

Segunda y última parte

El presente obligatorio es continuación del primer obligatorio y será realizado en equipos de dos integrantes. El propósito es implementar la gestión del juego “La Mosqueta” con la gestión de los usuarios y juegos.

Administración de jugadores

El programa deberá contar con el siguiente menú:

- 1) **Gestionar usuario:** éste constará de las siguientes opciones (usar estructuras para guardar los datos):
 - 1) **Alta de jugador:** permitirá registrar los datos del jugador: CI (validar caracteres y largo*), fecha de nacimiento (validar que sea fecha válida*), nombre, apellido, alias (el cual no se debe repetir) y si está activo o no. Todos los jugadores al darse de alta contarán con un saldo de \$1000. **(Obligatorio)**
 - 2) **Baja de jugador:** permitirá marcar como inactivo al jugador, de manera que en el listado de jugadores no aparecerá, pero sí en los listados de partidas. Si el jugador dado de baja se registra nuevamente, simplemente se cambiará su estado de inactivo a activo. Para ello se ingresará el alias del jugador a dar de baja. **(Opcional)**
 - 3) **Modificación de jugador:** permitirá modificar los datos del jugador, para ello el jugador ingresará su alias y el programa mostrará todos los datos modificables permitiéndole guardar los datos o cancelar la operación. Todos los datos del jugador son modificables excepto, el alias. **(Obligatorio)**
- 2) **Consultas:**
 - 1) **Listado de jugadores:** Mostrará por cada jugador los siguientes datos: CI, nombre, apellido, alias y saldo actual. Ordenado ascendentemente por alias (considere sólo jugadores activos). **(Obligatorio)**
 - 2) **Listado de todas las apuestas:** Mostrará por cada apuesta, el alias del jugador, resultado de cada una de las apuestas (Ha acertado o No ha acertado) realizadas y monto apostado. **(Opcional)**

- 3) **Listado de apuestas por jugador:** se deberá solicitar el alias del jugador a consultar, salida: alias jugador, saldo actual, resultado de cada apuesta en orden cronológico (si acertó o no), el monto apostado en cada apuesta y el saldo resultante luego de esa apuesta. **(Obligatorio)**
- 3) **Apostar:** esta opción funcionará igual que el obligatorio anterior pero el jugador, al comenzar a apostar, ingresará el alias (que se deberá controlar que exista) y dará comienzo a la apuesta. Si es la primera vez que apuesta contará con un saldo inicial de \$1000. En caso que el jugador no tenga saldo disponible deberá indicarle que no puede seguir apostando porque no tiene más saldo disponible. Una vez que inicia el juego es la misma dinámica que se realizó en el obligatorio anterior, cuando termine su juego, sea por falta de saldo o por que decidió salir el jugador, se deberá volver al menú principal. Como se infiere de los listados se debe almacenar toda la información de cada apuesta y el saldo que tiene el jugador en cada apuesta. **(Obligatorio).**
- 4) **Salir:** Termina el programa. **(Obligatorio)**

Nota:

Todos los menús deben tener la opción de volver al menú anterior (excepto el primero que tendrá la opción de salir de la apuesta).

** Las validaciones deberán hacerse mediante funciones.*

Se pide

Continuar y entregar la implementación del programa en C/C++ que cumpla con los requisitos planteados anteriormente.

Restricciones

1. Para la entrada y salida de datos solamente se podrán utilizar las funciones scanf, printf, getchar, putchar.

2. No está permitido el uso de funciones de C/C++ que no se hayan visto en el curso (consultar si es necesario).

Se valorará además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. Se hará énfasis en buenas prácticas de programación como:

- Indentación.
- Utilización correcta de las estructuras de control.
- Algoritmos razonablemente eficientes.
- Uso de comentarios.
- Uso de constantes.
- Uso de nombres nemotécnicos.

Fecha de entrega

La fecha de entrega es el Domingo 15 de Junio