

Memoria: Fase de diseño

Metodologías de la Programación



Universidad
Rey Juan Carlos

Gonzalo Pérez Roca
Alba Velasco Marqués
Javier Bringas García
Paula Ruiz Rubio

Índice

Objetivo.....	3
Diagrama de clases.....	3
Clases implicadas en la aplicación	3
Diagrama de secuencia	4
Mecánica de desafíos entre jugadores	4
Dinámica del combate (más detallada)	5
Diagrama de estados	6
Desafío desde el punto de vista del usuario	6
Desafío desde el punto de vista del operador.....	6
Finalización de un combate.....	7
Diagramas de caso de uso.....	7
Caso de uso del usuario	7
Caso de uso del operador	8

Objetivo

En este documento, se mostrarán los diagramas de clases, secuencia, estados y casos de uso que describen el funcionamiento de la aplicación que en la siguiente fase codificaremos.

Diagrama de clases

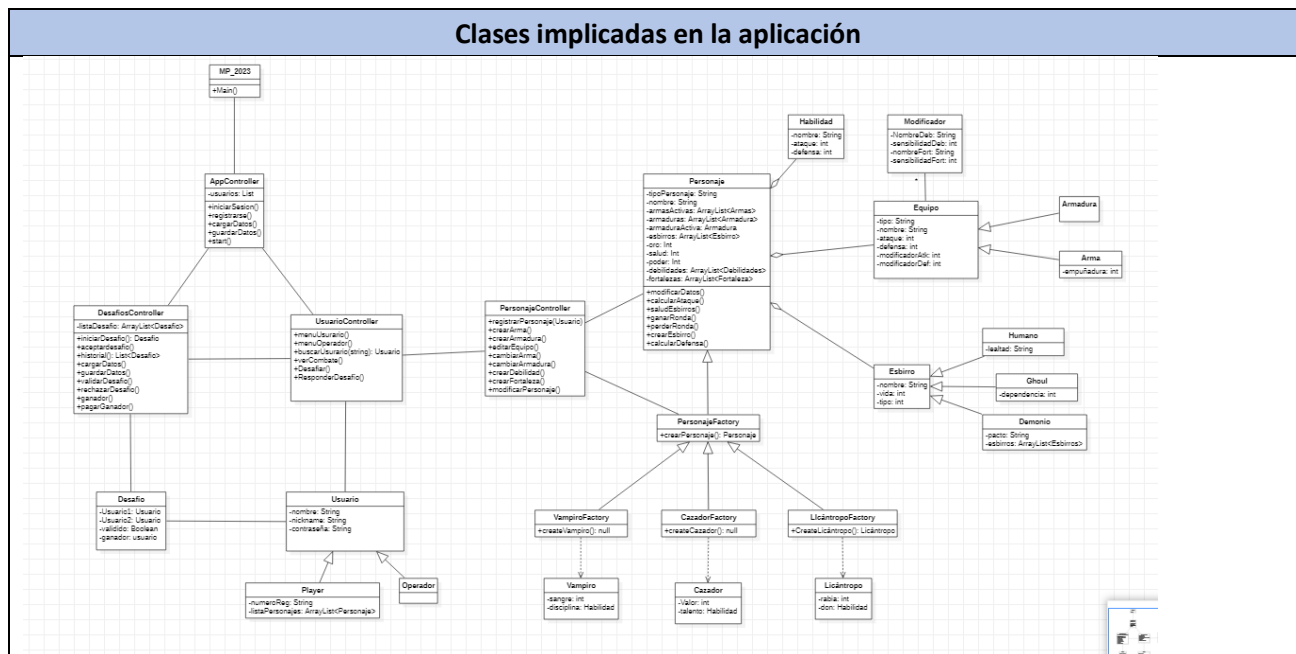


Diagrama de clases 1

Asumimos que toda propiedad privada lleva un get y un set asociado, así como los respectivos métodos constructores por cada clase.

Para diseñar la aplicación se han utilizado las siguientes clases:

Para definir los usuarios tenemos:

- **UsuarioController**: será la clase que contenga la operativa relacionada a los diversos usuarios.
- **Usuario**: clase que almacena los atributos de los usuarios y de la que se distinguen dos tipos Player(jugadores) y operadores del sistema.

Para la lógica de los desafíos tenemos:

- **DesafiosController**: Almacenará una lista con los desafíos y albergará todas las operaciones que se puedan realizar en un desafío.
- **Desafío**: Contiene los atributos de un desafío, tales como los usuarios que participan y la validez de este.

Para los personajes:

Dentro de los personajes hemos optado por utilizar el patrón **factory method** para generalizar la creación de estos mediante una fábrica general que llama a las diversas fábricas específicas.

- **Personaje**: Alberga los atributos de los personajes y algunas de sus funcionalidades básicas, de ella heredan los tres personajes específicos Vampiro, Cazador y Licántropo.
- **PersonajeController**: Ocupa la lógica compleja de la clase personaje.
- **Esbirro**: Contiene los atributos propios a todos los esbirros y de ella heredan los diferentes tipos de esbirros Humano, Demonio y ghoull. Se relaciona con la clase personaje, que alberga una lista de esbirros.
- **Habilidad**: Contiene los atributos de las habilidades. Se relaciona con la clase personaje.

- **Equipo:** Contiene los atributos propios al equipamiento, de ella heredan las clases Armadura y Arma.
- **Modificador:** Contiene los atributos de los modificadores, que irán asociados a los atributos del equipo.

Diagrama de secuencia

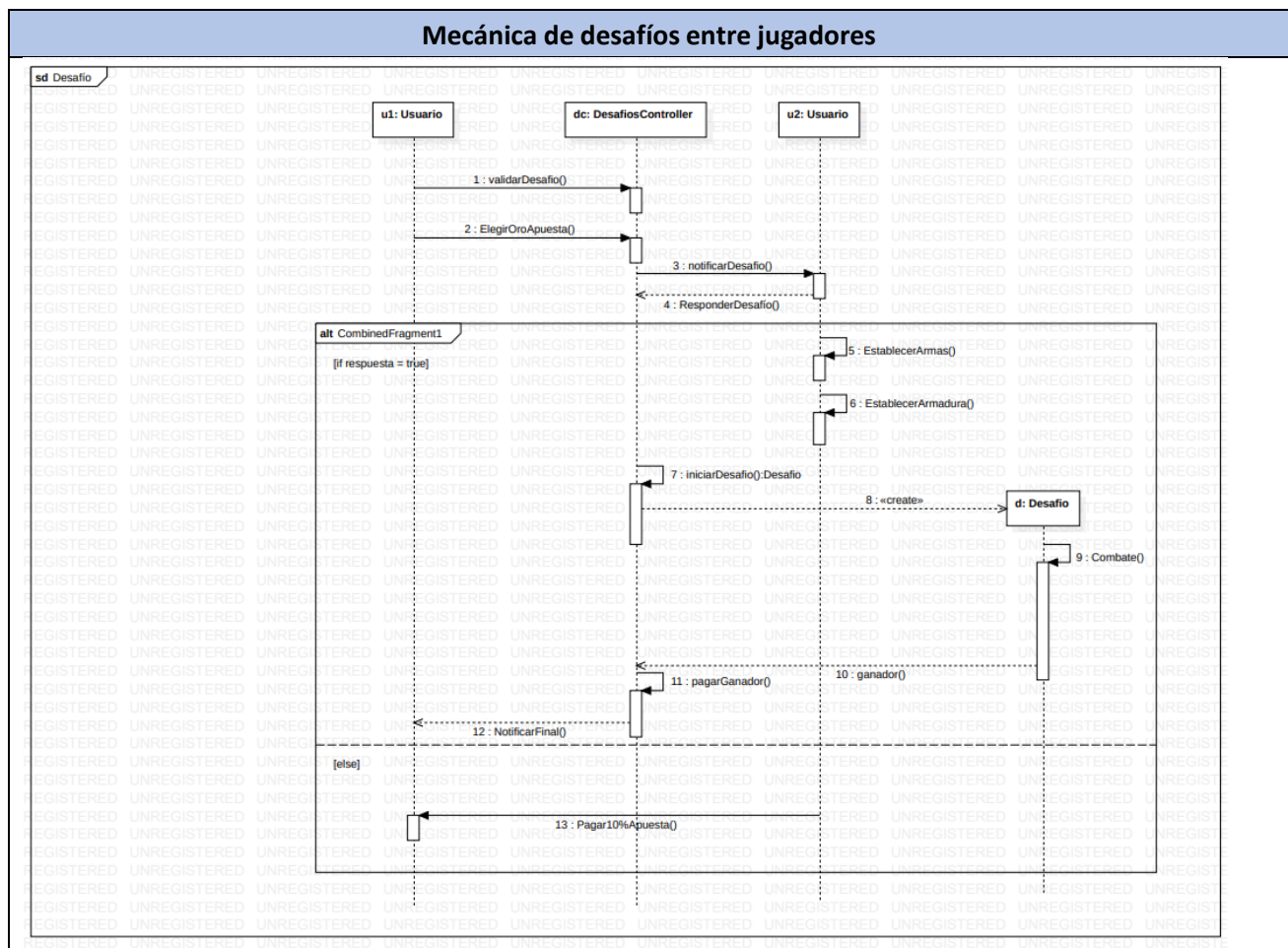


Diagrama de secuencia 1

En este diagrama se plasma el siguiente orden de acciones: lo primero se debe validar la solicitud de desafío y seleccionar el oro que el desafiante quiere apostar, una vez validado el desafío, el desafiado establece sus armas y armaduras y se inicia el combate.

Una vez terminado el combate se paga al ganador la cantidad de oro apostada y se envía un registro de todo el combate al desafiante.

En el caso de que el desafiado rechace el desafío, debe pagar el 10% de la cantidad apostada.

Dinámica del combate (más detallada)

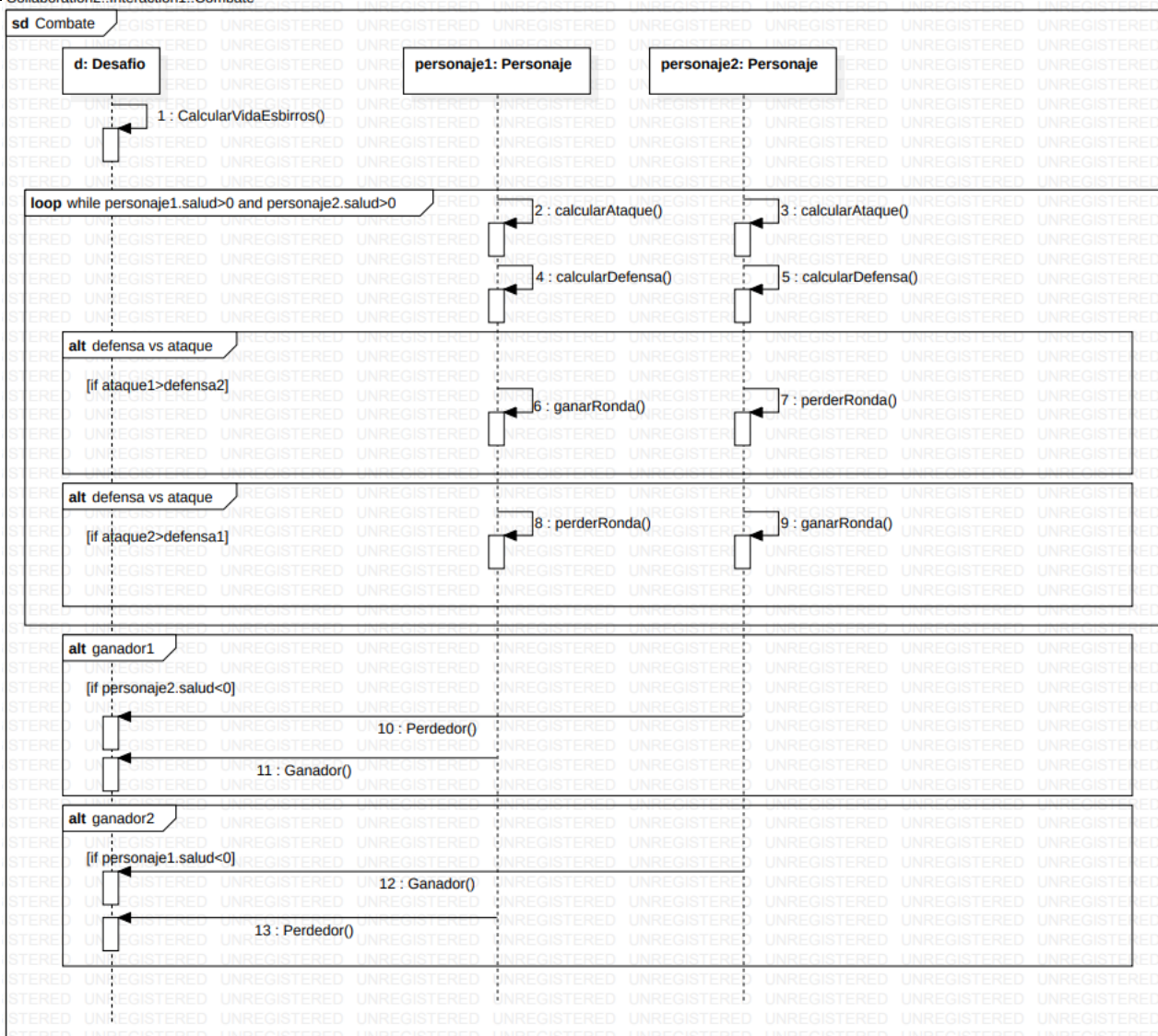


Diagrama de secuencia 2

Este combate, se mantiene un bucle de turnos hasta que uno de los combatientes, sino los dos, pierdan todos los puntos de vida. En cada iteración se calculan los puntos de ataque y defensa de cada contrincante y se comparan respectivamente (el ataque de un participante con la defensa del opuesto) decidiendo el vencedor de cada ronda, el vencido resta un punto de vida. Una vez se sale del bucle, se comunica el ganador y perdedor.

Diagrama de estados

Desafío desde el punto de vista del usuario

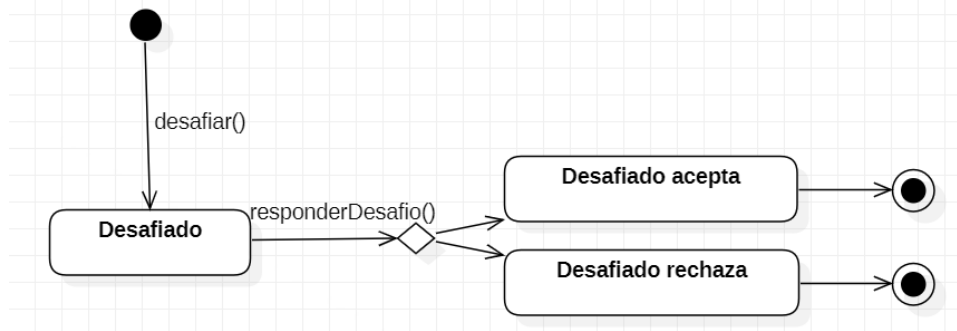


Diagrama estados 1

Un usuario desafía a otro, y una vez validado ese desafío (diagrama estados 2) el usuario desafiado puede aceptarlo o rechazarlo.

Eso es lo que se ilustra en este diagrama como el usuario acepta o rechaza un desafío.

Desafío desde el punto de vista del operador

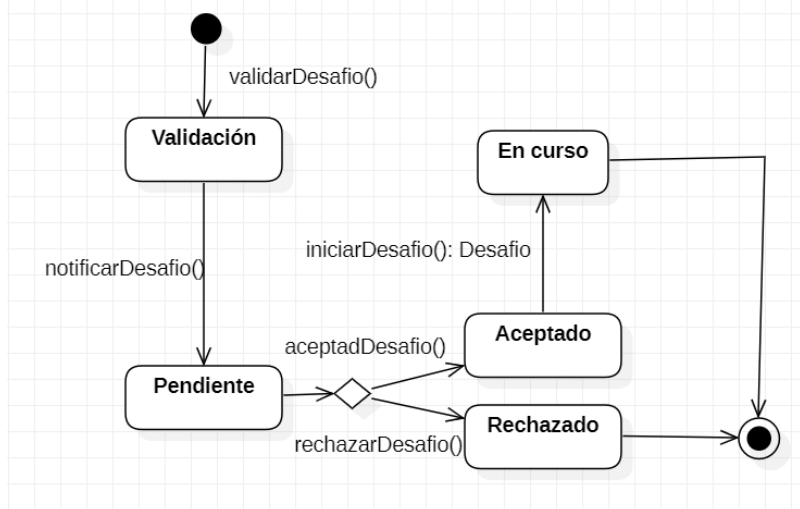


Diagrama estados 2

Como se puede ver, el desafío pasa por distinto estados hasta que el usuario desafiado puede aceptarlo o rechazarlo (diagrama estados 1):

1. **Validación:** el operador comprueba las fortalezas y debilidades de ambos (desafiante y desafiado).
2. **Pendiente:** el desafío será publicado y se enviará una notificación al desafiado.

Aceptado o rechazado: que será una decisión que haya tomado el desafiado.

3. **Aceptado:** se inicia el combate y pasa a estar en curso.

3.1. **En curso:** se está desarrollando el desafío entre ambos usuarios.

4. **Rechazado:** Se termina el desafío y el desafiante tendrá que pagar el 10% del oro que el desafiante haya apostado.

Finalización de un combate

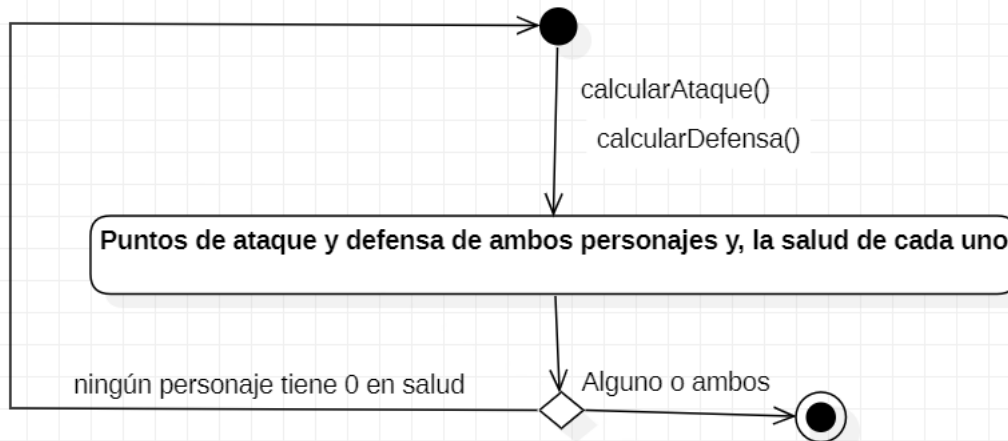


Diagrama estados 3

Cada vez que se empieza una ronda del combate, se calcula el ataque, defensa y salud de cada uno de los dos personajes, que después se compararán estos datos para saber cuál es el ganador de cada ronda.

Pero este diagrama ilustra en concreto lo que pasa si alguno de los personajes muere.

- Si los dos mueren a la vez se acaba el combate o alguno de ellos, el combate acabará.
- Si no muere ninguno en esa ronda, se pasará a la siguiente.

Diagramas de caso de uso

Estos diagramas que se van a describir a continuación permiten conocer el tipo de roles del juego y cómo interactúan con el sistema.

En este caso, van a existir dos roles: rol cliente usuario y rol cliente operador.

Caso de uso del usuario

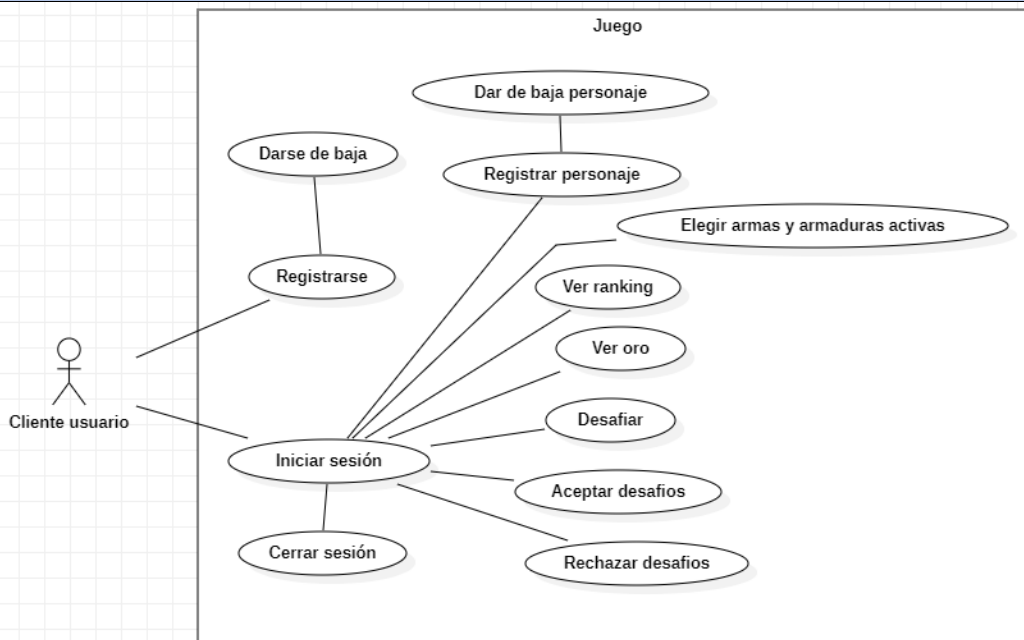


Diagrama de caso de uso 1

Caso de uso del operador

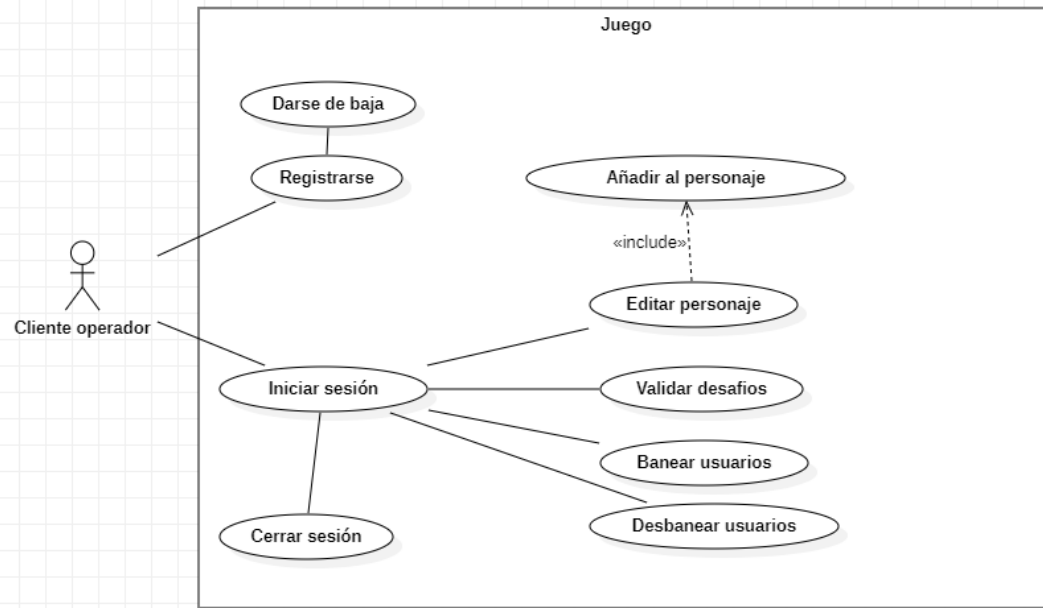


Diagrama de caso de uso 2