



## **Anteproyecto**

Fecha entrega: 27/04/2015  
Autores: Rafael Navarro Prieto

# Introducción

Como idea de anteproyecto pretendo realizar un juego que te permita crear unidades con unas estadísticas fijas (especificadas según una facción) y unas estadísticas variables elegidas por el usuario. Una vez se creen las unidades se podrán probar la unidad para comprobar el daño que realiza al atacar.

## ArrayList

Para cubrir el tema de los arraylist, las unidades creadas por los usuarios se almacenaran en un arraylist. Este arraylist deberá de estar parametrizado para que todas las unidades puedan almacenarse (herencia mas adelante).

## Enumeraciones

Se creara una enumeración tipo que contenga los tres tipos de unidades que se pueden crear:

1. Ligera
2. Mediana
3. Pesada

Ambas enumeraciones tendrán como parámetros vida y ataque. Estos parámetros serán una base que posteriormente al crear una unidad se le aplicara una interfaz creacionable.

## Herencia

La herencia tendrá la siguiente estructura:

La clase padre de todas sera Unidad, esta clase tendrá como atributos:

1. Nombre → nombre único de la unidad controlado con un patrón  $^{([a-z]^+)}\{3,\}\$$  solo permitirá nombres como mínimo con 3 caracteres, números no incluidos.
2. Victorias → número de victorias de la unidad
3. Derrotas → número de derrotas de la unidad
4. Vida → vida de la unidad
5. Ataque → indicador de fuerza de la unidad
6. Tipo → variable que almacena un valor de una enum indicando 1 de las 3 posibles facciones.
7. Aciertos → Contador que deshabilitara y habilitara la opción de realizar ataques mas fuertes.
8. Fecha de creación → fecha que indica cuando se creo la unidad.

De la clase Unidad tendremos los diferentes tipos de unidades según una raza:

1. Protos:

Escudo → vida complementaria que debe de estar a 0 para poder perder vida

Regeneración escudo → cantidad de escudo que gana por turno

Regenerable → campo booleano que controle si puede regenerar la unidad escudo o no

2. Terran:

HabUtilizada → campo booleano que comprueba si la habilidad ya fue usada o no

Método stimpack → método exclusivo de la raza que le permite duplicar su ataque a costa de un 50% de la vida actual que tenga en el combate.

### 3. Zerg:

Regeneración vida → cantidad de vida que ganara por turno

## Interfaces

Tendremos una interfaz creacionable que contendrá los siguientes métodos

1. CalcularVida → calculara la vida final de cada unidad dependiendo del tipo de raza y el tipo de unidad.
2. CalcularAtaque → calcula el ataque de cada unidad dependiendo del tipo de raza y el tipo de unidad.

## Flujo de datos y ficheros

Para el tema de ficheros y flujos se permitirá exportar e importar una lista que contenga unidades creadas previamente, el fichero que almacenara esa lista tendrá que tener una extensión específica que se controlara con un patrón en concreto.

## Excepciones

Todo problema que surja a la hora de crear una unidad o al cargar y guardar archivos se lanzara una correspondiente excepción y se controlara con una ventana emergente que mostrara el error causado.

## Fechas

Siempre que se arranque el programa se mostrara en una parte del programa la fecha actual. También habrá un campo en las unidades donde se vera cuando se creo la unidad.

## Gui

Para el interfaz gráfico tendremos un panel principal a través del cual podremos cargar una lista de unidades ya creada previamente, guardarla, acceder al panel de creación de unidades.

Ademas tendrá una serie de funcionalidades:

1. Añadir unidades
2. Eliminar unidades
3. Búsqueda de unidades de 3 formas diferentes:
  1. Búsqueda colectiva (muestra toda la lista de unidades)
  2. Búsqueda por raza
  3. Búsqueda por tipo de unidad
4. Dentro de la búsqueda habrá una opción que permita a los usuarios probar la efectividad de esas unidad en combate. Esta ventana constara de la visualización de las estadísticas de la unidad y un historial donde se visualizara el daño realizado por la unidad si el ataque tiene éxito, en caso contrario, el ataque habrá fallado.

## Para subir nota

Se creara una interfaz combatible que contendrá los tres posibles ataques que pueden realizar todas las unidades:

- Un ataque básico
- Un ataque medio
- Un ataque avanzado

Los ataques se gestionaran en cada raza creándole una posibilidad de fallar el ataque para evitar que siempre se acierte un ataque y que exista un equilibrio entre las unidades.

Se añadirá una nueva ventana para probar las unidades. Cuando las unidades estén en combate se dividirá la ventana en dos paneles donde aparecerá la imagen de cada raza, vida y escudo de la unidad y en la parte inferior las diferentes acciones que podrá realizar. En la parte inferior aparecerán mensajes indicando si la unidad realizo daño o esquivo y el ganador.

Otra de las funcionalidades de los arraylist sera la visualización de un ranking mostrando cuales son las unidades que han conseguido mas victorias y cuales tienen un mayor número de derrotas.