# **Proyecto de Simulación Interactiva**



**Descripción:** Simula misiones en el universo de DeepRockGalactic. Elije misiones, mejora a tus mineros y accede a un historial de partidas. Observa como tus mineros progresan y se hacen más fuertes, recolecta créditos y experiencia hasta llegar a ser infalible en cada una de tus misiones.

**Características Principales:**

* Simulación y Gestión de Partidas, Mineros, Misiones.
* Sistema de Guardado Interno
* Conexión con Bases de Datos SQL

**Desarrollado por:** Bruno Acosta, Bruno Silva

**Fecha de Entrega:** 22/11/2023

**Versión:** 1.0.2

## Índice

[Simulador de Misiones para DeepRockGalactic 1](#_Toc1028614101)

[Proyecto de Simulación Interactiva 1](#_Toc964242968)

[Índice 1](#_Toc1227666218)

[1. Introducción 2](#_Toc135192540)

[2. Valoración de las Ventajas y los Inconvenientes 2](#_Toc735311114)

[3. Explicación del uso y funcionamiento 2](#_Toc2033917870)

[4 Elementos Adicionales 4](#_Toc1332796974)

[5 Entidad Relación de la base de datos 5](#_Toc1669936397)

## 1. Introducción

**Descripción:** Simula misiones en el universo de DeepRockGalactic. Elije misiones, mejora a tus mineros y accede a un historial de partidas. Observa como tus mineros progresan y se hacen más fuertes, recolecta créditos y experiencia hasta llegar a ser infalible en cada una de tus misiones.

**Funcionalidad:** La funcionalidad principal del aplicativo incluye la creación y gestión de partidas, la simulación de misiones con mineros y la exploración de resultados. Además, se incorpora un sistema de guardado tanto interno, mediante ficheros, como externo, mediante una conexión a bases de datos SQL.

**Alcance:** El alcance del proyecto se centra en proporcionar una experiencia interactiva y realista basada en el juego DeepRockGalactic. La simulación de misiones se lleva a cabo mediante la interacción con diversos elementos del juego, proporcionando a los usuarios un entorno inmersivo.

## 2. Valoración de las Ventajas y los Inconvenientes

**JDBC:** Aunque la implementación inicial de bases de datos puede presentar cierta complejidad, una vez que se han estructurado y planificado adecuadamente, ofrecen una comodidad notable en su ejecución. Proporcionan al usuario un mayor control sobre los datos almacenados, además de ser más eficientes y cómodas en el proceso de guardado y escritura de información.

**Ficheros:** Aunque al principio pueden parecer más sencillos de implementar, a medida que el programa crece y alcanza cierto tamaño, se convierten en un desafío considerable. Además, su eficacia en operaciones de lectura y escritura disminuye significativamente en comparación con las bases de datos. Asimismo, existe un riesgo notablemente mayor de pérdida de datos en caso de fallos.

## 3. Explicación del uso y funcionamiento

**Paquete AccesoDatos.Interfaces:**

**UsuarioDAO**

**PartidaDAO**

**MineroDAO**

**MisionesDAO**

Todas las interfaces DAO incorporan los métodos esenciales que permiten a sus implementadores correspondientes, ya sea mediante JDBC o a través de ficheros, operar de manera efectiva. Estos métodos son sobrescritos dentro de las clases JDBC o de ficheros para adaptarse de manera específica a las necesidades particulares, ya sea en el contexto de una base de datos o de archivos.

**Paquete Factoría:**

La función de la factoría es ser llamada para crear la clase necesaria con su tipo ya sea Fichero o JDBC, los métodos que son llamados dentro de la factoría son:

**public static UsuarioDAO crearUsuario**

**public static PartidaDAO crearPartida**

**public static MineroDAO crearMinero**

**public static MisionDAO crearMision**

Al llamar a estas clases, siempre se obtiene la clase creada con su tipo correspondiente, ya sea Fichero o JDBC, garantizando así la correcta y consistente creación de instancias.

**Paquete AccesoDatos. JDBC:**

**Configuración de la Conexión:** La aplicación establece una conexión con bases de datos SQL utilizando JDBC. Se definen esquemas y tablas para almacenar información relacionada con usuarios, partidas y mineros.

**Operaciones SQL Utilizadas:** Operaciones SQL como SELECT, INSERT, UPDATE y DELETE se utilizan para gestionar la información almacenada en la base de datos. Esto incluye la recuperación de datos de usuarios, el registro de nuevas partidas y la actualización de información de mineros.

**Paquete AccesoDatos. Ficheros:**

**EntradaSalida.java:** Es una clase abstracta que proporciona métodos útiles, como la lectura y el guardado en ficheros, a las clases dentro del paquete AccesoDatos.Ficheros. Cada clase posteriormente utiliza estos métodos, ajustándolos según sus propias necesidades

**Estructura de Ficheros:** El sistema de guardado interno utiliza archivos para almacenar información crucial de las partidas, incluyendo datos de usuarios, detalles de misiones y configuraciones de mineros.

**Proceso de Guardado:** El proceso de guardado implica la serialización de objetos clave, como usuarios, partidas y mineros, en archivos específicos utilizando mecanismos como Object Serialization en Java.

**Proceso de Lectura:** La lectura de datos implica la deserialización de archivos guardados para reconstruir las instancias de objetos originales, permitiendo la recuperación de partidas y configuraciones previas.

## 4 Elementos Adicionales

**Implementación de interfaz gráfica:** Se emplea la librería Java Swing para incorporar el código en una interfaz gráfica amigable y accesible para el usuario.

## 5 Entidad Relación de la base de datos

