

Guía de Instalación y Uso del Software – Proyecto TRIVIA-UCAB

30-05-2025

Luis Dos Santos, Jaiverson Bonillo y Ricardo Escalona

A continuación, presentamos una guía detallada sobre la instalación, compilación y uso de las aplicaciones desarrolladas para el proyecto TRIVIA-UCAB. El objetivo es facilitar la revisión y ejecución del sistema, que consiste en dos aplicaciones de consola Java: una para la configuración y otra para el juego en sí.

1. Introducción y Objetivos del Proyecto

El proyecto TRIVIA-UCAB simula el conocido juego de trivia, adaptado a una experiencia por consola. Se han desarrollado dos aplicaciones principales:

- **Aplicación de Configuración (AppConfig):** Permite la gestión de usuarios (registro y autenticación con datos cifrados) y la administración completa del banco de preguntas (creación, modificación, consulta, eliminación y un flujo de aprobación/rechazo). Esta aplicación también maneja la configuración del tiempo máximo de respuesta para cada pregunta.
- **Aplicación del Juego (AppJuego):** Permite a los jugadores participar en partidas de TRIVIA-UCAB, utilizando las preguntas aprobadas desde la aplicación de configuración. Incluye la lógica de turnos, movimiento en un tablero hexagonal simulado, obtención de "quesitos" por categoría, y la condición de victoria.

2. Requisitos del Sistema para la Ejecución

Para la correcta ejecución y compilación del proyecto, se requiere:

- **Java Development Kit (JDK):** Versión 17 o superior. El proyecto ha sido configurado y probado con Java 17 (LTS), pero es compatible con versiones más recientes como JDK 21 o 24.
- **Apache Maven:** Versión 3.6.x o superior, necesario para la gestión de dependencias y la compilación del proyecto.
- **Sistema Operativo:** Cualquier SO compatible con Java (Windows, macOS, Linux).
- **Terminal o Símbolo del Sistema:** Para la ejecución de comandos Maven y los archivos JAR de las aplicaciones.

3. Obtención y Estructura del Código Fuente

El proyecto completo se encuentra disponible en el repositorio de GitHub especificado para la evaluación. La estructura del proyecto es multi-módulo Maven, organizada de la siguiente manera:

- **trivia-ucab-parent**: Módulo padre que gestiona las configuraciones y dependencias comunes.
 - **trivia-domain**: Módulo que contiene las clases de entidad y enumeraciones compartidas por ambas aplicaciones (ej. PreguntaDetallada, CategoriaTrivia, CoordenadaHex).
 - **trivia-config-app**: Módulo de la aplicación de configuración.
 - Contiene en `src/main/resources/` el archivo `preguntasJuegoTrivia.json` que sirve como set inicial de preguntas.
 - **trivia-game-app**: Módulo de la aplicación del juego.

Para obtener el código fuente, se recomienda clonar el repositorio utilizando Git con el comando: `git clone <URL_DEL_REPOSITORIO_GITHUB>`. Alternativamente, se puede descargar el código fuente como un archivo ZIP desde GitHub y descomprimirlo en una carpeta local. Llamaremos a esta carpeta la "raíz del proyecto" (ej. `trivia-ucab-parent`).

4. Proceso de Compilación con Apache Maven

Para compilar el proyecto y generar los artefactos ejecutables (archivos JAR), se deben seguir estos pasos:

Primero, abra una terminal o símbolo del sistema. Luego, navegue a la carpeta raíz del proyecto (`trivia-ucab-parent`) que contiene el `pom.xml` principal, utilizando el comando `cd ruta/a/tu/proyecto/trivia-ucab-parent`.

Una vez en la carpeta raíz, ejecute el comando de Maven: `mvn clean package`. El comando `clean` limpia cualquier compilación anterior, y `package` compila el código fuente de todos los módulos y empaqueta las aplicaciones `trivia-config-app` y `trivia-game-app` en JARs ejecutables con todas sus dependencias incluidas.

Una vez que el comando termine exitosamente ("BUILD SUCCESS"), encontrará los JARs ejecutables en las siguientes ubicaciones dentro de la carpeta del proyecto: Para la Aplicación de Configuración, el JAR se ubicará en `trivia-ucab-parent/trivia-config-app/target/trivia-config-app-1.0-SNAPSHOT-executable.jar`. Para la Aplicación del Juego, el JAR se encontrará en `trivia-ucab-parent/trivia-game-app/target/trivia-game-app-1.0-SNAPSHOT-executable.jar`.

5. Gestión de Archivos de Datos

Las aplicaciones utilizan archivos JSON para la persistencia de datos. Cuando los JARs se ejecutan, estos archivos JSON se crearán y se leerán desde el mismo directorio donde se encuentre el JAR en ejecución. Para que ambas aplicaciones compartan la información correctamente (especialmente `banco_preguntas_gestion_compartido.json` y `usuarios_compartido.json`), se recomienda copiar ambos JARs ejecutables a una misma carpeta y ejecutarlos desde allí. Los archivos de datos son:

- `usuarios_compartido.json`: Almacena los datos de los usuarios registrados, con información sensible cifrada. Es creado y gestionado por `trivia-config-app.jar`.
- `banco_preguntas_gestion_compartido.json`: Contiene el banco de preguntas completo, incluyendo su estado (Esperando Aprobación, Aprobada, Rechazada), categoría, tiempo de respuesta, etc. Es gestionado por `trivia-config-app.jar` y leído por `trivia-game-app.jar` (solo preguntas aprobadas).
- `preguntasJuegoTrivia.json`: Es el archivo de set inicial de preguntas que se incluye como recurso dentro de `trivia-config-app.jar`. La aplicación de configuración puede importar estas preguntas al `banco_preguntas_gestion_compartido.json` si este último se encuentra vacío.
- `partida_guardada_compartida.json`: Almacena el estado de una partida del juego en curso, permitiendo continuarla posteriormente. Es gestionado por `trivia-game-app.jar`.

6. Ejecución y Uso de la Aplicación de Configuración (trivia-config-app.jar)

Para ejecutar la aplicación de configuración desde el JAR, primero copie el archivo `trivia-config-app-1.0-SNAPSHOT-executable.jar` a una carpeta de ejecución designada. Luego, abra una terminal en esa carpeta y ejecute el comando: `java -jar trivia-config-app-1.0-SNAPSHOT-executable.jar`.

Si prefiere ejecutar desde IntelliJ IDEA para desarrollo o una demostración rápida, navegue a `trivia-config-app/src/main/java/com/ucab/trivia/config/AppConfig.java`, haga clic derecho en el archivo y seleccione "Run 'AppConfig.main()'". El código está ajustado para que los archivos JSON se creen/lean en la carpeta `trivia-ucab-parent/` al usar rutas relativas como `../archivo.json`.

Las funcionalidades principales de la Aplicación de Configuración incluyen:

- **Registro de Usuario:** Permite crear nuevas cuentas con correo (validado) y contraseña de 6 caracteres (cifrada).
- **Inicio de Sesión:** Permite a usuarios registrados acceder a las funciones de gestión.
- **Gestión de Preguntas:** Incluye agregar nuevas preguntas (que inician en estado "Esperando Aprobación"), modificar preguntas no aprobadas (texto, respuesta, categoría, tiempo máximo configurable), eliminar preguntas, consultar preguntas (todas o por estado), y un sistema de aprobación/rechazo donde un usuario puede gestionar preguntas de otros usuarios, pero no las propias. Las preguntas aprobadas son las que se usarán en el juego.

7. Ejecución y Uso de la Aplicación del Juego (trivia-game-app.jar)

Para ejecutar la aplicación del juego desde el JAR, asegúrese de que `banco_preguntas_gestion_compartido.json` (generado previamente por `AppConfig`) exista en la carpeta de ejecución. Copie `trivia-game-app-1.0-SNAPSHOT-executable.jar` a esta misma carpeta y ejecute desde la terminal: `java -jar trivia-game-app-1.0-SNAPSHOT-executable.jar`.

Si ejecuta desde IntelliJ IDEA, navegue a `trivia-game-app/src/main/java/com/ucab/trivia/juego/AppJuego.java`, haga clic derecho y seleccione "Run 'AppJuego.main()'". Los archivos JSON compartidos se buscarán/crearán en `trivia-ucab-parent/`.

Las funcionalidades principales de la Aplicación del Juego son:

- **Inicio de Partida:** Permite elegir entre "Juego Nuevo" o "Cargar Partida Anterior" si existe un archivo de guardado. Para un juego nuevo, se ingresa el número de jugadores (1 a 6) y sus correos electrónicos para identificación. Se elige si el juego será por tiempo o no. La dificultad está fijada en "Fácil" para esta versión.
- **Mecánica del Juego:** Los jugadores comienzan en el centro de un tablero hexagonal simulado. Se determina el primer jugador por lanzamiento de dado. En cada turno, el jugador lanza un dado de 6 caras y mueve su ficha, eligiendo la dirección en la representación hexagonal. Al caer en una casilla, debe responder una pregunta de la categoría correspondiente. Si la respuesta es correcta (verificada por comparación de texto), el jugador rellena el espacio de esa categoría en su ficha y vuelve a lanzar. Si ya tenía esa categoría, igual vuelve a lanzar si acierta. Si responde incorrectamente, pierde el turno. Algunas casillas especiales permiten volver a lanzar. Si se juega por tiempo, un cronómetro en pantalla indicará el tiempo restante para responder.
- **Objetivo y Victoria:** El objetivo es conseguir los "quesitos" de las 6 categorías (Geografía, Historia, Deportes y pasatiempos, Ciencias y naturaleza, Arte y Literatura, Entretenimiento). Con la ficha completa, el jugador debe dirigirse al centro y caer exactamente en él. Allí, elige una categoría, responde una pregunta; si acierta, gana. Si falla, permanece en el centro para intentarlo en su siguiente turno.
- **Guardado Automático:** El juego se guarda después de cada movimiento o acción significativa.
- **Estadísticas:** Al finalizar, se muestran estadísticas como tiempo total de respuestas, preguntas correctas por categoría y juegos ganados.

8. Consideraciones Adicionales

- **Principios POO y Estándares:** Se ha procurado aplicar los principios de Programación Orientada a Objetos (DRY, YAGNI, KISS, SOLID) y los estándares de codificación Java discutidos en clase.
- **Documentación Javadoc:** El código fuente ha sido documentado utilizando Javadoc.
- **Control de Versiones:** El proyecto ha sido gestionado mediante Git y GitHub, con commits que reflejan el desarrollo de las diferentes funcionalidades.