

Informe proyecto Trivia UCAB

Programación orientada a objetos

1. Descripción General

TriviaUCAB es una aplicación de consola en Java que implementa un juego de trivia por turnos, inspirado en el clásico juego de mesa Trivial Pursuit. El sistema permite a varios usuarios registrarse, iniciar sesión, gestionar preguntas de trivia y jugar en un tablero virtual con casillas de diferentes categorías.

2. Estructura del Proyecto

El código fuente se encuentra en el paquete TriviaUCAB.models y está organizado en clases, interfaces y enumeraciones que representan tanto la lógica del juego como la gestión de usuarios y preguntas.

Estructura de carpetas:

```
TriviaUCAB/  
  App.java  
  models/  
    brazo.java  
    Category.java  
    CategoryQuestion.java  
    Ficha.java  
    movimientoBidireccional.java  
    Question.java  
    Questions.java  
    Square.java  
    SquareCategory.java  
    SquareCenter.java  
    SquareRayo.java  
    SquareSpecial.java  
    TableTop.java  
    Turno.java  
    Usuario.java  
    Validator.java
```

3. Principales Componentes

3.1. Gestión de Usuarios

Usuario: Representa a un jugador, almacenando su nombre de usuario, contraseña (encriptada con SHA-256) y estadísticas de victorias.

Validator: Clase utilitaria para validaciones de entrada, conversión a JSON y encriptación de contraseñas.

3.2. Preguntas y Categorías

Category (enum): Enumera las categorías de preguntas (Geografía, Historia, Deportes/Pasatiempos, Ciencias Naturaleza, Arte/Literatura, Entretenimiento).

Question: Representa una pregunta de trivia, con su respuesta, categoría, creador y aprobador.

Questions: Administra las listas de preguntas (en espera, aprobadas, rechazadas), permite modificar, aprobar y obtener preguntas aleatorias por categoría.

3.3. Tablero y Casillas

Square (abstracta): Clase base para todas las casillas del tablero.

SquareCenter: Casilla central del tablero, de donde parten los "brazos" hacia las diferentes categorías.

SquareCategory: Casilla de una categoría específica, enlazada a otras casillas.

SquareRayo: Casilla especial que conecta los brazos del tablero.

SquareSpecial: Casilla especial con rutas alternativas.

Interfaces:

brazo: Define métodos para salir y entrar desde el centro a los brazos del tablero.

movimientoBidireccional: Define métodos para moverse adelante y atrás en el tablero.

CategoryQuestion: Define métodos para reaccionar ante preguntas de una categoría.

3.4. Juego y Turnos

Ficha: Representa la pieza de un jugador en el tablero, con su posición y progreso en las categorías.

TableTop: Controla la lógica principal del juego, el ciclo de turnos y la impresión del tablero.

Turno: Implementa la lógica de un turno individual usando Callable para permitir la ejecución concurrente (aunque en la versión actual se ejecuta secuencialmente).

3.5. Aplicación Principal

App: Clase principal que gestiona el flujo del programa, registro e inicio de sesión de usuarios, carga/guardado de datos y arranque de la partida.

4. Flujo de Juego

Inicio de Sesión/Registro: Los usuarios pueden registrarse o iniciar sesión. Las credenciales se validan y almacenan en archivos JSON.

Gestión de Preguntas: Los usuarios pueden agregar preguntas, que deben ser aprobadas por otros usuarios antes de ser usadas en el juego.

Inicio de Partida: Se crea un tablero virtual y se asignan fichas a los jugadores.

Turnos: Cada jugador tira el dado y mueve su ficha por el tablero, respondiendo preguntas de la categoría correspondiente.

Victoria: El primer jugador en completar todas las categorías y regresar al centro gana la partida.

5. Persistencia de Datos

Los usuarios y las preguntas se almacenan en archivos JSON en la carpeta .config del usuario.

Se utiliza la librería Gson para la serialización y deserialización de objetos.

6. Relaciones entre Clases

Ficha tiene una referencia a Usuario y a Square (posición en el tablero).

TableTop contiene una lista de Ficha y una referencia al SquareCenter.

SquareCenter contiene un arreglo de SquareCategory (los "brazos" del tablero).
Questions administra listas de Question.
Las casillas del tablero implementan interfaces según su comportamiento (brazo, movimiento bidireccional, reacción a preguntas).

7. Características Técnicas

Lenguaje: Java 17+ (recomendado)
Persistencia: Archivos JSON (usuarios y preguntas)
Interfaz: Consola (terminal)
Dependencias: Gson para JSON
Gestión de dependencias: Gradle

8. Observaciones y Recomendaciones

El diseño es modular y extensible, permitiendo agregar nuevas categorías o tipos de casillas fácilmente.
El uso de interfaces favorece la reutilización y la separación de responsabilidades.
Sería recomendable implementar pruebas unitarias y mejorar la gestión de errores para robustecer la aplicación.
La interfaz gráfica podría mejorarse en el futuro para una experiencia de usuario más amigable.

10. Conclusión

TriviaUCAB es un proyecto educativo y lúdico que integra conceptos de programación orientada a objetos, manejo de archivos, validación de datos y lógica de juegos de mesa. Su estructura modular y el uso de patrones de diseño lo hacen una base sólida para futuras mejoras, como una interfaz gráfica o integración en red.