

# **“Manual Técnico”**

## **Presentación**

El presente documento es el manual Técnico del Juego de Combate en consola que se presenta como parte de la Practica 1. Este juego es un RPG (role playing game) de combate en el que el jugador cuenta con 3 vehículos para atacar a enemigos sobre un escenario. Crear Vehículos o escenarios según le parezca al jugador siempre y cuando vaya desbloqueando estas opciones a través de la experiencia y puntaje en el juego la cual se obtendrá de acuerdo a sus capacidades y tiempo de uso de la consola.

El juego está diseñado en lenguaje JAVA y es reproducible (.jar) por medio de cualquier IDEA con la capacidad de interpretar este lenguaje de programación, así mismo se incluye el funcionamiento básico del mismo y una guía para su primera ejecución y posteriormente el uso del mismo.

## **Objetivos del Programa**

### Generales

- Familiarizar al estudiante con el lenguaje Java.
- Aplicar conceptos de programación orientada a objetos recibidos en clase magistral y laboratorio.
- Elaborar la lógica para la solución del problema planteado.

### Específicos

- Construcción de algoritmos para los requerimientos de la actividad.
- Ampliar el conocimiento del lenguaje JAVA.
- Ampliar el conocimiento de Programación orientada a objetos en JAVA.
- Desarrollar diagramas de clase como parte del análisis del problema.
- Implementación de ciclos, sentencias de control y arreglos.
- Implementación de clases, herencia y polimorfismo.
- Construcción de aplicaciones en consola.
- Desarrollo de manual técnico y de usuario.

## **Instrucciones de Programa**

1. Ejecutar Main
2. Crear Nickname
3. Elegir Nickname
4. Jugar
5. Elegir escenario
6. Continuar nivel

7. El personaje posee puntos de vida (HP) y puntos de poder (PP).
8. Si los puntos de vida llegan a cero durante una batalla, muere el personaje.
9. Los puntos de poder se utilizan durante la batalla realizar un ataque especial.
10. El objetivo es destruir todos los vehículos del enemigo (finalizar batalla)
11. Se realiza Landon de inicio
12. Se sigue el orden lógico por medio de turnos hasta finalizar batallas

## **Diagrama de Clases**

**FALTA EL DIAGRAMA DE CLASES**

## **Requerimientos Técnicos**

- SO (Linux o Windows)
- IntelliJ IDEA o cualquier editor de lenguaje JAVA
- .Jar ejecutable o proyecto completo
- Entender lenguaje JAVA (para ver creación de programa)
- Experiencia previa en Juegos RPG

## **Herramientas de Construcción**

- IntelliJ IDEA
- Package de JAVA
- Clases de JAVA
- Métodos creados en JAVA
- Arreglos de JAVA
- PC funcional
- Conocimientos de Programación.

## **Como Acceder al Programa**

Se debe tener el proyecto completo luego se carga a la terminal por medio de IntelliJ o cualquier IDEA con la capacidad de interpretar el lenguaje JAVA. Una vez cargado el código correspondiente al Juego RPG se ubica la carpeta Main y se compila el código a través de la opción de IntelliJ.

JuegoDamas [C:\Users\luedu\IdeaProjects\JuegoDamas] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

New ▶ any > CasillaBlanca

Open...

Open Recent ▶

Close Project

Settings... Ctrl+Alt+S

Project Structure... Ctrl+Alt+Mayús+S

Other Settings ▶

Import Settings...

Export Settings...

Settings Repository...

Export Project to Eclipse...

Export to Zip File...

Save All Ctrl+S

Synchronize Ctrl+Alt+Y

Invalidate Caches / Restart...

Print...

Associate with File Type...

Line Separators ▶

Power Save Mode

Exit

Practica1

Juegos1IPC

ProjectObj

ElSeñorAnillos

IdeaProjects

SeñorAnillos

ClaseDamas

ProjectObjj

ProyectoObj

ProyectoObj

ProyectoObj

Clase1502

JuegosIPC

Oliver

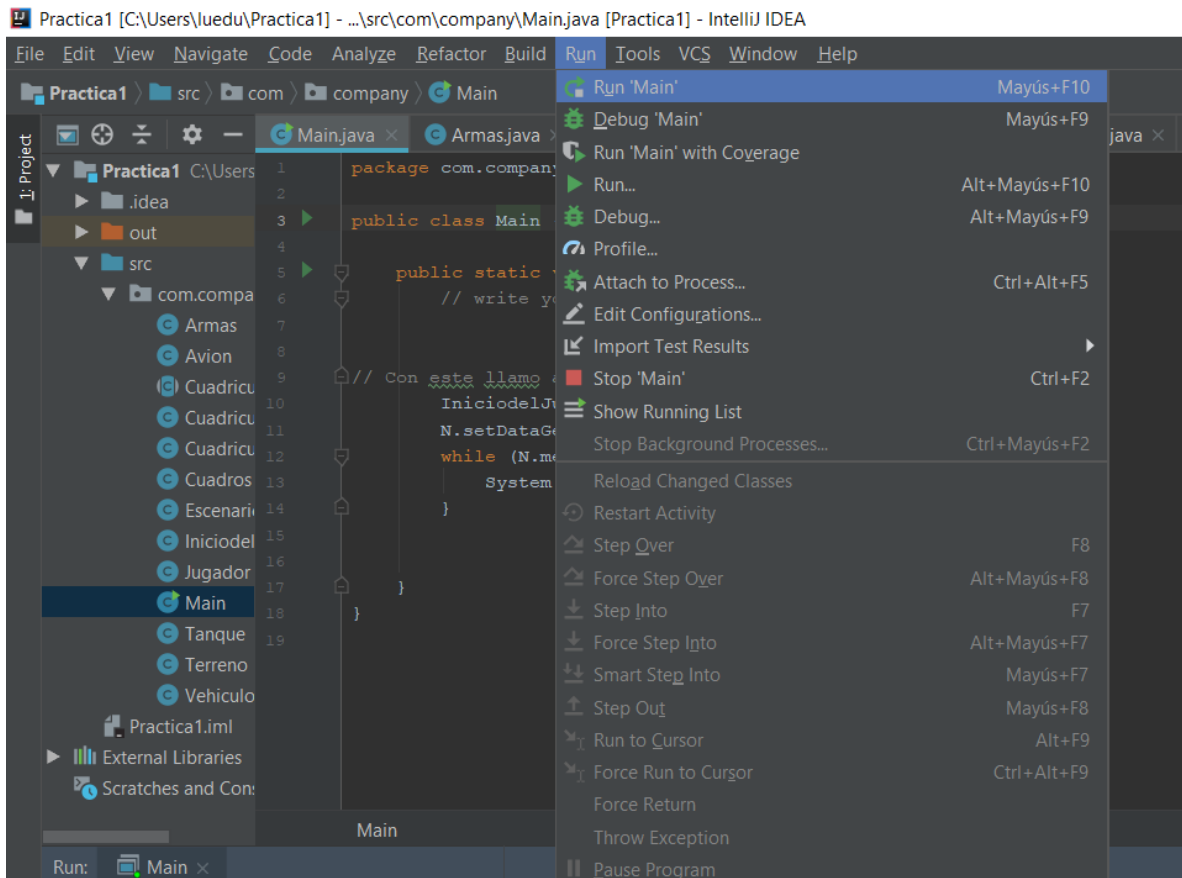
Juegoppt

Oliver1

HolaMundo

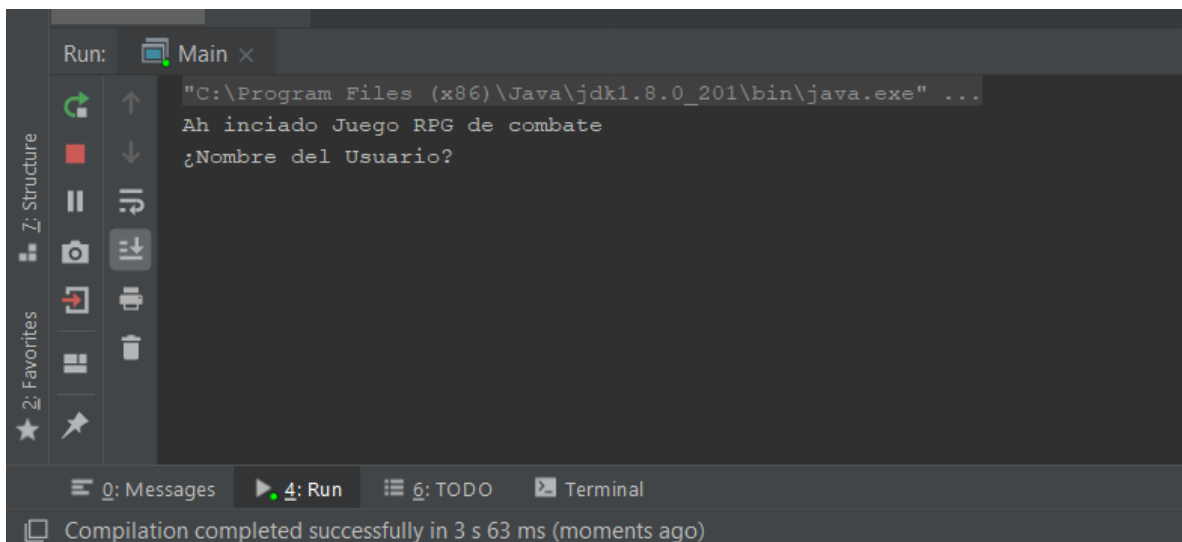
untitled

Manage Projects...

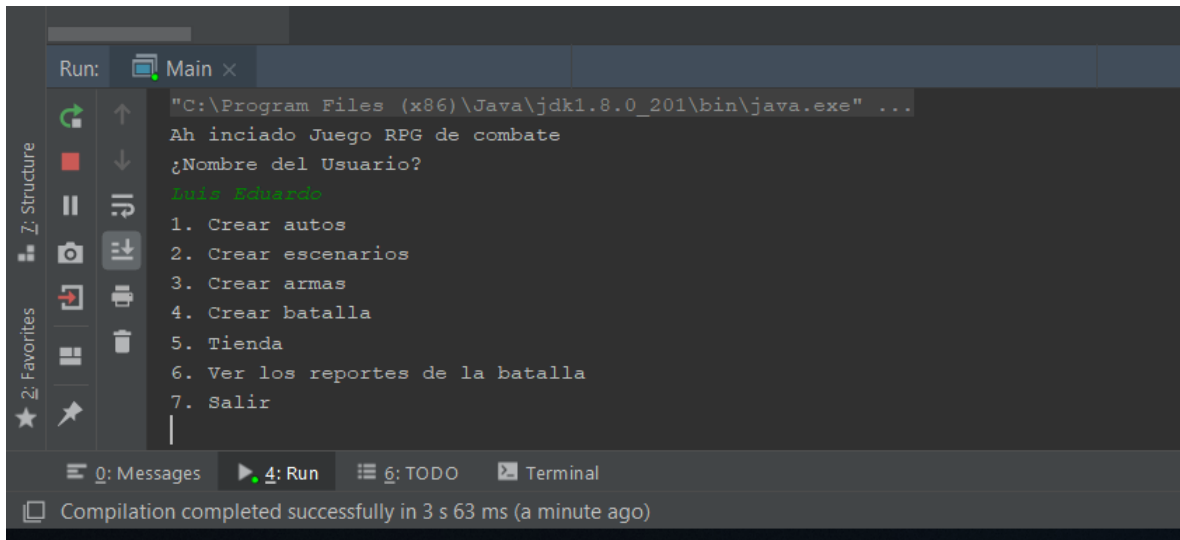


## Descripción del Programa

### Captura Ejecutar Programa



## Descripción de Funciones de Juego

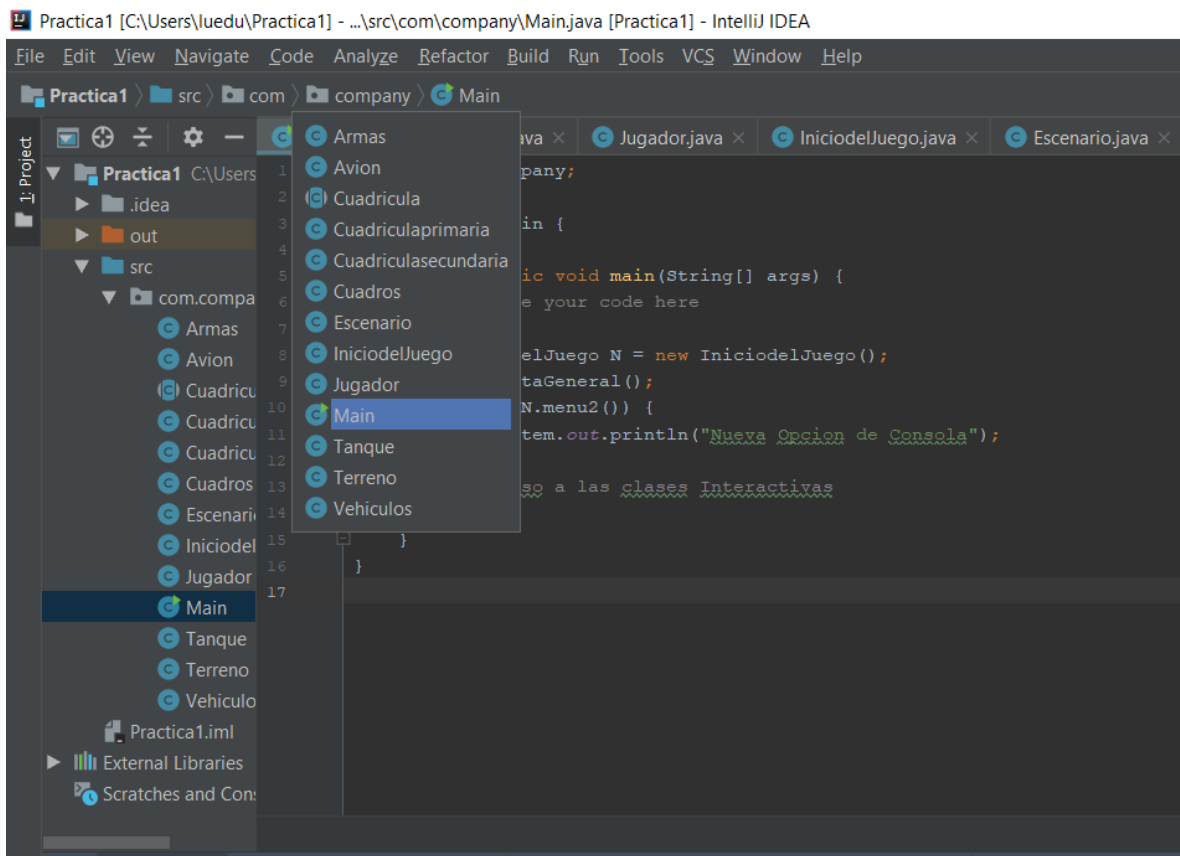


```
Run: Main x
"C:\Program Files (x86)\Java\jdk1.8.0_201\bin\java.exe" ...
Ah iniciado Juego RPG de combate
¿Nombre del Usuario?
Luis Eduardo
1. Crear autos
2. Crear escenarios
3. Crear armas
4. Crear batalla
5. Tienda
6. Ver los reportes de la batalla
7. Salir
|
```

0: Messages 4: Run 6: TODO Terminal

Compilation completed successfully in 3 s 63 ms (a minute ago)

## Bloques de Clases



Practica1 [C:\Users\luedu\Practica1] - ...src\com\company\Main.java [Practica1] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Practica1 > src > com > company > Main

Armas  
Avion  
Cuadrícula  
Cuadrícula primaria  
Cuadrícula secundaria  
Cuadros  
Escenario  
Inicial del Juego  
Jugador  
Main  
Tanque  
Terreno  
Vehículos

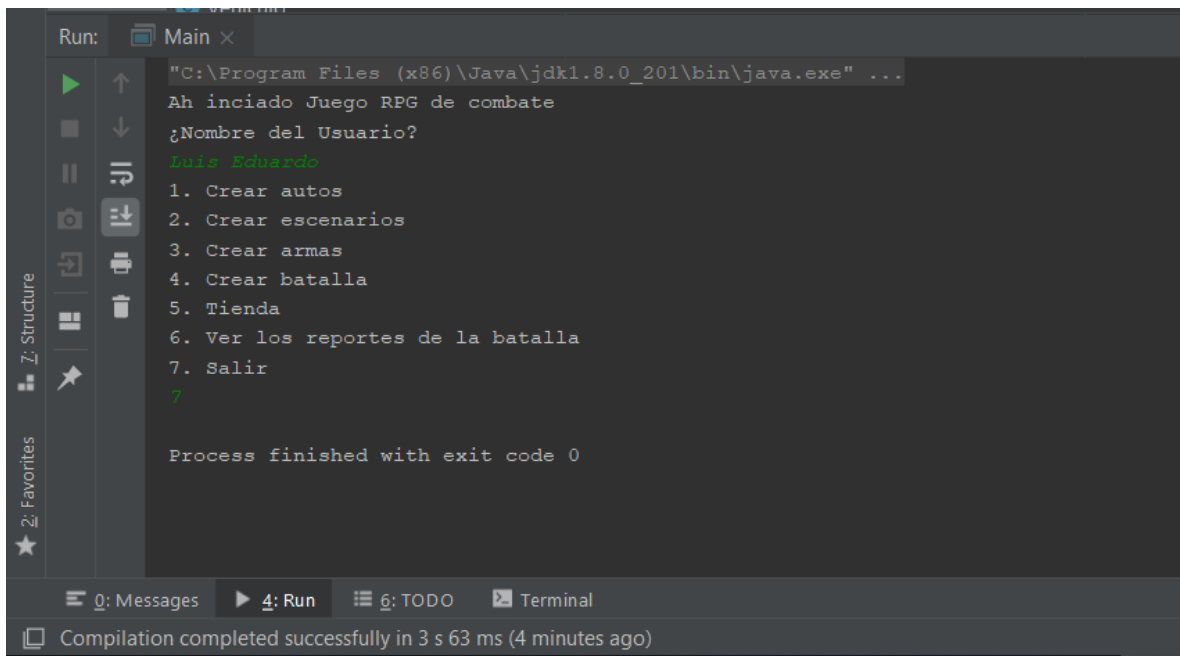
```
1 package com.company;
2
3 import java.util.Scanner;
4
5 public class Main {
6     public static void main(String[] args) {
7         // TODO: your code here
8
9         Inicial del Juego N = new Inicial del Juego();
10        N.menu2();
11        System.out.println("Nueva Opcion de Consola");
12
13        // TODO: your code here
14
15        // TODO: your code here
16    }
17 }
```

Iniciar Juego

Inicio de Juego

## Finalizar Juego

### Salir del Programa



```
Run: Main x
"C:\Program Files (x86)\Java\jdk1.8.0_201\bin\java.exe" ...
Ah iniciado Juego RPG de combate
¿Nombre del Usuario?
Eduardo
1. Crear autos
2. Crear escenarios
3. Crear armas
4. Crear batalla
5. Tienda
6. Ver los reportes de la batalla
7. Salir
7

Process finished with exit code 0

Q: Messages 4: Run 6: TODO Terminal
Compilation completed successfully in 3 s 63 ms (4 minutes ago)
```

### Ver registros

- Listado de vehículos: Por cada vehículo desplegar en pantalla el nombre, imagen y el estado (activo o destruido). Incluir también la cantidad de enemigos destruidos por el vehículo y la cantidad de veces que el vehículo ha sido destruido. El listado debe poder ordenarse de forma ascendente o descendente basado en la cantidad de enemigos destruidos por cada vehículo.
- Reporte de batallas: Muestra la cantidad total de batallas, indicando la cantidad de batallas perdidas y ganadas.
- Reporte de mejor vehículo: Muestra el vehículo que más enemigos a destruido, mostrar también el listado de todos los enemigos, incluyendo el escenario donde se destruyó al enemigo.
- Reporte de peor vehículo: Muestra el vehículo que más se ha destruido por enemigos, mostrar también el listado de todos los enemigos, incluyendo el escenario donde se ha destruido al vehículo.

## Juego

En el primer nivel se le solicitara un Nickname que debe ser el cual use siempre que desee jugar (esto para las estadísticas), la experiencia es de 0, se poseerá un arma básica, y se crearán 3 vehículos los cuales servirán para el combate posterior.

Un turno se compone de una acción del jugador y acciones de los enemigos, ejecutando primero la acción de quien inició el combate (random), la arena donde se pelee esta compuesta por valles, montañas y agua esto con el fin de poner restricciones a los vehículos. Se permite atacar, cambiar de vehículo, Utilizar Items y Rendirse. El juego se dará por finalizado al perder todos los vehículos o salir de manera rápida (rendirse)

### **Glosario (tecnicismos)**

Java.Util: Clases de utilidades para el lenguaje, como acceso a recursos del sistema.

Import: Para importar clases de un paquete o una librería se usa el comando import.

Public static void: Clase estática para usarla durante el código (lo que lee primero).

Boolean: Boolean es un tipo de variable que sólo tiene dos valores posibles: "true" (verdadero, 1) y "false" (falso, 0). (la cual es en base a una pregunta).

While: (condición) {instrucciones a ejecutarse} donde condición es una expresión que da un resultado true o false en base al cual el bucle se ejecuta o no.

Int: para declarar una variable y luego darle valor (se asigna la letra y un entero).

Break: Se usa para salir rápidamente de un ciclo. Mientras se ejecuta una iteración de un ciclo, si se llega a ejecutar la instrucción break; se abandona de inmediato esa iteración y se sale del ciclo.

Sacanner: buscar lo que devuelve el usuario.

Final int: Constante que se le asigna a una variable específica.