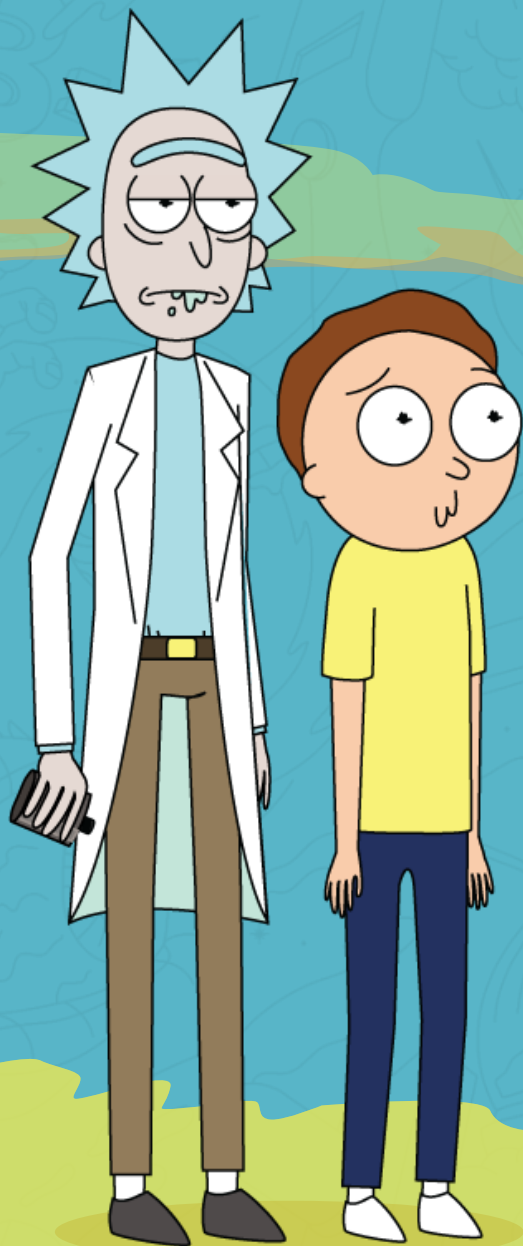


The background of the entire image is a dark, green-tinted space scene. Rick Sanchez and Morty Smith are depicted in a state of freefall, their bodies oriented upside down. Rick, with his characteristic spiky hair and lab coat, has a look of intense shock or panic. Morty, in his blue jumpsuit, also exhibits a similar expression. The title 'Rick and Morty' is superimposed over the center of the image in a large, blue, jagged font with a bright yellow outline, giving it a neon-like appearance. In the lower-left corner, a small, green, alien-like character with a large head and a single eye is visible. The overall aesthetic is a dark, monochromatic green with yellow highlights, creating a dramatic and chaotic atmosphere.

Rick and Morty

José David García Pineda - 192349
Michell Juliana Pérez Gómez - 192354
Euder Julián Pacheco Ascanio - 192356
Geraldine García Torrado - 192390



RICK AND MORTY GAME

Desarrollamos un videojuego inspirado en Rick and Morty, con batallas por turnos entre versiones alternativas de los personajes, integrando estrategia, trama y habilidades únicas. Empleamos una arquitectura multicapa para el mantenimiento y modularidad del código.

OBJETIVOS

- Aplicar POO para facilitar la ampliación y actualización del juego.
- Crear un sistema de combate estratégico e intuitivo.
- Incorporar un sistema de persistencia de datos eficiente.
- Implementar una arquitectura multicapa para modularidad y mantenimiento.
- Ofrecer una experiencia accesible con una interfaz gráfica simple.
- Mejorar el juego según la retroalimentación de los usuarios.



ALCANCE

El proyecto busca crear un juego funcional de combates por turnos con versiones alternativas de personajes y habilidades únicas. A futuro, se incluirán más personajes, modos de juego, multijugador, efectos visuales y sonoros, y seguridad de datos. Aspira a ser un estándar de excelencia en diseño y desarrollo de software.



ROLES



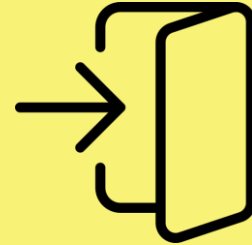
GERALDINE GARCÍA

Lógica del negocio



EUDER PACHECO

Persistencia



MICHELL PÉREZ

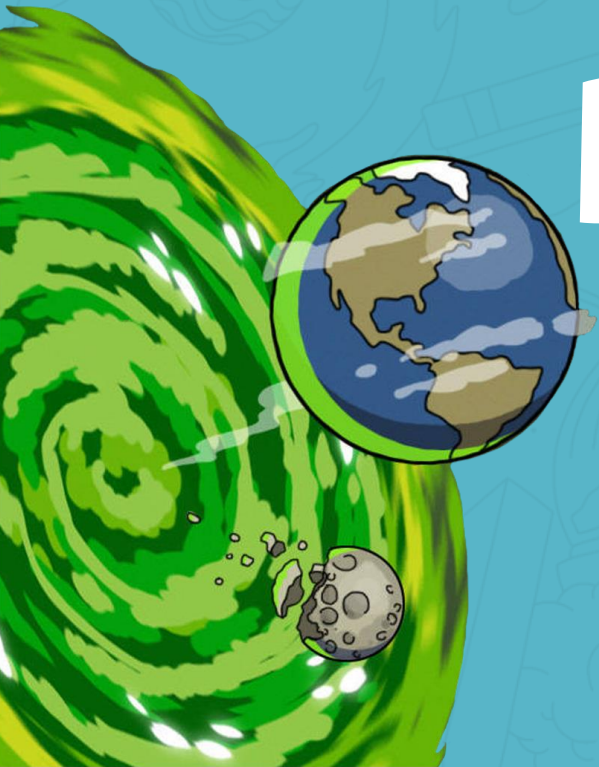
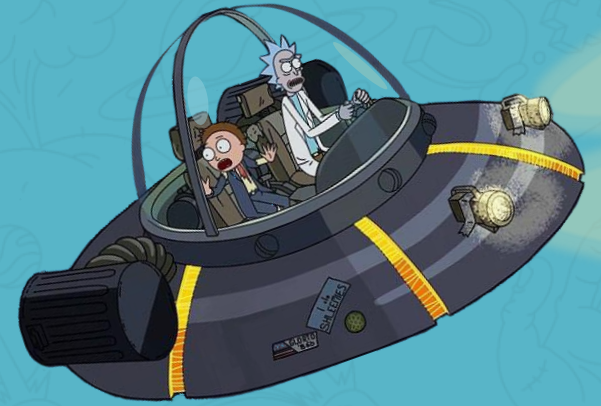
Acceso a datos



JOSÉ GARCÍA

Interfaz gráfica

MÓDULOS DESARROLLADOS

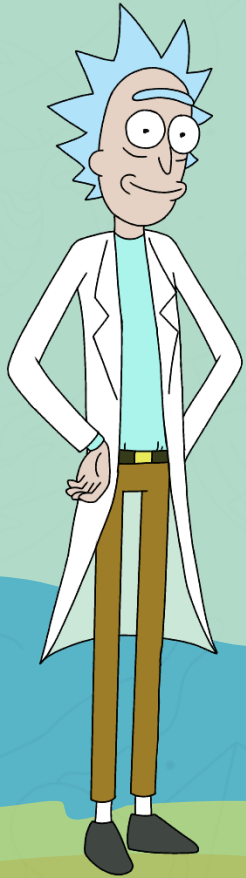


1. MÓDULO DE LA LÓGICA DEL NEGOCIO

Este módulo define las reglas del juego, controla las interacciones entre personajes y gestiona el flujo de la partida. Incluye:

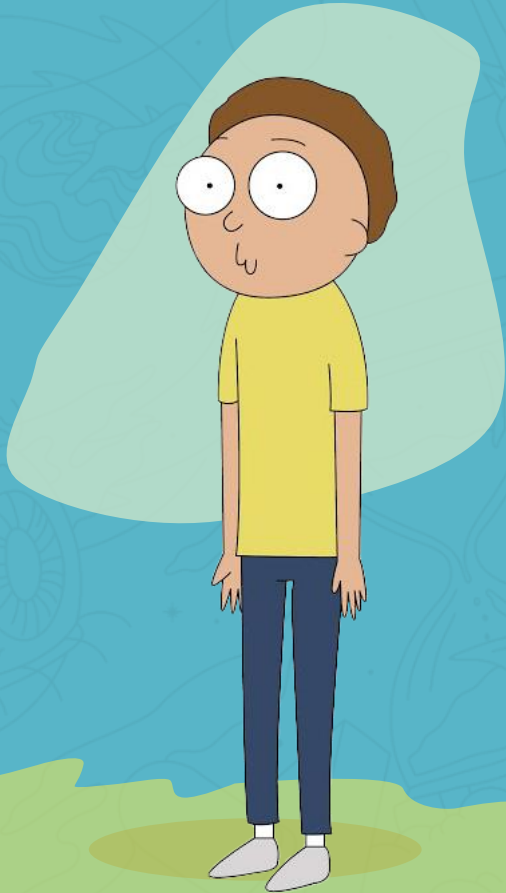
- Clase Character: Base abstracta con atributos comunes y métodos como attack, defend, heal y useSpecialAbility.
- Subclases Rick y Morty: Heredan de Character y sobrescriben useSpecialAbility con habilidades únicas.
- Clase Game: Gestiona la lógica del juego, los turnos, las acciones de los jugadores y las condiciones de victoria, asegurando un diseño modular y robusto.

2. MÓDULO DE PERSISTENCIA



Se utiliza un archivo JSON como base de datos para el videojuego, con dos categorías: Ricks y Mortys. Cada personaje tiene atributos como nombre, salud y poder. El archivo se usa para cargar los personajes, mostrar opciones de selección, definir estadísticas de combate y mantener una estructura organizada y fácil de actualizar.

3. MÓDULO DE ACCESO A DATOS



Este módulo lee y carga los datos de los personajes desde un archivo JSON, convirtiéndolos en objetos Java para su uso en el juego. Contiene:

- Clase DatabaseHandler: Utiliza la biblioteca Gson para cargar personajes desde characters.json y convertirlos en objetos Rick y Morty.
- Método loadCharacters: Lee el archivo y convierte los datos en listas de objetos organizados.
- Proporciona acceso a las listas de personajes para la capa lógica del juego.

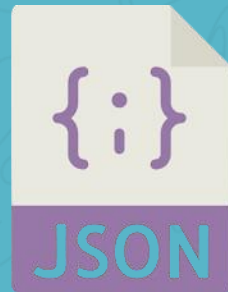
4. MÓDULO DE PRESENTACIÓN



El módulo proporciona una interfaz interactiva y funcional desarrollada con Java Swing, con pantallas de selección de personajes y batalla. Incluye elementos como:

- Paneles de estadísticas, registros de eventos, botones de acción, y un indicador de turno.
- Se utilizan CardLayout y ActionListener para gestionar las pantallas y las acciones, y las barras de progreso visualizan la salud de los personajes.
- Los personajes se cargan desde el DatabaseHandler integrado.

TECNOLOGÍAS USADAS



5. Mantenimiento

La fase de mantenimiento garantiza la calidad continua del juego, resolviendo errores, optimizando funcionalidades, actualizando la documentación y desarrollando nuevas características según la retroalimentación de los usuarios, manteniendo un registro de todos los cambios realizados.

4. Pruebas

En pruebas, se validan:

- Unitarias: Métodos individuales.
- Integración: Interacción entre módulos.
- Usabilidad: Facilidad de la interfaz.
- Rendimiento: Carga de datos desde JSON.

3. Implementación

Cada integrante desarrolla su módulo de forma paralela:

- Lógica del Juego: Clases base y flujo de combate.
- Persistencia: Elaboración del JSON y documentación.
- Acceso a Datos: Carga de datos desde JSON a objetos Java.
- Interfaz Gráfica: Diseño de la UI y eventos de botón.

SDLC

1. Análisis y planificación

En la planificación, se asignaron roles y responsabilidades al equipo, definiendo los objetivos y seleccionando las tecnologías adecuadas. En el análisis, se establecieron los requisitos del juego, y se identificaron riesgos con estrategias para mitigarlos.

2. Diseño

Se crea una estructura detallada con una arquitectura multicapa: presentación con Java Swing, lógica del juego y persistencia de datos. Se usan diagramas UML para representar las relaciones entre los componentes y se planifica la integración de la lógica y los datos con la UI.

DIAGRAMA DE CLASES

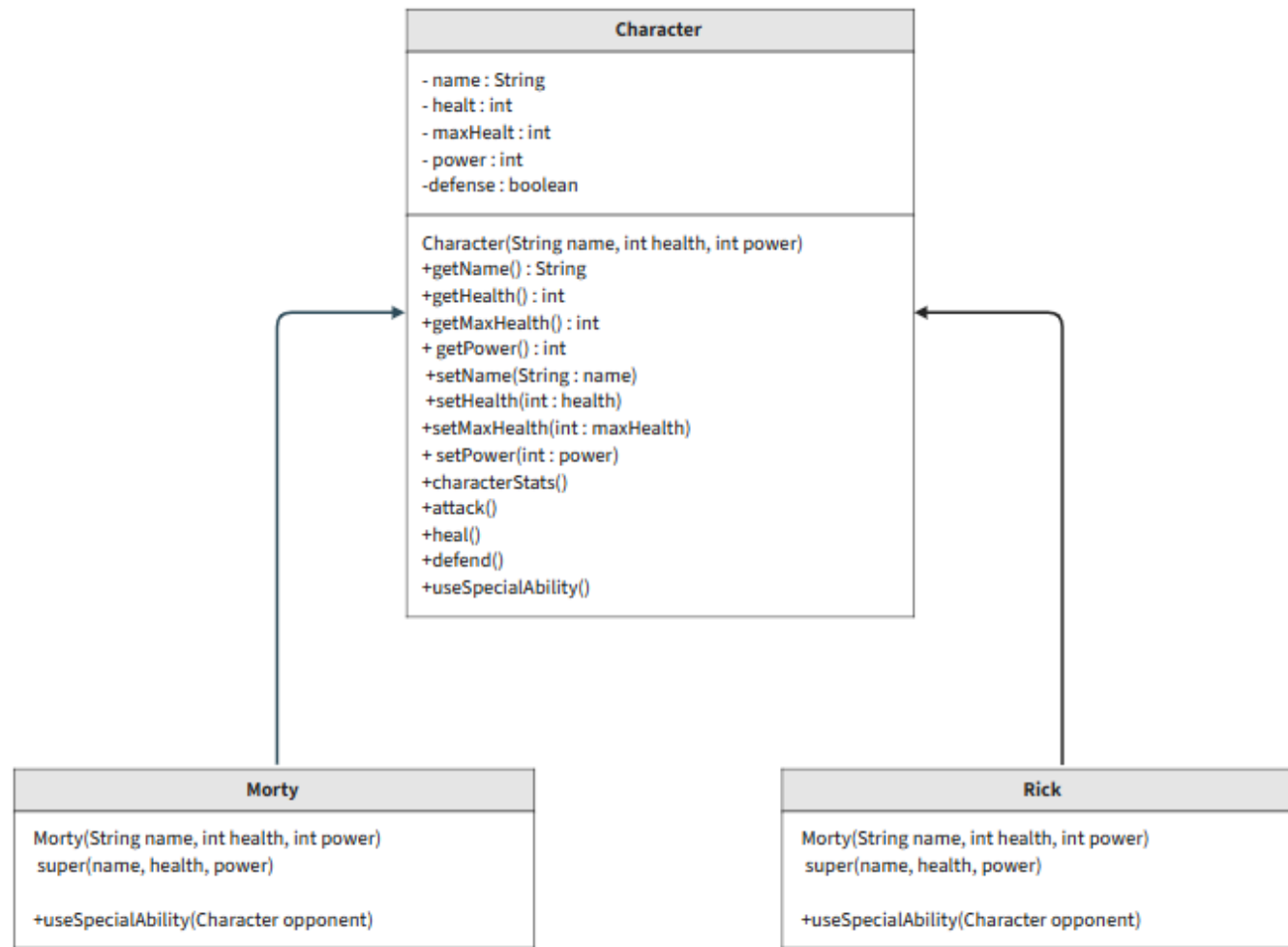
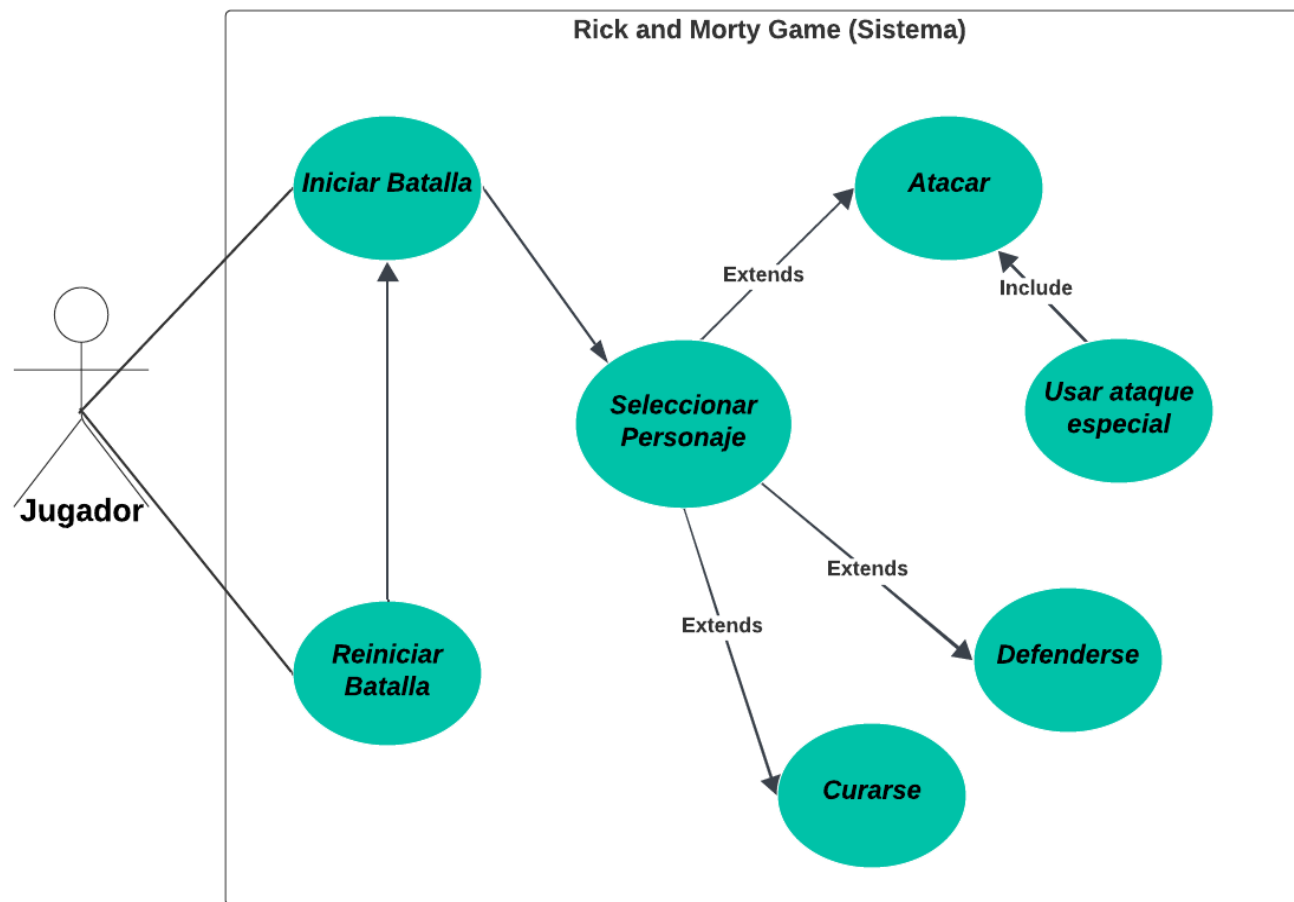


DIAGRAMA DE CASOS DE USO





THANK YOU