicio del sistema se consultarán los archivos binarios ( <b>archivo hacia arreglo</b> ) y se actualizará esa información ( <b>arreglo hacia archivo</b> ) al finalizar la ejecución del programa.
<b>Inicio de sesión</b>	Ingresar con email y contraseña.
	El usuario podrá cerrar sesión (terminar el programa) en cualquier momento, menos durante la ejecución de una partida.
	El usuario podrá cerrar definitivamente su cuenta. Eliminar jugador.

<b>Registro de usuario</b>	<p>Registrarse validando email, contraseña y DNI y username.</p> <ul style="list-style-type: none"> <li>- El email debe contener un @ y un .com. Además se debe validar que el email no este registrado previamente por otro jugador.</li> <li>- La contraseña debe contener por lo menos 1 mayúscula y 1 minúscula.</li> <li>- El username debe ser único por jugador.</li> <li>- Al momento en que finaliza el proceso de registro, se lo debe redirigir a la pantalla de login.</li> </ul>
<b>Perfil de jugador</b>	<p>Modificar información personal:</p> <ul style="list-style-type: none"> <li>- El jugador podrá modificar su username.</li> </ul>
	<p>El jugador va a poder ver sus estadísticas:</p> <ul style="list-style-type: none"> <li>- Cantidad y porcentaje de partidas jugadas, ganadas, empate y derrotas.</li> <li>- Ver ranking de jugadores (ordenados por puntos totales)</li> </ul>
<b>Partidas</b>	<p>Crear nueva partida.</p>
	<p>Elegir jugador contra jugador o contra CPU.</p>
	<p>Seleccionar dificultad al jugar contra CPU:</p> <ul style="list-style-type: none"> <li>- Mínimo 2 niveles de dificultad.</li> </ul>
	<p>Jugar partida con turnos alternados (Modo 2 jugadores)</p> <ul style="list-style-type: none"> <li>- Si elige este modo, primero debe pedir las credenciales del jugador 2 (proceso de login) y la información de la partida debe registrarse en ambos jugadores.</li> </ul>

	Visualizar tablero y actualizar estado de partida.
	Finalizar partida y registrar resultado.

## Persistencia de la información y estructuras: [25 puntos]

<p>El sistema almacenará toda la información en los siguientes archivos binarios:</p> <p><b>jugadores.dat:</b></p> <p>Contendrá los datos personales del jugador.</p>	<p>Se usará la siguiente estructura:</p> <pre>typedef struct {     int idJugador;     char nombre[30];     char apellido[30];     char email[50];     char username[30];     char password[20];     char dni[10];     int ptsTotales;     int eliminado; // 0 = activo, 1 = eliminado } stJugador;</pre>
<p><b>partidas.dat:</b></p> <p>Guardará la información de cada partida.</p>	<pre>typedef struct {     int idPartida;     int dificultad;     int esContraCpu; } stPartida;</pre>

<p><b>partidaXjugador.dat:</b></p> <p>Relación entre jugadores y partidas, resultado y puntaje.</p>	<pre>typedef struct {     int idPartidaJugador;     int idPartida;     int idJugador;     int resultado;    // 0 = pierde, 1 = gana, 2 = empate     int puntosJugador; // 3 = gana, 1 = empate y 0 = derrota } stPartidaXJugador;</pre>
---	---