

## 2° Parcial Lab II - 2°A - 1°Cuat. 2023

### Requerimientos

Nuestro cliente nos ha pedido que desarrollemos un simulador de juegos de mesa. Para lo cual quiere, por esta vez, que desarrollemos uno de los siguientes juegos:

- Truco.
- La generala.
- Uno.
- La escoba de 15.
- Chinchón.

La aplicación deberá simular partidas en tiempo real. Es decir, el usuario que corra la aplicación deberá poder crear salas de juego donde se jugarán las partidas de forma simultánea.

Mientras las partidas se están jugando, el usuario de la aplicación deberá poder consultar en cada una de ellas:

- La información de la partida seleccionada.
- Ver las estadísticas del historial de partidas jugadas.
- Mover el formulario por toda la pantalla sin que la partida se vea afectada.
- Cancelar la partida en curso.

Las partidas deberán tener las siguientes características:

- 1) Serán partidas cortas, es decir, no hace falta que la partida sea extensa.  
Por ejemplo:

- Si el desarrollador elige desarrollar *el truco*, la partida deberá simular 4 rondas (manos).
- Si el jugador elige *La Generala*, deberán ser simuladas las primeras 4 tiradas de dados de cada uno de los jugadores (pueden ser 2 o 4).

2) Las partidas podrán crearse en cualquier momento de la ejecución del programa, no hay restricción de cantidad de salas abiertas. Las partidas pueden o no terminar juntas.

Una vez finalizada la partida, deberá notificarse en un nuevo formulario quien fue el ganador de la partida y toda la información de la misma.

Si los desarrolladores lo desean, una vez terminada la funcionalidad que simula las partidas, los desarrolladores podrán hacer el modo donde el usuario pueda jugar contra la máquina. Esto último será opcional.

Una vez desarrollada la aplicación, las pruebas unitarias de la misma deberán cubrir la mayor cantidad de funcionalidades posibles. No se dará por cumplido este ítem si la aplicación presenta mucho código sin testear.

## Recomendación

No centrar el desarrollo del programa en la interfaz gráfica, lo más importante es la calidad del código y sus funcionalidades. Por lo tanto, eviten invertir tiempo excesivo en la parte gráfica / visual.

## Formato de entrega y documentación

El examen será entregado el día **miércoles 21 de Junio**.

Por medio de un formulario de Google se pedirá que registren el repositorio de **github** dónde se encontrará ubicado su código.

El nombre del repositorio será **Apellido.Nombre.TP2**.

El repositorio **debe** configurarse cómo **privado** y agregar cómo colaboradores los usuarios de los docentes:

- ★ Maximiliano Neiner: **maxineinerutn**
- ★ Facundo Rocha: **facc15**

Esto último se realiza con el fin de que su código no sea 'copiado' sin su consentimiento.

Recordar que, códigos parecidos / muy parecidos / idénticos implica la reprobación de ambos parciales.

El repositorio debe tener informados varios **commits** (desde su creación hasta la fecha de entrega).

### **IMPORTANTE**

*La creación del repositorio privado y el posterior agregado de los colaboradores deberá realizarse de inmediato.*

## Documentación

Se deberá completar la información indicada en el archivo **README.md** que se encuentra en el repositorio.

Para trabajar con este archivo se deberá utilizar el lenguaje de marcado [Markdown](#).

Secciones e información a documentar

- **Título:** Ponerle un nombre de fantasía a la aplicación.
- **Sobre mí:** Presentarse brevemente. Contar su experiencia programando y lo que significó para vos este trabajo.
- **Resumen:** Explicar qué hace la aplicación y cómo se usa a grandes rasgos.
- **Diagrama de clases:** Pegar una foto del diagrama de clases correspondiente a la lógica de negocio. Se debe construir con la herramienta de Visual Studio y deberá estar actualizado a la última versión entregada de la solución.
- **Justificación técnica:** explicar tema a tema dónde se fue aplicando en el código **y por qué se decidió implementarlo de esa forma**. Toda decisión tiene que estar argumentada con razones técnicas que giren alrededor de los pilares de la programación orientada a objetos y cuestiones de mantenibilidad, código limpio, flexibilidad al cambio, experiencia de usuario, accesibilidad, uso seguro, rendimiento y eficiencia.
  - Suma identificar las partes del código dónde se implemente cada tema.
  - Dar pros y contras, si los tienen en mente.
  - El objetivo es que demuestren que saben lo que hacen y que tomaron decisiones con criterio y no mecanizadas.
  - Si se utilizó alguna biblioteca externa también se deberá justificar la elección.

**No se corregirá ni revisará ningún parcial que no presente esta documentación completa con el formato pedido previamente.**

## Condiciones de evaluación

Se evaluarán las siguientes pautas:

- Respetar TODAS las reglas de estilo de la cátedra y buenas prácticas indicadas en clase. Se corregirán todas las pantallas (visualización) y calidad de código según las exigencias de la cursada.
- Compilar sin errores ni advertencias (sí se admiten sugerencias del IDE/warnings).
- Debe resolver TODAS las necesidades del cliente (planteadas en la consigna y en el requerimiento) y no tener errores en el tiempo de ejecución.
- Separar de forma física (distintos proyectos) la capa de presentación (interfaz de usuario) de la lógica de negocio.
- No podrá haber información previa hardcodeada como en el primer TP/Parcial:  
Todos los datos de índole estadísticos, jugadores, etc, deberán ser recuperados de una fuente de datos ajena a la aplicación.
- Las clases y sus miembros deberán estar correctamente documentados con la herramienta de documentación xml o ser lo suficientemente autodescriptivos (esto último queda a criterio del docente corrector).
- Validar todos los ingresos de datos (cuando corresponda) mostrando mensajes claros para el usuario cuando un dato sea inválido.
- Deberán implementarse los siguientes temas de forma obligatoria:
  - SQL
  - Manejo de excepciones
  - Unit Testing
  - Generics
  - Serialización
  - Escritura de archivos
  - **Interfaces**
  - **Delegados**
  - **Task**
  - **Eventos**
- Los temas en **negrita** deben estar correctamente implementados para lograr la promoción. Quedará a criterio del docente corrector la aprobación del parcial en caso de no haber implementado un tema o estar mal implementado.
- No será obligatorio el uso del pilar Herencia, pero puede ser usado.
- Todas las entidades deberán estar correctamente encapsuladas exponiendo sólo sus operaciones y características esenciales, protegiendo el acceso y modificación libre de datos, y ocultando los detalles de la implementación.
- **No podrá tener nota de promoción** aquel alumno/a que no implemente correctamente delegados, interfaces, programación multihilo y eventos.

## Calificación

**Solo si todos los puntos de evaluación antes mencionados se cumplen**, se procederá a definir la nota por tema evaluando un uso correcto y bien justificado de cada uno de los temas anteriormente mencionados.

La nota de este parcial complementará la nota de la evaluación que se llevará a cabo de forma presencial en el día pactado por su comisión.

### **A tener en cuenta**

---

Reglas de estilo, buenas prácticas y calidad de documentación.

---

Algoritmos y resolución de problemas. Reutilización de código. Elección de tipos de datos. Trabajo con cadenas de texto, fechas, enumerados, etc.

---

Uso correcto y justificado de los temas utilizados en el primer parcial que también sean utilizados en este.

---

Abstracción. Diseño de clases y objetos (diagrama de clases, atributos, métodos, constructores).

---

Creatividad en el diseño de formularios. Uso correcto de controles y eventos. Buena experiencia de usuario.

---

Uso correcto y justificado de Interfaces

---

Encapsulamiento y uso correcto y justificado de modificadores de acceso.

---

Uso correcto y justificado de Serialización y archivos.

---

Uso correcto y justificado de programación Multihilos.

---

Uso correcto de pruebas unitarias y cobertura del 70% de la aplicación.

---

Solución robusta, con validaciones y control de errores.

---

Uso correcto y justificado de Delegados.

---

Uso correcto y justificado de Eventos

---

Defensa oral / modificación en vivo del programa ( agregar nueva funcionalidad o modificar una existente)

## Consultas sobre el parcial

Se podrán responder consultas sobre el parcial durante las clases (NO ocupará toda la clase). De existir necesidad de hacer alguna consulta en otro momento de la semana, no se garantizará una respuesta veloz ya que la misma quedará sujeta a disponibilidad del docente y/o sus ayudantes.

**Todas las preguntas deberán realizarse por Classroom como único medio y sin excepción por la clase [SEGUNDO PARCIAL](#) (en los Comentarios de la clase, NO cómo comentarios privados).**