TFG

PROYECTO FIN DE CICLO (TFG)

NOMBRE: FreeDartsGame

AUTOR: JOSÉ ANTONIO SERRANO PIÑAR

TUTOR: ANDRÉS MAROTO

I. E. S. VALLE-INCLÁN

Proyecto

Introducción

La aplicación "*FreeDartsGame*" es una aplicación móvil multiplataforma desarrollada en .NET MAUI que simula y gestiona partidas de dardos. La aplicación permite a los usuarios jugar diferentes modalidades de dardos, registrar estadísticas de juego y gestionar perfiles de usuario. Representa una solución completa para la gestión de partidas de dardos, combinando funcionalidades de juego robustas con un sistema de estadísticas avanzado; demostrando así un equilibrio exitoso entre simplicidad de uso y funcionalidades avanzadas, y posicionándose como una herramienta valiosa tanto para jugadores casuales como para entusiastas serios de los dardos.

Esta aplicación ha sido mi idea de TFG debido a un mero propósito personal de crear algo relacionado con una de mis aficiones favoritas a la que me dedico de manera amateur, decantándome por este proyecto básico con el que considero que personas que se interesen por los dardos pueden introducirse, ofreciendo modos de juego curiosos y divertidos aparte de los clásicos de 501 y Cricket y pudiendo registrar y consultar sus resultados obtenidos con las partidas que jueguen.

Contenido

Requisitos

REQUISITOS FUNCIONALES

1. Gestión de Usuarios

- Registrar usuarios con username y email
- Validar que no existan usuarios duplicados (username/email)
- Iniciar sesión con username
- Cerrar sesión
- Eliminar usuarios (con confirmación)
- Asociar usuarios registrados a jugadores en partidas

2. Gestión de Partidas

- Crear partidas de entre 2 y 4 jugadores
- Seleccionar usuarios registrados como jugadores
- Agregar jugadores invitados (sin registro)
- Eliminar jugadores de una partida (mínimo 2)
- Iniciar nueva partida sin salir de la aplicación

3. Modos de Juego

- **Modo X01:**
  - Configurar puntuación inicial (301, 501 o 701)
  - Restar puntos de cada lanzamiento
  - Validar que no se exceda el puntaje restante
  - Ganador: primer jugador en llegar a 0
  - Mostrar combinaciones de checkout posibles (hasta 5)
- **Modo Cricket:**
  - Cerrar números: 15, 16, 17, 18, 19, 20, 25 (Bull)
  - 3 marcas para cerrar un número

- Ganar puntos si el número ya está cerrado
- Manejar overmarks (marcas extras después de cerrar)
- Ganador: primer jugador en cerrar todos los números con mayor puntuación

4. Sistema de Lanzamientos

- Registrar lanzamientos (número 1-20, 25 para Bull exterior, 50 para Bull interior, 0 para fallido)
- Registro de marcas: 1 (simple), 2 (doble), 3 (triple)
- Validar combinaciones válidas:
  - Números 1-20: marcas 1-3
  - Bull (25 o 50): marcas 1-2
  - Fallido (0): marca 1
- Rondas de 3 lanzamientos por jugador
- Deshacer último lanzamiento de la ronda actual
- Cooldown de 5 segundos entre rondas

5. Estadísticas de Jugadores

- Estadísticas generales:
  - Total de lanzamientos
  - Total de puntos
  - Puntuación promedio por lanzamiento
  - Partidas jugadas y ganadas
  - Porcentaje de victorias
  - Mejor puntuación
  - Promedio de turnos por partida
- Estadísticas por modo:
  - Partidas jugadas en X01 y Cricket
  - Victorias en cada modo
- Estadísticas específicas de Cricket:
  - Total de marcas de Cricket
  - Promedio de marcas por ronda (3 lanzamientos)

- Fecha de última partida y de registro

6. Interfaz de Usuario

- Página principal con navegación
- Selección de jugadores antes de iniciar
- Página de juego con:
  - Indicador de jugador actual
  - Puntuación actual de cada jugador
  - Interfaz diferente según el modo (X01/Cricket)
  - Historial de lanzamientos de la ronda
  - Combinaciones de checkout (modo X01)
  - Estado de marcas de Cricket (modo Cricket)
- Página de estadísticas del usuario logueado
- Página de gestión de usuarios (registro, login, eliminación)

7. Persistencia de Datos

- Guardar usuarios en archivo JSON (users.json)
- Guardar estadísticas en archivo JSON (user_statistics.json)
- Cargar datos al iniciar la aplicación
- Guardar automáticamente tras cambios

REQUISITOS NO FUNCIONALES

1. Plataforma y Tecnología

- Framework: .NET MAUI
- Lenguaje: C#
- Plataformas soportadas:
  - Android
  - iOS
  - macOS
  - Windows
- Versión .NET: 8.0

2. Arquitectura

- Patrón MVVM con INotifyPropertyChanged
- Singleton para UserService
- Separación de modelos, servicios y vistas
- Binding bidireccional con XAML

3. Rendimiento

- Actualización de UI en tiempo real (PropertyChanged)
- Cálculo eficiente de combinaciones de checkout (límite de 5)
- Persistencia asíncrona (guardado tras eventos, no síncrono)

4. Usabilidad

- Interfaz intuitiva con pickers para entrada
- Validación en tiempo real de lanzamientos
- Mensajes de error descriptivos
- Confirmación para acciones destructivas (eliminar usuario/jugador)
- Cooldown visual entre rondas (5 segundos)

5. Validación y Control de Errores

- Validación de lanzamientos antes de aceptarlos
- Validación de nombres duplicados
- Validación de límites de jugadores (2-4)
- Manejo de excepciones en carga/guardado de archivos
- Verificación de estado del juego antes de acciones

6. Mantenibilidad

- Código organizado en namespaces
- Modelos con responsabilidades claras
- Logging con System.Diagnostics.Debug
- Comentarios en código crítico

7. Escalabilidad

- Arquitectura extensible para nuevos modos de juego
- Estadísticas separadas por usuario

- Sistema de usuarios que permite múltiples jugadores simultáneos

8. Confiabilidad

- Manejo de archivos corruptos o faltantes (carga con valores por defecto)
- Estado consistente del juego
- Validación de transiciones de estado (rondas, cambio de jugador)

9. Seguridad

- Validación de entrada para evitar datos inválidos
- Verificación de usuarios activos antes de login
- Limpieza de datos al eliminar usuarios (estadísticas asociadas)

10. Compatibilidad

- Soporte multi-plataforma con .NET MAUI
- Adaptación de UI según plataforma
- Compatibilidad con diferentes tamaños de pantalla

Planificación

Para la planificación, me he centrado en crear la estructura visual, es decir, el diseño de la interfaz de la aplicación para hacerla tanto estética como funcional de cara a la experiencia de usuario; y en paralelo a cada segmento, configuro la lógica y aplico las funciones de cada elemento del segmento.

Primero creo el segmento del menú principal, después creo el menú del juego con los modos de juego aplicados y configurados, a continuación creo el menú de usuarios y estadísticas, y por último creo la base de datos para almacenar los datos de los jugadores y sus resultados obtenidos.

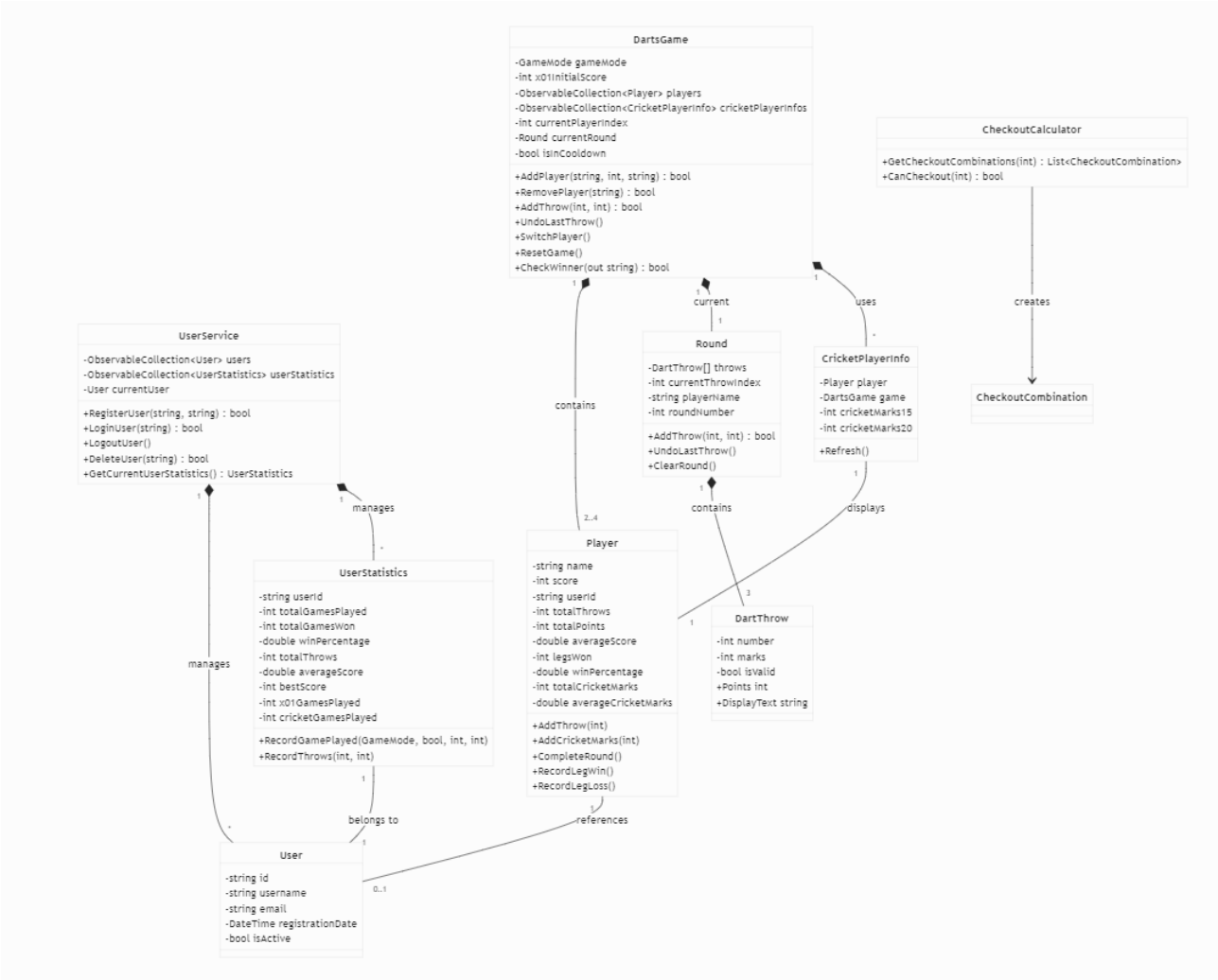
El tiempo invertido en el desarrollo de esta aplicación ha sido de unas 125 horas aproximadamente, debido mayormente a la complejidad de varios métodos y herramientas que he utilizado y que requerían de estudio y documentación para comprender sus respectivas funciones.

Presupuesto

Concepto	Horas	Tarifa/h	Subtotal
Análisis y Diseño	40	$50	$2,000
Desarrollo Core	120	$60	$7,200
Desarrollo UI/UX	100	$55	$5,500
Funcionalidades Avanzadas	60	$60	$3,600
Testing y QA	80	$50	$4,000
Deployment	40	$55	$2,200
Documentación	30	$45	$1,350
Subtotal Desarrollo	**470**	-	**$25,850**
Herramientas y Servicios	-	-	$1,323 - $1,623
TOTAL ESTIMADO	-	-	**$27,173 - $27,473**

Documentación

Diagrama de clases



Clase DartsGame (Núcleo del juego)

DartsGame gestiona la partida. Contiene una colección de Player (2-4), el modo (X01 o Cricket), el jugador actual, la ronda actual (Round), y el estado (cooldown, puntuación inicial en X01). Expone métodos para agregar/eliminar jugadores, registrar lanzamientos, cambiar de jugador, reiniciar y verificar ganador. También mantiene información de Cricket para cada jugador (CricketPlayerInfo).

Clase Player (Jugador)

Player representa a un jugador en la partida. Tiene nombre, puntuación actual, y opcionalmente un userId si es usuario registrado. Calcula estadísticas: total de

lanzamientos, puntos totales, promedio por lanzamiento, victorias/derrotas, porcentaje de victorias, y estadísticas específicas de Cricket (marcas totales y promedio). Actualiza estas métricas cuando se registran lanzamientos o se completan rondas.

Clase User (Usuario del sistema)

User representa un usuario registrado. Contiene identificador único, nombre de usuario, email, fecha de registro y estado activo/inactivo. No contiene lógica de juego; solo datos de identificación.

Clase UserStatistics (Estadísticas persistentes)

UserStatistics almacena estadísticas históricas de un usuario. Se vincula al usuario mediante userId. Incluye partidas jugadas y ganadas, porcentaje de victorias, totales de lanzamientos y puntos, promedio de puntuación, mejor puntuación, promedio de turnos por partida, estadísticas por modo (X01 y Cricket), y estadísticas específicas de Cricket (marcas totales y promedio). Se actualiza cuando el usuario participa en partidas y se persiste en JSON.

Clase UserService (Gestión de usuarios)

UserService es un singleton que gestiona usuarios y estadísticas. Mantiene colecciones de User y UserStatistics, y el usuario actual en sesión. Expone registro, login, logout, eliminación de usuarios, y obtención/actualización de estadísticas. Persiste y carga datos desde archivos JSON.

Clase Round (Ronda de juego)

Round representa una ronda de 3 lanzamientos de un jugador. Contiene un array de 3 DartThrow, el índice del lanzamiento actual, el nombre del jugador y el número de ronda. Permite agregar lanzamientos, deshacer el último, y calcula el total de puntos de la ronda. Indica si está completa.

Clase DartThrow (Lanzamiento individual)

DartThrow representa un lanzamiento. Tiene número (1-20, 25 para Bull exterior, 50 para Bull interior, 0 para fallido), marcas (1=simple, 2=doble, 3=triple), y valida si es válido según las reglas. Calcula puntos (número por marcas) y genera texto de visualización.

Clase CricketPlayerInfo (Información de Cricket para UI)

CricketPlayerInfo es un wrapper para mostrar información de Cricket en la interfaz. Se vincula a un Player y a un DartsGame. Expone propiedades para las marcas de cada número válido en Cricket (15, 16, 17, 18, 19, 20, 25) y los números cerrados. Se actualiza cuando cambian las marcas en el juego.

Clase CheckoutCalculator (Calculadora de checkout)

CheckoutCalculator es una clase estática que calcula combinaciones de checkout para X01. Dado un puntaje restante, busca combinaciones válidas de 1, 2 o 3 dardos que permitan cerrar (terminar en doble o Bull). Retorna una lista de CheckoutCombination con el texto y los puntos totales.

Relaciones entre clases

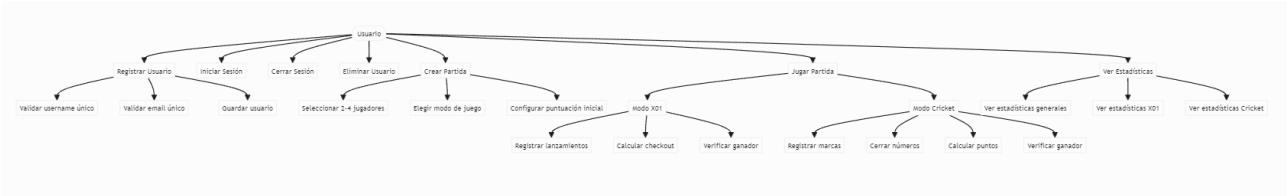
DartsGame contiene múltiples Player (2-4). Cada Player puede referenciar un User mediante userId. DartsGame tiene una Round actual, que contiene 3 DartThrow. DartsGame también mantiene CricketPlayerInfo para cada jugador en modo Cricket.UserService gestiona colecciones de User y UserStatistics. Cada UserStatistics está vinculado a un User mediante userId (relación 1:1). Cuando un usuario registrado participa, DartsGame actualiza sus estadísticas a través de UserService.CheckoutCalculator es independiente y se usa cuando DartsGame necesita calcular combinaciones de checkout en modo X01.

Flujo de funcionamiento

Al iniciar una partida, DartsGame crea Player (con o sin userId). Cada turno crea una Round con 3 DartThrow vacíos. Al registrar un lanzamiento, se crea un DartThrow, se valida, se agrega a la Round, y si el jugador tiene userId, se actualizan sus estadísticas en UserStatistics vía UserService.En modo Cricket, CricketPlayerInfo refleja las marcas de cada número. En modo X01, CheckoutCalculator calcula combinaciones posibles.Al finalizar una partida, se registran resultados en UserStatistics y se persisten en JSON mediante UserService.En resumen, DartsGame coordina el juego, Player representa jugadores, Round y DartThrow modelan los lanzamientos, User y UserStatistics gestionan datos persistentes, y UserService centraliza la persistencia y el acceso a usuarios y estadísticas.

TFG

Diagrama de casos de uso



Casos de uso principales

El diagrama muestra las acciones que puede realizar un usuario en la aplicación. Todos parten del actor Usuario.

Gestión de usuarios

Registrar Usuario: Permite crear una cuenta. Requiere validar que el username y el email sean únicos. Si son únicos, se guarda el usuario y se crea su registro de estadísticas.**Iniciar Sesión**: Permite autenticarse con un username existente. El sistema verifica que el usuario exista y esté activo, y establece la sesión actual.

Cerrar Sesión: Finaliza la sesión del usuario actual sin eliminar datos.

Eliminar Usuario: Permite eliminar una cuenta. Requiere confirmación y elimina también sus estadísticas asociadas.

Creación y configuración de partidas

Crear Partida: Permite iniciar una nueva partida. Incluye tres subcasos:

Seleccionar Jugadores: Elegir entre 2 y 4 jugadores. Pueden ser usuarios registrados o invitados. El sistema valida el rango.

Elegir Modo de Juego: Seleccionar entre X01 o Cricket.

Configurar Puntuación Inicial: Solo en X01. Elegir 301, 501 o 701.

Jugar Partida

Jugar Partida se divide en dos flujos según el modo:

Modo X01

Registrar Lanzamientos: El jugador ingresa número (1-20, 25, 50, 0) y marcas (1-3). El sistema valida y registra el lanzamiento, resta puntos y actualiza estadísticas.

Calcular Checkout: Muestra combinaciones posibles para cerrar con el puntaje restante (hasta 5 opciones).

Verificar Ganador: Comprueba si algún jugador llegó a 0. Si es así, declara ganador y registra resultados.

Modo Cricket

Registrar Marcas: El jugador ingresa número válido (15-20, 25) y marcas (1-3). El sistema registra marcas y actualiza el estado de cada número.

Cerrar Números: Cuando un jugador alcanza 3 marcas en un número, se marca como cerrado. Si otros jugadores no lo tienen cerrado, puede ganar puntos.

Calcular Puntos: Si un número ya está cerrado por el jugador y no por todos los demás, las marcas adicionales generan puntos. También se procesan overmarks.

Verificar Ganador: Comprueba si algún jugador cerró todos los números (15-20 y 25). Si hay varios, gana el de mayor puntuación.

Ver Estadísticas

Permite consultar estadísticas del usuario logueado. Se divide en tres subcasos:

Ver Estadísticas Generales: Muestra totales de partidas, victorias, porcentaje de victorias, mejor puntuación, promedio de turnos, totales de lanzamientos y puntos, y promedio de puntuación.

Ver Estadísticas X01: Muestra partidas jugadas y ganadas en X01.

Ver Estadísticas Cricket: Muestra partidas jugadas y ganadas en Cricket, total de marcas y promedio de marcas por ronda.

Relaciones y dependencias

Los casos de uso tienen dependencias lógicas:

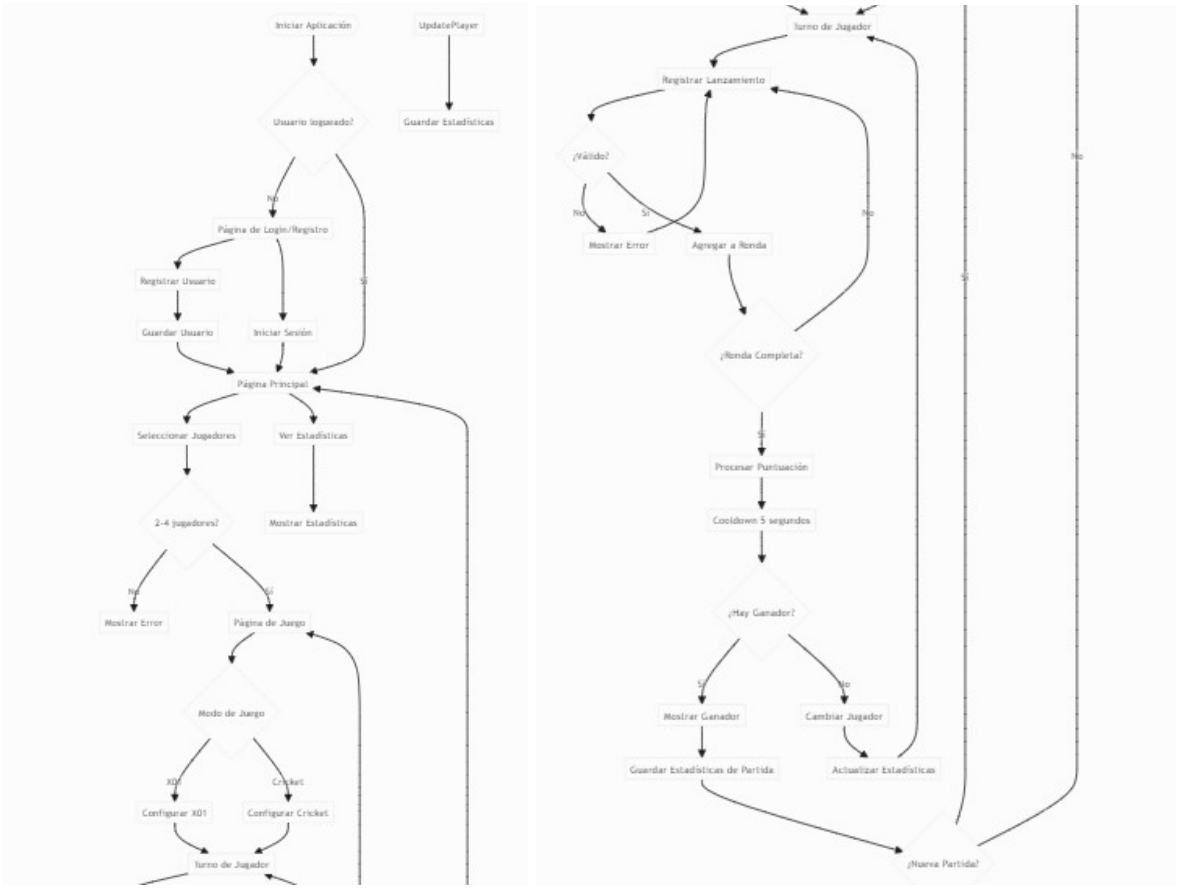
- Para crear una partida, primero se debe seleccionar jugadores y configurar el modo.
- Para jugar una partida, primero se debe crear y configurar.
- En X01, calcular checkout depende de registrar lanzamientos.
- Verificar ganador depende de registrar lanzamientos o marcas según el modo.

- Ver estadísticas requiere que el usuario esté logueado.
- Eliminar usuario elimina también sus estadísticas.

Flujo típico de uso

Un usuario nuevo primero se registra. Luego inicia sesión. Crea una partida seleccionando jugadores y modo. Durante el juego, registra lanzamientos o marcas según el modo. El sistema calcula checkout (en X01) o procesa cierres y puntos (en Cricket). Al finalizar, se verifica el ganador y se registran estadísticas. El usuario puede consultar sus estadísticas en cualquier momento.En resumen, el diagrama organiza las acciones del usuario desde el registro hasta jugar partidas y consultar estadísticas, con validaciones y dependencias entre casos de uso.

Diagrama de flujo



Inicio de la aplicación

Al iniciar, el sistema verifica si hay un usuario logueado. Si no hay sesión, redirige a la página de Login/Registro. Si hay sesión, va a la página principal.

Proceso de autenticación

En la página de Login/Registro hay dos opciones:

Registrar Usuario: Se ingresan username y email. El sistema valida que no existan duplicados. Si son válidos, se guarda el usuario, se crean sus estadísticas iniciales y se redirige a la página principal. Si hay duplicados, se muestra error.

Iniciar Sesión: Se selecciona un usuario existente. El sistema valida que exista y esté activo. Si es válido, se establece la sesión y se redirige a la página principal. Si no, se muestra error.

Selección de jugadores

En la página principal, al iniciar una partida se abre la selección de jugadores. El usuario selecciona entre 2 y 4 jugadores (registrados o invitados). El sistema valida el rango. Si hay menos de 2 o más de 4, muestra error y permite corregir. Si es válido, continúa a la página de juego.

Configuración del juego

En la página de juego, el usuario elige el modo:

Modo X01: Se configura la puntuación inicial (301, 501 o 701). Cada jugador inicia con ese puntaje.

Modo Cricket: No requiere configuración adicional. Cada jugador inicia con puntuación 0 y sin números cerrados.

Bucle principal del juego

Comienza el turno del primer jugador. El sistema muestra el jugador actual y su puntuación.

Registrar lanzamiento: El usuario ingresa número y marcas. El sistema valida:

- Número válido (1-20, 25, 50, 0)
- Marcas válidas (1-3)

- Combinaciones permitidas (por ejemplo, Bull solo 1-2 marcas)

Si es inválido, muestra error y permite reintentar. Si es válido, agrega el lanzamiento a la ronda actual.

Verificar estado de la ronda: Tras agregar un lanzamiento válido, el sistema verifica si la ronda está completa (3 lanzamientos). Si no, vuelve a pedir el siguiente lanzamiento. Si está completa, procesa la puntuación.

Procesamiento de puntuación

Al completar una ronda, el sistema procesa según el modo:

En modo X01: Resta los puntos de la ronda del puntaje del jugador. Valida que no quede negativo. Si queda negativo, la ronda no cuenta (bust). Si es válido, actualiza el puntaje.

En modo Cricket: Procesa cada lanzamiento de la ronda. Para cada número válido, actualiza marcas. Si un número llega a 3 marcas, se marca como cerrado. Si ya estaba cerrado y otros jugadores no lo tienen cerrado, se otorgan puntos por marcas adicionales (overmarks).

Cooldown entre rondas

Tras procesar la puntuación, se activa un cooldown de 5 segundos. Durante este tiempo no se pueden registrar lanzamientos. El sistema muestra una cuenta regresiva visual. Al terminar, continúa automáticamente.

Verificación de ganador

Tras el cooldown, el sistema verifica si hay ganador:

En modo X01: Si algún jugador tiene puntuación exactamente 0, es ganador.

En modo Cricket: Si algún jugador cerró todos los números válidos (15-20 y 25), es ganador. Si varios los cerraron, gana el de mayor puntuación.Si no hay ganador, cambia al siguiente jugador y comienza una nueva ronda.

Cambio de jugador

Si no hay ganador, el sistema avanza al siguiente jugador en orden. Actualiza el indicador de jugador actual, crea una nueva ronda (incrementa el número de ronda), reinicia los 3 lanzamientos y vuelve al bucle principal.

Finalización de la partida

Si hay ganador, el sistema muestra un mensaje con el nombre del ganador. Luego registra los resultados:

- Marca victoria para el ganador y derrota para los demás.
- Para cada jugador registrado, actualiza estadísticas: incrementa partidas jugadas, victorias (si corresponde), totales de lanzamientos y puntos, mejor puntuación, rondas jugadas, y estadísticas por modo (X01 o Cricket).
- Guarda las estadísticas en el archivo JSON.

Opciones post-partida

Tras guardar estadísticas, el sistema pregunta si iniciar una nueva partida. Si elige nueva partida, vuelve a la página de juego con los mismos jugadores y configuración. Si no, regresa a la página principal.

Navegación adicional

Desde la página principal, el usuario puede:

Ver Estadísticas: Muestra estadísticas del usuario logueado. Si no hay sesión, muestra valores en cero. Si hay sesión, muestra datos actualizados desde los archivos JSON.

Gestión de Usuarios: Permite registrar, iniciar sesión, cerrar sesión o eliminar usuarios. Cualquier cambio se guarda en los archivos JSON.

Flujo de datos y persistencia

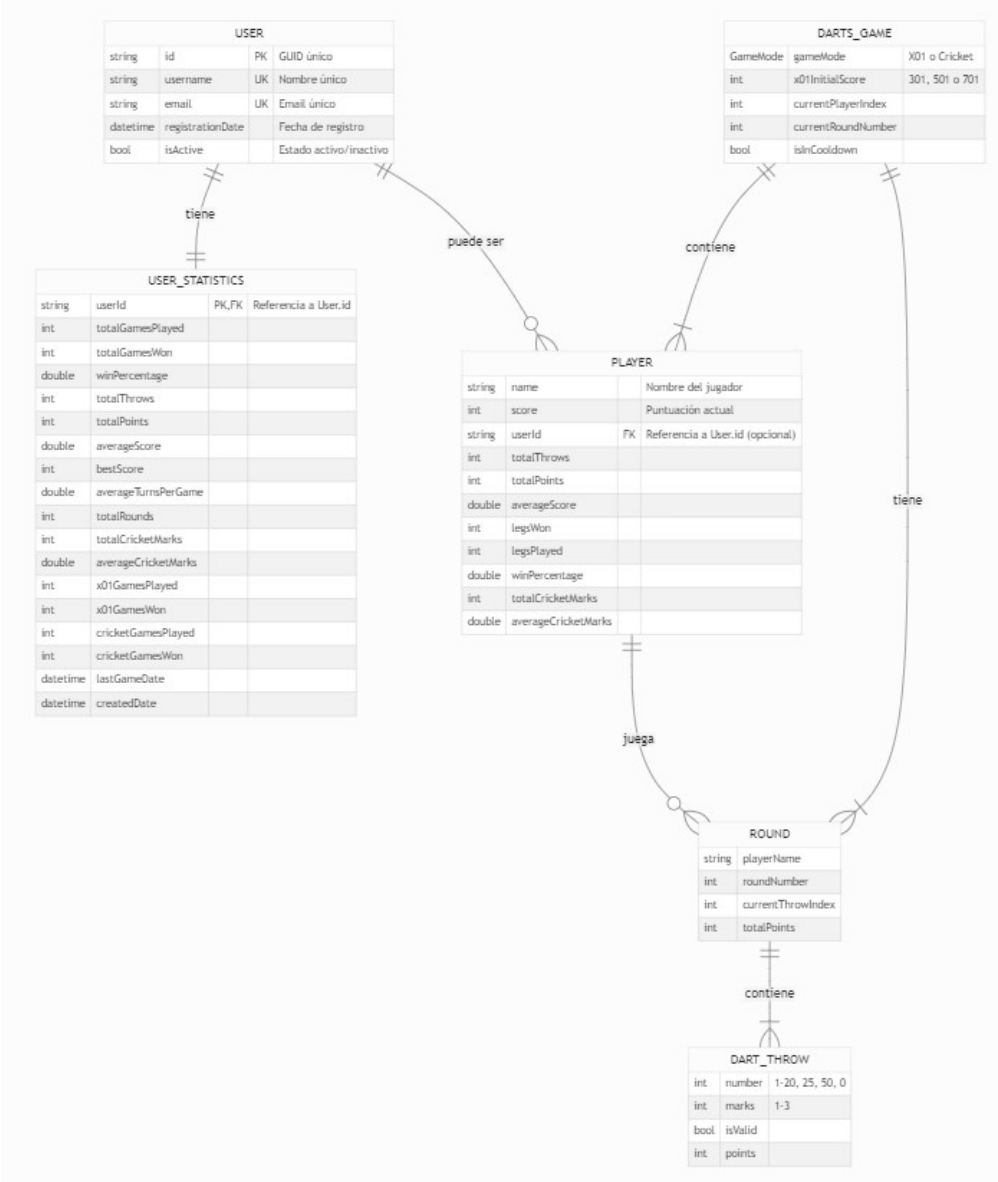
Durante el juego, los datos están en memoria. Al registrar lanzamientos, se actualizan estadísticas en memoria. Al finalizar una partida, se persisten las estadísticas de usuarios registrados en JSON. Los usuarios y sus estadísticas se cargan al iniciar la aplicación y se guardan tras cambios importantes (registro, eliminación, finalización de partida).

Manejo de errores

En cada paso hay validaciones. Si falla una validación, se muestra un mensaje y se permite corregir sin perder el progreso. Los errores comunes incluyen lanzamientos inválidos, nombres duplicados, límites de jugadores y usuarios inexistentes.En resumen, el flujo va desde el inicio de la aplicación, pasando por autenticación, selección de

jugadores, configuración, bucle de juego con validaciones, procesamiento de puntuación, verificación de ganador, registro de estadísticas y opciones post-partida, con persistencia en archivos JSON y manejo de errores en cada etapa.

Diagrama de base de datos



Entidad USER (Usuario)

Representa un usuario registrado. Contiene:

- **id**: identificador único (GUID). Clave primaria.
- **username**: nombre de usuario único. No puede repetirse.

- **email**: email único. No puede repetirse.
- **registrationDate**: fecha y hora de registro.
- **isActive**: indica si el usuario está activo o inactivo.

Se almacena en users.json. Cada registro es un usuario del sistema.

Entidad USER_STATISTICS (Estadísticas de Usuario)

Almacena las estadísticas históricas de un usuario. Contiene:

- **userId**: identificador del usuario. Clave primaria y foránea hacia USER.
- **totalGamesPlayed**: total de partidas jugadas.
- **totalGamesWon**: total de partidas ganadas.
- **winPercentage**: porcentaje de victorias (calculado).
- **totalThrows**: total de lanzamientos realizados.
- **totalPoints**: total de puntos acumulados.
- **averageScore**: promedio de puntos por lanzamiento.
- **bestScore**: mejor puntuación alcanzada.
- **averageTurnsPerGame**: promedio de turnos por partida.
- **totalRounds**: total de rondas jugadas.
- **totalCricketMarks**: total de marcas de Cricket acumuladas.
- **averageCricketMarks**: promedio de marcas de Cricket por ronda.
- **x01GamesPlayed**: partidas jugadas en modo X01.
- **x01GamesWon**: victorias en modo X01.
- **cricketGamesPlayed**: partidas jugadas en modo Cricket.
- **cricketGamesWon**: victorias en modo Cricket.
- **lastGameDate**: fecha y hora de la última partida.
- **createdDate**: fecha de creación del registro de estadísticas.

Se almacena en user_statistics.json. Cada registro corresponde a un usuario.

Relación entre USER y USER_STATISTICS

Relación uno a uno (1:1). Cada usuario tiene exactamente un registro de estadísticas, y cada registro pertenece a un usuario. La relación se establece mediante userId en USER_STATISTICS, que referencia id de USER.Al registrar un usuario, se crea automáticamente su registro de estadísticas con valores iniciales en cero. Si se elimina un usuario, se eliminan también sus estadísticas (eliminación en cascada).

Estructura de almacenamiento

Aunque no es una base de datos relacional, los datos se organizan como si lo fuera:**Archivo users.json**: contiene un array de objetos USER. Cada objeto tiene los campos mencionados. Se carga al iniciar la aplicación y se guarda tras cambios (registro, eliminación, cambios de estado).**Archivo user_statistics.json**: contiene un array de objetos USER_STATISTICS. Cada objeto tiene los campos mencionados y su userId vincula con USER. Se carga al iniciar y se guarda tras actualizaciones durante las partidas.

Integridad referencial

El sistema mantiene la integridad de forma lógica:

- Al crear un usuario, se crea su registro de estadísticas con el mismo userId.
- Al eliminar un usuario, se eliminan sus estadísticas.
- No puede existir un USER_STATISTICS sin un USER correspondiente.
- No puede existir un USER sin su USER_STATISTICS inicial.

Actualización de datos

Las estadísticas se actualizan durante las partidas:

- Al registrar lanzamientos: se incrementan totalThrows y totalPoints, y se recalcula averageScore.
- Al completar rondas: se incrementa totalRounds y se recalcula averageTurnsPerGame.

- Al finalizar partidas: se incrementan totalGamesPlayed, totalGamesWon (si corresponde), y los contadores por modo. Se actualiza lastGameDate y, si aplica, bestScore.
- En Cricket: se actualizan totalCricketMarks y averageCricketMarks.

Consultas y acceso a datos

El sistema accede a los datos mediante UserService:

- Para obtener un usuario: busca en la colección en memoria cargada desde users.json.
- Para obtener estadísticas: busca por userId en la colección cargada desde user_statistics.json.
- Para actualizar estadísticas: modifica el objeto en memoria y luego guarda el archivo completo.

Datos transitorios no persistentes

Algunos datos no se almacenan:

- **Partidas en curso**: se mantienen en memoria durante la ejecución.
- **Jugadores activos**: se crean al iniciar una partida y se eliminan al finalizar.
- **Rondas y lanzamientos**: solo existen durante la partida.
- **Jugadores invitados**: no se guardan como usuarios.

Solo se persisten los datos de usuarios registrados y sus estadísticas históricas.

Ventajas y limitaciones del modelo

Ventajas:

- Simplicidad: no requiere servidor ni base de datos.
- Portabilidad: los archivos JSON son fáciles de mover.
- Suficiente para el alcance actual.

Limitaciones:

- Escalabilidad: puede ser lento con muchos usuarios.

- Concurrencia: no maneja acceso simultáneo de múltiples usuarios.
- Consultas complejas: requiere cargar todo en memoria.
- Integridad: depende de la lógica de la aplicación.

Flujo de datos

Al iniciar la aplicación:

1. Se cargan usuarios desde users.json a memoria.
2. Se cargan estadísticas desde user_statistics.json a memoria.
3. Se establecen las relaciones mediante userId.

Durante la ejecución:

- Los datos se modifican en memoria.
- Las relaciones se mantienen mediante referencias en código.

Al guardar:

1. Se serializan las colecciones a JSON.
2. Se escriben los archivos completos.
3. Se mantiene la consistencia entre ambos archivos.

En resumen, el modelo E-R tiene dos entidades principales (USER y USER_STATISTICS) con una relación 1:1, almacenadas en archivos JSON separados pero vinculadas por userId, con actualizaciones en memoria y persistencia periódica, manteniendo la integridad mediante la lógica de la aplicación.

Tecnologías usadas

Visual Studio (.NET MAUI)

XAML: interfaz

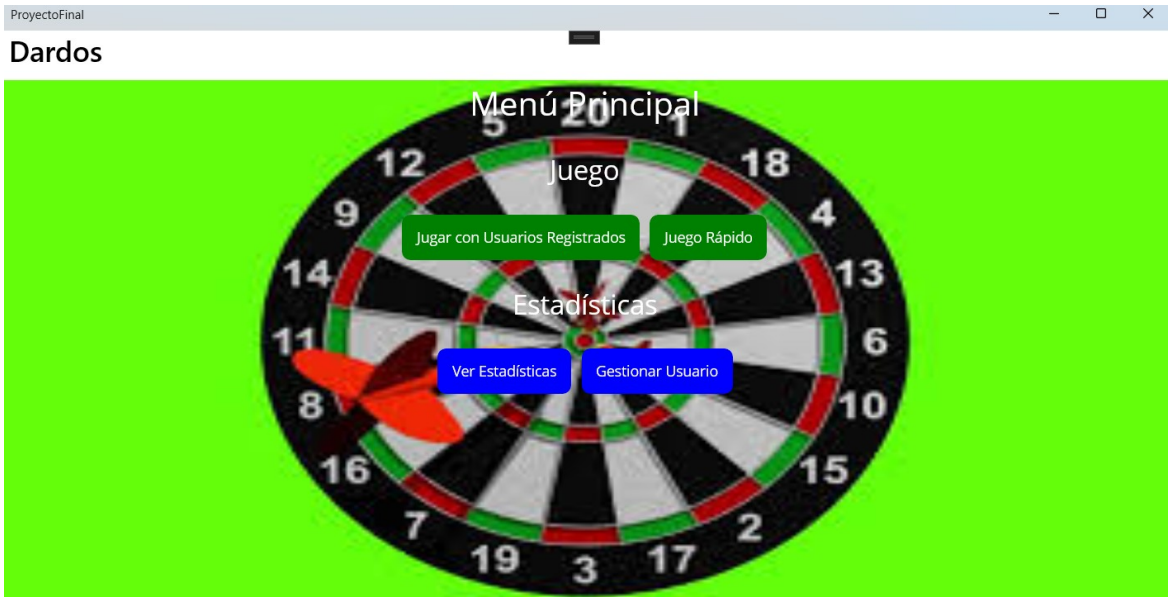
C#: lógica

Base de datos: JSON

Repositorio: GitHub

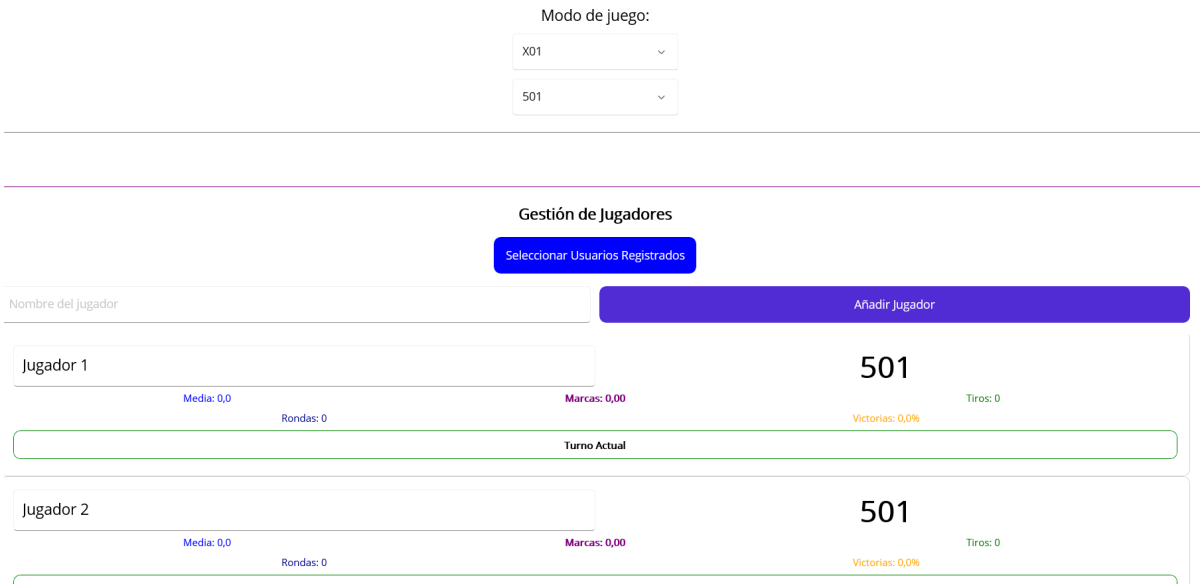
Manual de usuario

Una vez se inicia la aplicación, nos encontramos con el menú principal, que cuenta con una interfaz muy sencilla que indica claramente los modos con los que cuenta.



JUEGO

Si nos metemos en el modo “Juego Rápido”, entramos en el menú para jugar. De primeras, contamos con un selector de modo de juego, a la vez que se indican los jugadores que vayan a participar en el juego.



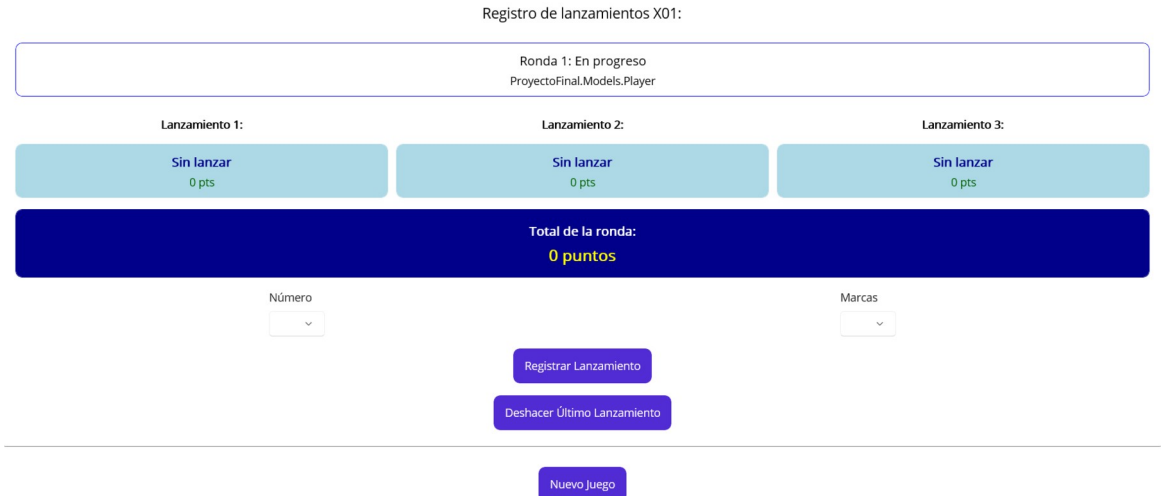
Aprovechando que, por defecto, el menú se inicia en el modo X01, debajo del selector de modo de juego, se puede escoger con qué puntuación se iniciará el juego, pudiendo ser seleccionables las puntuaciones 301, 501 y 701. Esta opción es exclusiva del modo de juego en el que nos encontramos.

Volviendo al apartado de los jugadores, por defecto se incorporan dos jugadores, que es el mínimo que deben participar para que se inicie el juego; sin embargo, se pueden

agregar hasta 4 jugadores agregándoles un nombre, tanto siendo nuevos como modificando los nombres de los jugadores ya agregados...

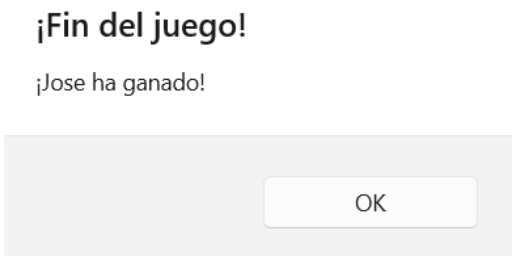
...dando paso a este resultado, siendo 4 jugadores con los nombres 1, 2, 3 y 4, cuando anteriormente los dos primeros jugadores eran Jugador 1 y Jugador 2.

Siguiendo con la aplicación, después de configurar el modo de juego y los jugadores, aparece el registro de los lanzamientos, en el que cada jugador en su correspondiente turno, cuando lanza un dardo, selecciona el número (del 0 al 20, siendo el 0 un fallo) y la marca (que multiplica el número obtenido x1, x2 o x3) obtenidos con un máximo de 3 registros seguidos por jugador, ya que cada uno lanza 3 dardos por cada ronda, hasta que pasa el turno al siguiente.

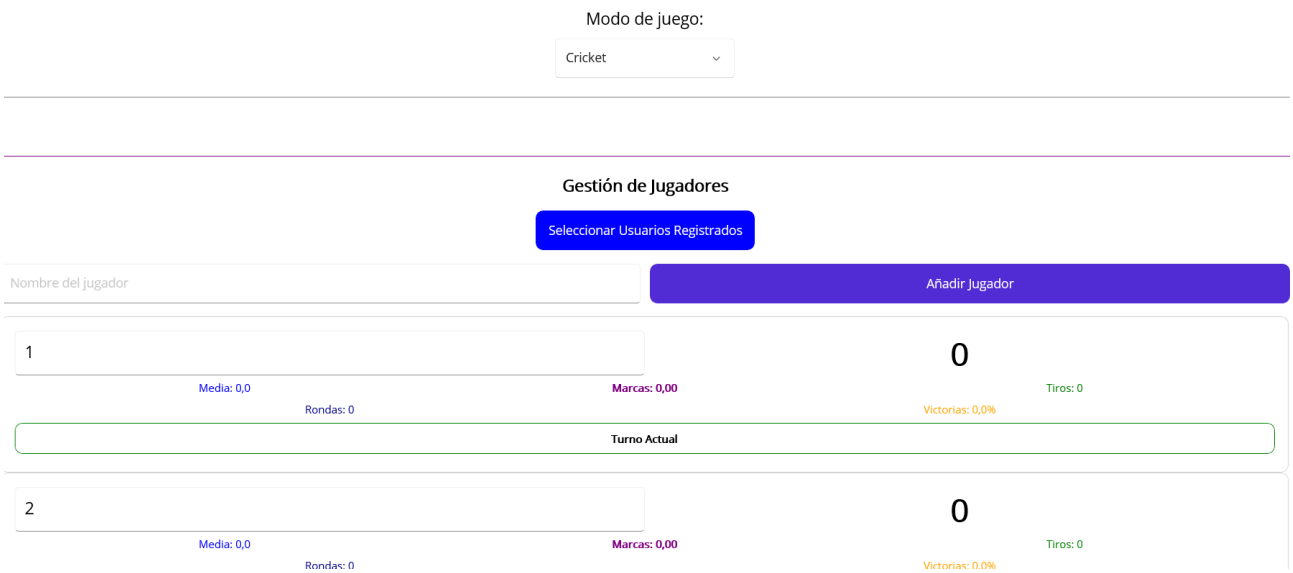


Cuando un jugador tiene puntuación suficiente para reducirlo a 0 en la ronda que le toca, la aplicación indica varias opciones de cierre para ayudarle a escoger la que mejor le convenga en base a su nivel de habilidad.

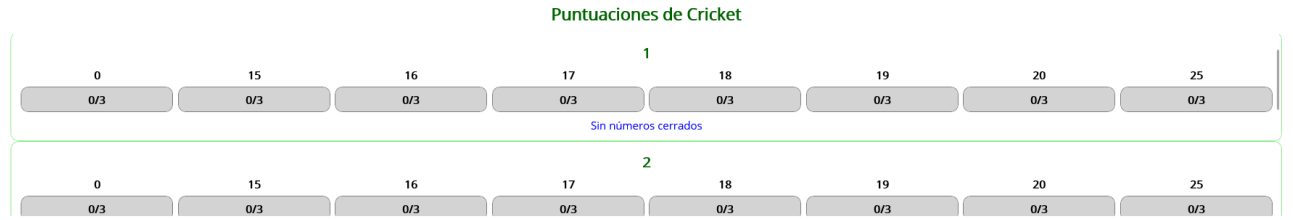
Cuando un jugador consigue cerrar, o sea, llegar a 0 sin pasarse, gana el juego y se reinicia.



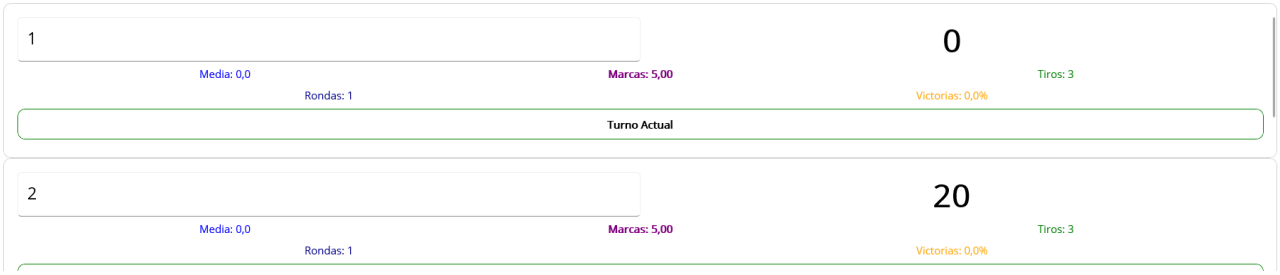
Ya explicado este modo de juego, pasamos al siguiente modo de juego disponible en la aplicación: Cricket.



En Cricket, cada jugador tiene varios números que tiene que cerrar, siendo éstos desde el 20 hasta el 15, incluyendo la diana (25), tal como se muestra en el indicador.



Utilizando el mismo registro de lanzamiento replicado para este modo, los jugadores tienen que hacer 3 marcas en cada uno de los números disponibles para obtenerlo y utilizarlo para sumar puntos en ese número obteniendo más marcas. Esos resultados se muestran en el indicador y la puntuación “extra” obtenida se agrega al marcador.



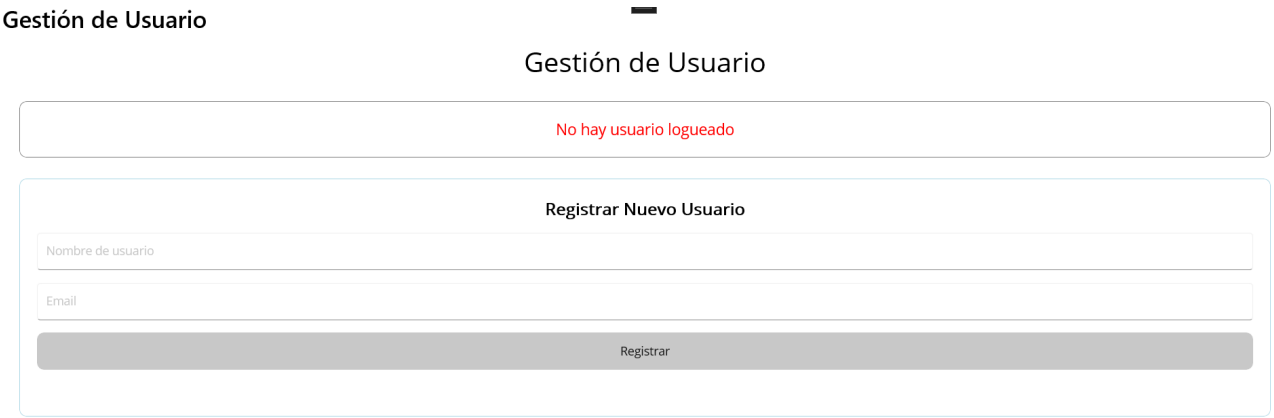
Para que un jugador no pueda seguir sumando puntos en un número, los demás jugadores deben obtener ese mismo número para cerrarlo y que nadie lo pueda utilizar.

El jugador que logre todos los números y esté por delante de los demás en la puntuación gana el juego.

En ambos modos de juego, cada vez que se lanza, los jugadores pueden revisar durante la partida una media de la puntuación obtenida y de las marcas realizadas por cada ronda que disputan, sirviendo como indicador para valorar las estadísticas de los jugadores y determinar el nivel de habilidad de cada jugador, aunque importa más una media que otra dependiendo de cada modo de juego.

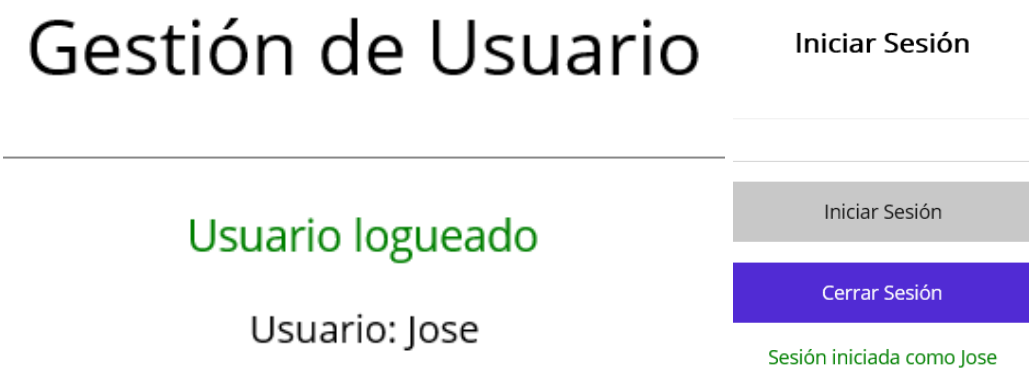
ESTADÍSTICAS

Ahora nos dirigimos desde el menú principal al modo “Gestionar Usuario”, en donde se pueden registrar usuarios agregando su nombre y su email, con la opción de ser eliminados de manera fácil y con un solo botón.

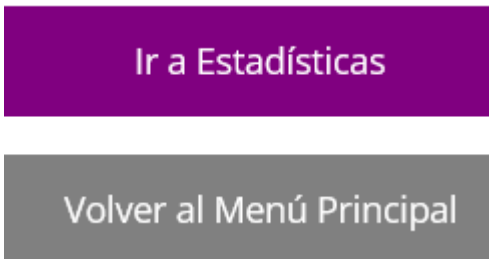


Una vez agregado, se puede iniciar sesión en la aplicación con ese usuario para utilizarlo de cara a las funciones avanzadas que ofrece.

Cuando se haya iniciado sesión, aparecerán estos indicadores de que se ha iniciado sesión.



Ahora, se puede acceder a la sección “Ir a Estadísticas”, opción que se puede acceder directamente en el menú principal con la opción “Ver Estadísticas”.



Lo que nos encontramos es un registro de las estadísticas en cuanto a juegos o partidas disputadas, puntuaciones obtenidas con los lanzamientos y datos de cada modo de juego.

Estadísticas Generales

Partidas jugadas: 2

Partidas ganadas: 2

Porcentaje de victorias: 100,0%

Mejor puntuación: 0

Promedio de turnos por partida: 5,0

Estadísticas de Lanzamientos

Total de lanzamientos: 18

Total de puntos: 1002

Puntuación promedio: 55,67

Marcas de Cricket: 0

Promedio de marcas Cricket: 0,00

Estadísticas por Modo de Juego

X01:

Partidas X01: 2

Victorias X01: 2

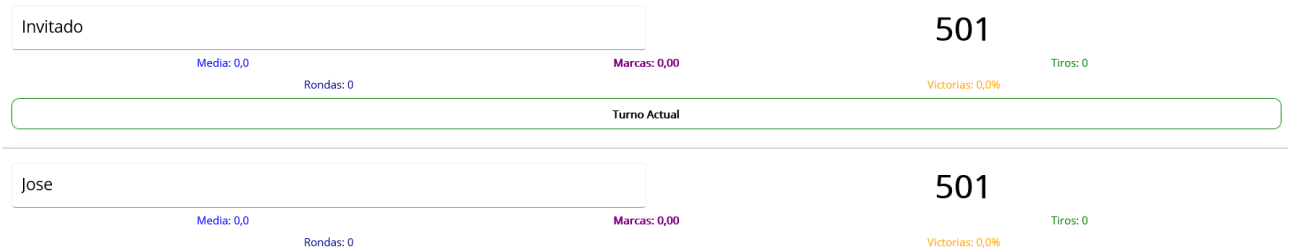
Cricket:

Partidas Cricket: 0

Victorias Cricket: 0

Para obtener registros, en el menú principal existe el modo “Jugar con Usuarios Registrados”, en el que se seleccionan los usuarios registrados en la aplicación que vayan a participar. En caso de que vayan a participar un jugador con su usuario y uno que no está registrado, el 2º jugador puede agregar temporalmente una cuenta de invitado para disputar las partidas, aunque sus estadísticas no se van a guardar en la cuenta de invitado.

Y una vez se incluyen los jugadores (máximo 4), se selecciona el modo de juego y se da inicio al juego.



Conclusiones

Hacer esta aplicación me ha supuesto una ampliación del conocimiento acerca de la programación. He tenido complicaciones a la hora de almacenar los datos en la base de datos, ya que tenía pensado hacerlo en una base de datos distinta a la utilizada pero la aplicación se quedaba congelada y, aprovechando la oportunidad de aprender una base de datos de la que personalmente no conocía su funcionamiento, opté por la base de datos que se utiliza en la aplicación, que me solucionó los problemas que se me presentaron.

Por lo demás, me ha venido bien refrescar la memoria de algunas herramientas aprendidas durante el grado que olvidaba cómo se implementaban y que me han reforzado mis habilidades de programación.

Bibliografía/Webgrafía

JSON:

https://youtu.be/RhxOTqFbI5Q?si=L-4ZGz8XNxL-c7z6

https://www.youtube.com/watch?v=YYfediyCwAU

https://www.json.org/json-es.html

https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve

Anexos