

# **Laboratorio de Computación III**

## **CRUD RPG**

**By: Ducamp, Gonzalez, Woloch**

# Resumen de la aplicación

CRUD RPG es un juego de rol cuya complejidad la centramos en el manejo de los datos (alta, baja, modificación y listado de los Usuarios, Personajes y Habilidades). El usuario puede crearse una cuenta y acceder a su personaje para así poder pelear contra monstruos que le dan experiencia y le permiten subir de nivel. La aplicación tiene persistencia de los datos mediante la escritura de archivos, por lo que el usuario puede cerrar la aplicación y no pierde su progreso, personajes, ni información personal.

La idea base que teníamos de CRUD RPG consistía en armar un juego con selección de personajes, con los cuales el usuario podía armar un equipo y así luchar contra otros grupos de personajes controlados por la máquina. Éstos iban a tener un sistema de items con consumibles y armas equipables que iban a modificar su daño, defensa u otros stats.

Debido al poco tiempo que teníamos para desarrollar decidimos acortar esta idea y tratar de hacer un juego simple de 1vs1, que tenga algunas clases distintas de Personajes para que sea más interactiva la app a la hora de usarla y enfocarnos más en el ABML de los datos más importantes como los de Usuarios, Personajes y Habilidades.

Primero armamos el UML de la app, una vez aprobado nos pusimos a codear. Nos mantuvimos comunicados mediante Discord, para solucionar dudas, compartir ideas e ir compartiendo los pull request creados, etc.

Decidimos usar Git y hacer code review entre todos para encontrarnos errores, ver mejoras y a su vez para que todos tengamos idea de cómo se iba armando la aplicación.

Comenzamos a codear las clases padres y algunas interfaces básicas, a medida que fuimos avanzando creamos nuevas interfaces y clases encargadas de solucionar algunos problemas que veníamos teniendo.

Como la clase DataManager, encargada de manejar los archivos y de tener las listas con los datos para que el Admin y el Menú puedan modificarlas / usarlas. O la interfaz de SystemMessage encargada de tener los mensajes de error a enviar con las Excepciones para no tener tanto código repetido.

También intentamos hacer una implementación de alguna librería Gráfica como Swing o JFrame, pero debido a la falta de tiempo no lo pudimos implementar. De todos modos dejamos un pull request con el test que realizamos.

## Informe técnico

Con esta aplicación quisimos aplicar todo lo aprendido en la materia en un ambiente de libre diseño. La elección de una temática RPG fue por simple gusto de los integrantes del grupo,

dado que el foco del proyecto estaba en la capacidad de desarrollar un ABML. Ya que consideramos que este nos permitirá integrar todos los conocimientos adquiridos durante el transcurso de la cursada.

Nuestro ABML consiste de usuarios que se dividen en Jugadores y Administradores. El rol de un administrador es gestionar el ABML, a excepción de los Personajes que cada Jugador posee, los cuales son gestionados por el mismo.

Existen 2 subclases de personajes, Personajes no jugables (Monstruos con los que combatirán los usuarios) y Personajes jugables (Que se subdividen en Clase Física y Clase Mágica). Estos últimos serán el tipo de Personaje que poseerá un Jugador al cual él mismo administra

A si mismo la Clase Física se subdivide en Guerrero y Cazador, mientras que la Clase Mágica se subdivide en Hechizero y Clérigo. La original de esta subdivisión era la limitación en el tipo de armas que podían poseer, pero aún no se han implementado las clases de Objetos (Armas, Armadura, Consumibles) por falta de tiempo.

Por otro lado están las Habilidades usadas para ataques más potentes. Estas se subdividen en Habilidades Físicas (Sólo utilizables por las Clases Físicas) y Hechizos (Solo utilizables por las Clases Mágicas).

Por más de que no se hayan completado todas las ideas de desarrollo que teníamos, llegamos a aplicar todo lo visto en la cátedra: Diseño de clases, Genéricos, Colecciones, Excepciones, Archivos, Interfaces, Clases abstractas, etc, y generamos conocimientos sobre estos temas.

## Manual de Usuario

Menues:

Menú principal, donde inicia el programa, le dará las siguientes opciones:

1. Create account (Crear Cuenta)
  - a. Aquí pedirán los datos para crear una nueva cuenta (Comprobando que el nombre de usuario no haya sido tomado), y la creara como Jugador, las cuentas de Administradores, serán definidas por código.
2. Login (Iniciar Sesión)
  - a. Aqui pedira el Username, y comprobara que este en el archivo antes de dejarlo poner la contraseña. Dependiendo si es una cuenta Jugador o Administrador, lo mandara a distintos menues.
0. Exit(Salir)

Menú de Jugador, donde se accede luego de iniciar sesión con una cuenta Jugador, le dará las siguientes opciones basadas en sí tiene un Personaje Jugable asignado, o no.

Si no tiene un Personaje Jugable asignado:

1. Create a Character. (Crear un personaje)
  - a. Aquí hara que ingrese el nombre del personaje, y dará a elegir entre las clases:
    - i. Cleric (Clérigo)
    - ii. Mage (Mago)
    - iii. Hunter (Cazador)
    - iv. Warrior (Guerrero)
0. Exit(Salir, en este caso cerrara la sesion)

Si tiene un Personaje Jugable asignado:

1. View Character (Ver personaje)
  - a. Donde se mostrará los datos del personaje.
2. Modify Character (Modificar personaje)
  - a. Donde se cambiará algún atributo del mismo.
3. Fight Some Monsters (Pelear con algunos Monstruos)
  - a. Donde se pasará al menú de la parte jugable.
0. Exit(Que lo llevará al menú anterior)

Menú de Combate, donde se accedera a la parte jugable del programa, tendrá las siguientes opciones:

1. Fight a monster (Pelear con un monstruo)
  - a. Donde se hará un combate con un monstruo turno a turno, en el que si nuestro personaje sale victorioso, ganará experiencia, y si no, no ganara nada.
0. Exit(Que lo llevará al menú anterior)

## Bitácora

Semana 31/05 - 7/06

- Decisión de idea.
- Creación de distintos UMLs y puesta en común para la definición del UML final.
- Decisión de temática RPG y enfoque en ABML con sistema básico de combate. (Reestructuración de UML)
- Seteo de git local y de repositorio en GitHub.
- Creación de tickets a trabajar.
- Separación de algunas tareas tareas tareas (asignaciones).
- Creación de proyecto base.

Semana 7/06 - 14/06

- Creación de clases base: User, Character, Ability.
- Creación de algunas clases hijas: NonPlayableCharacter, Player, PlayableCharacter.
- Creación de la clase Main.

Semana 14/06 - 21/06

- Implementación de manejo de errores con if / SOUT.
- Creacion clases: PhysicalSkill, Spell, Hunter, Warrior, .
- Creación de Combat interface.

Semana 21/06 - 28/06

- Creación clase genérica CRUD (ABML).
- Creación Interface SystemMessage para comunicar mensajes de error.
- Implementación de manejo de errores: try-catch / throws Exceptions.
- Creación de clases: Cleric, Mage.

Semana 28/06 - 2/07

- Implementación de Archivos al Proyecto.
- Test de implementación de JFrame.
- Creación de clase DataManager encargada de manejar datos.
- Finalización y corrección de las clases restantes.
- Armado del Informe.
- Limpieza de UML.

## UML, Trello & Repositorio

**Trello:** <https://trello.com/b/KK49NOqM/ducamp-gonzalez-woloch>

**UML:** [https://www.draw.io/#G1SI7chdaGwr5ghxq\\_RELusz3d8ludUHr](https://www.draw.io/#G1SI7chdaGwr5ghxq_RELusz3d8ludUHr)

**Repositorio:** <https://github.com/AlejandroWoloch/RPGABML>

## Bibliografía consultada

<https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

<https://docs.oracle.com/javase/tutorial/essential/exceptions/index.html>

<https://es.wikipedia.org/wiki/CRUD>

[https://www.w3schools.com/java/java\\_hashmap.asp](https://www.w3schools.com/java/java_hashmap.asp)

[https://www.w3schools.com/java/java\\_enums.asp](https://www.w3schools.com/java/java_enums.asp)

<https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html>

[https://www.tutorialspoint.com/java/java\\_generics.htm](https://www.tutorialspoint.com/java/java_generics.htm)

<https://docs.oracle.com/javase/7/docs/api/java/text/MessageFormat.html>

<https://www.javatpoint.com/java-swing>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>

<https://www.youtube.com/channel/UCEUdgJSqHi-mwFLprIIWoXA>