

José Victor García Llorente

Alejandro Martín Carrera

Jorge Padilla Rodríguez

Eloy de Sande de las Heras

Olga Somalo Serrano

Descripción pormenorizada de las clases presentes en el programa “MetProg URJC GAME”, así como de las relaciones entre ellas. También se detallan diagramas que explican el desarrollo de la aplicación.

Fase Diseño

Contenido

Diagrama de clases 2

Diagramas de secuencia..... 4

 Inicio del programa 4

 Verificación del desafío 5

 Consultar equipo 6

 Desarrollo del desafío 7

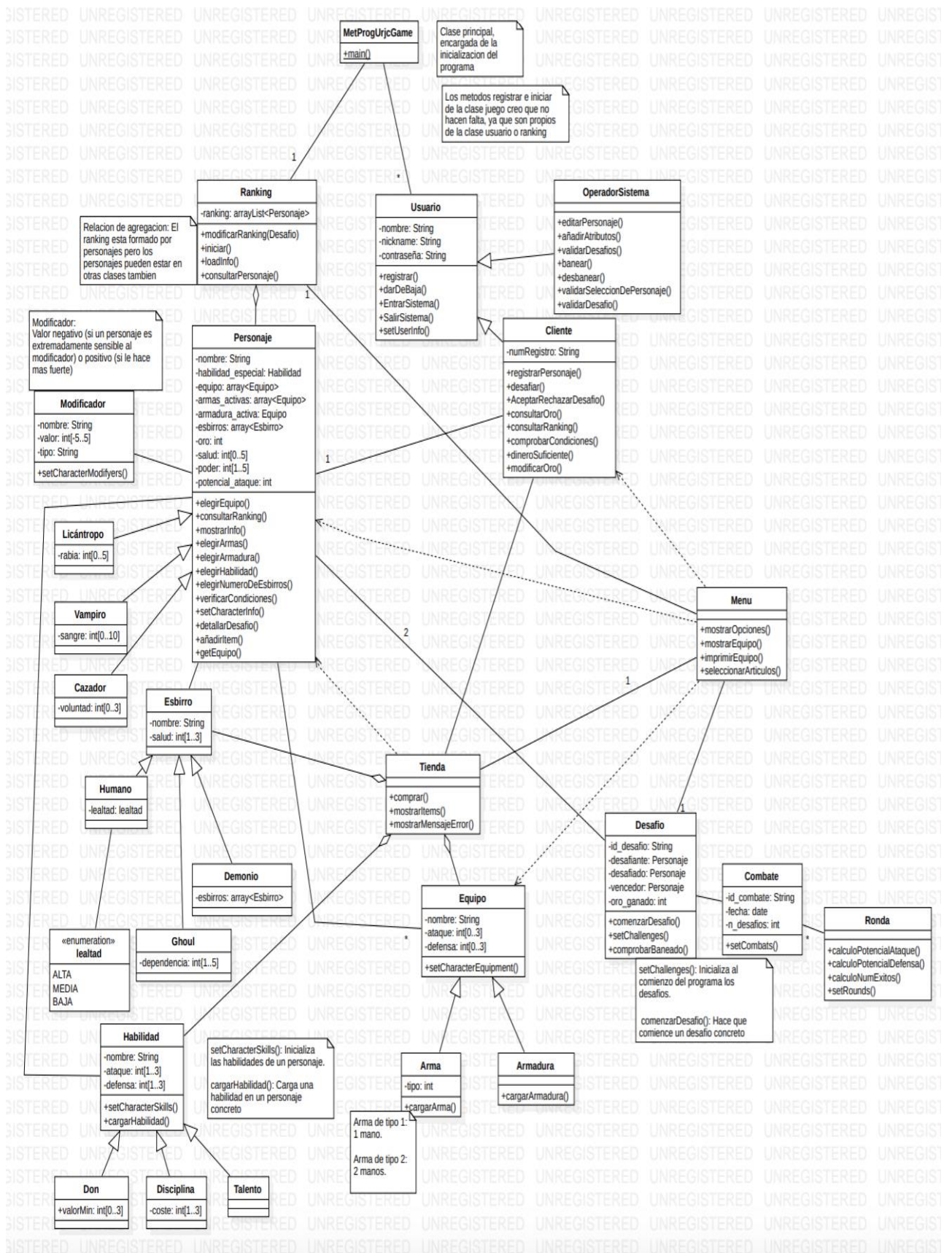
 Comprar item 9

Diagramas de estados 10

 Menú 10

 Combate 12

Diagrama de clases

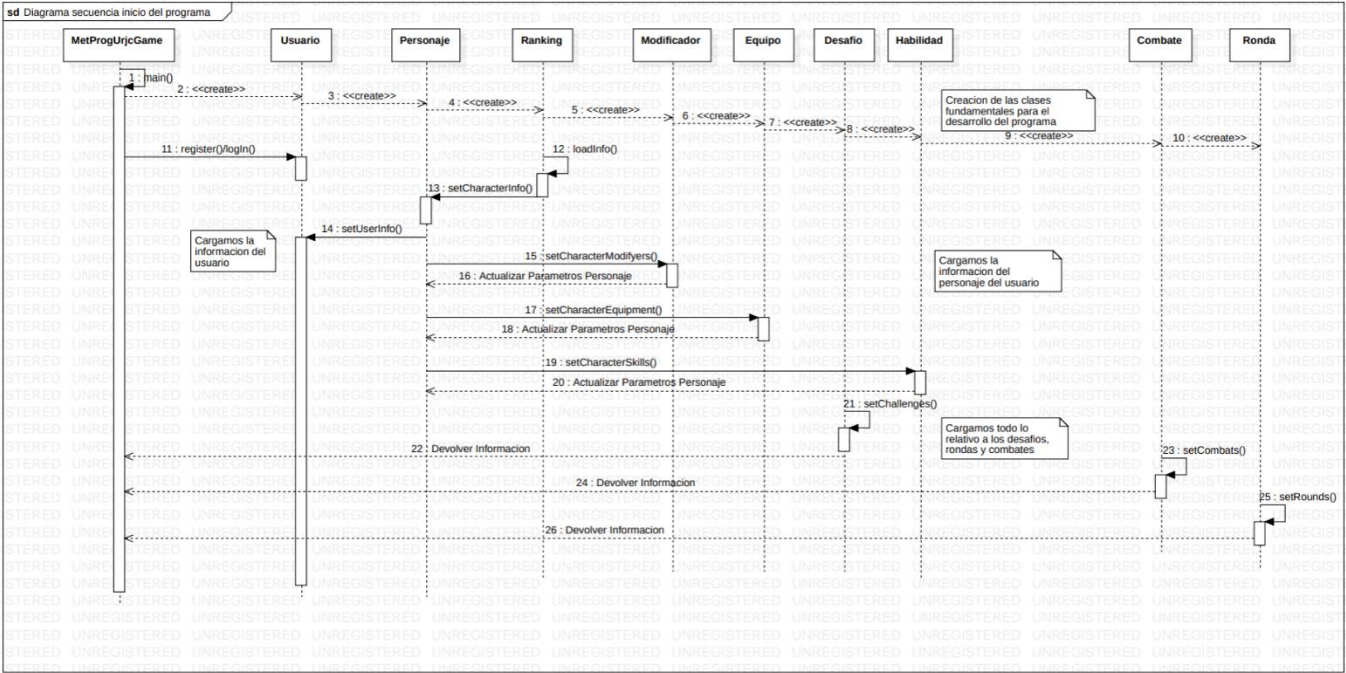


Para realizar el programa nos servimos de varias clases diferentes relacionadas entre sí. Veamos algunas de las clases principales:

- MetProgUrcGame: clase principal encargada de inicializar el programa.
- Personaje: clase correspondiente a las criaturas principales del juego. Guarda su información para poder manejarla. Hay otras clases que guardan información asociada a los personajes como Esbirro, Habilidad, Equipo, Modificador y derivados.
- Menú: clase encargada de la interfaz de usuario. Muestra las opciones y espera respuesta para actuar en función de esta.
- Usuario: clase correspondiente a una cuenta que da uso al programa. Dos clases derivan de usuario: Cliente, que corresponde a los jugadores, y OperadorSistema, que corresponde a los administradores. Estas clases guardan información acerca de la cuenta e incluyen métodos con las acciones que pueden realizar.
- Desafío: clase correspondiente a los desafíos entre jugadores. Guardan la información y la lógica necesaria para que se lleven a cabo. Precisa de otras clases para aspectos más específicos del desafío como Combate y Ronda.
- Ranking: clase encargada de almacenar y manejar el listado de personajes ordenado por victorias.

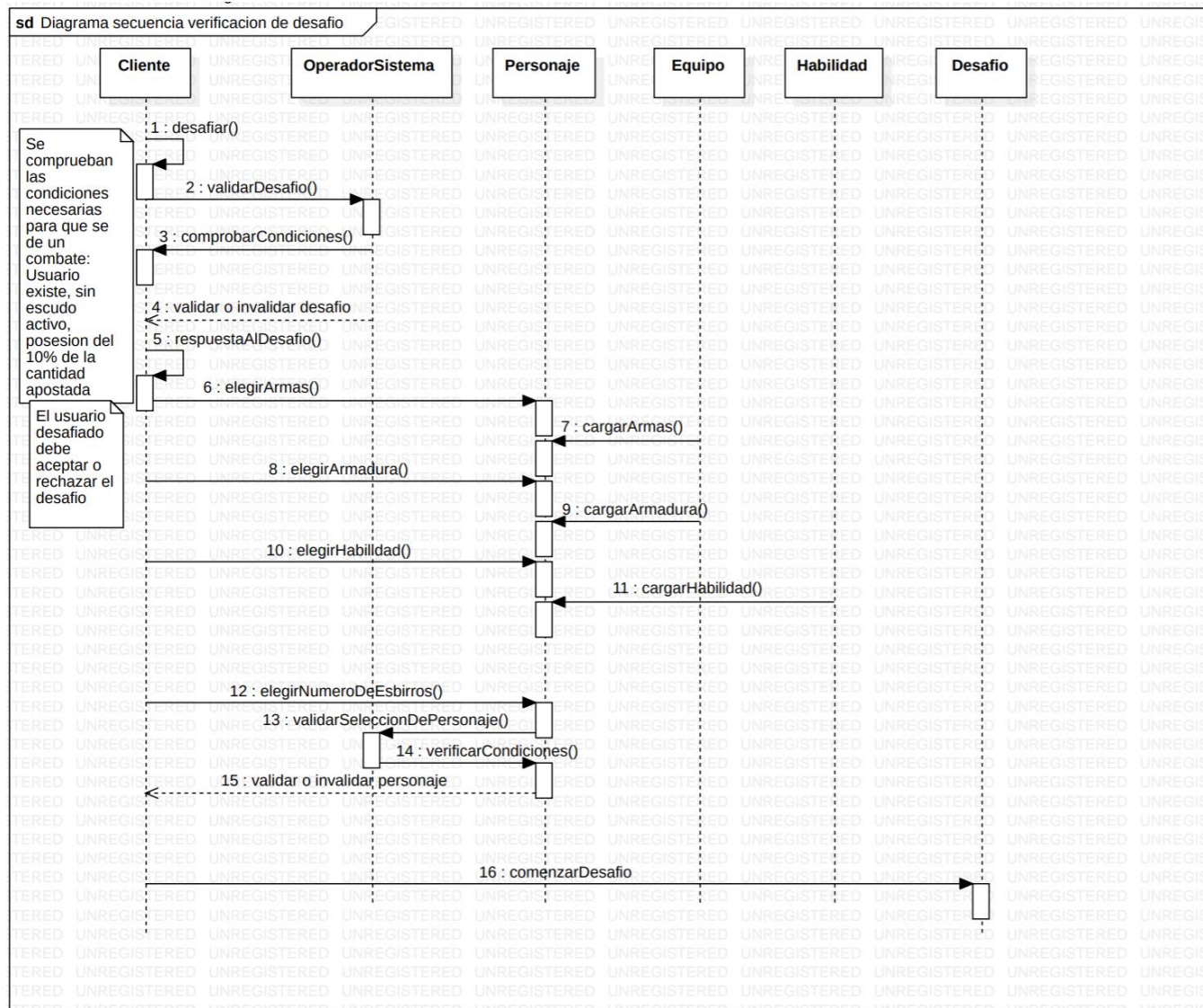
Diagramas de secuencia

Inicio del programa



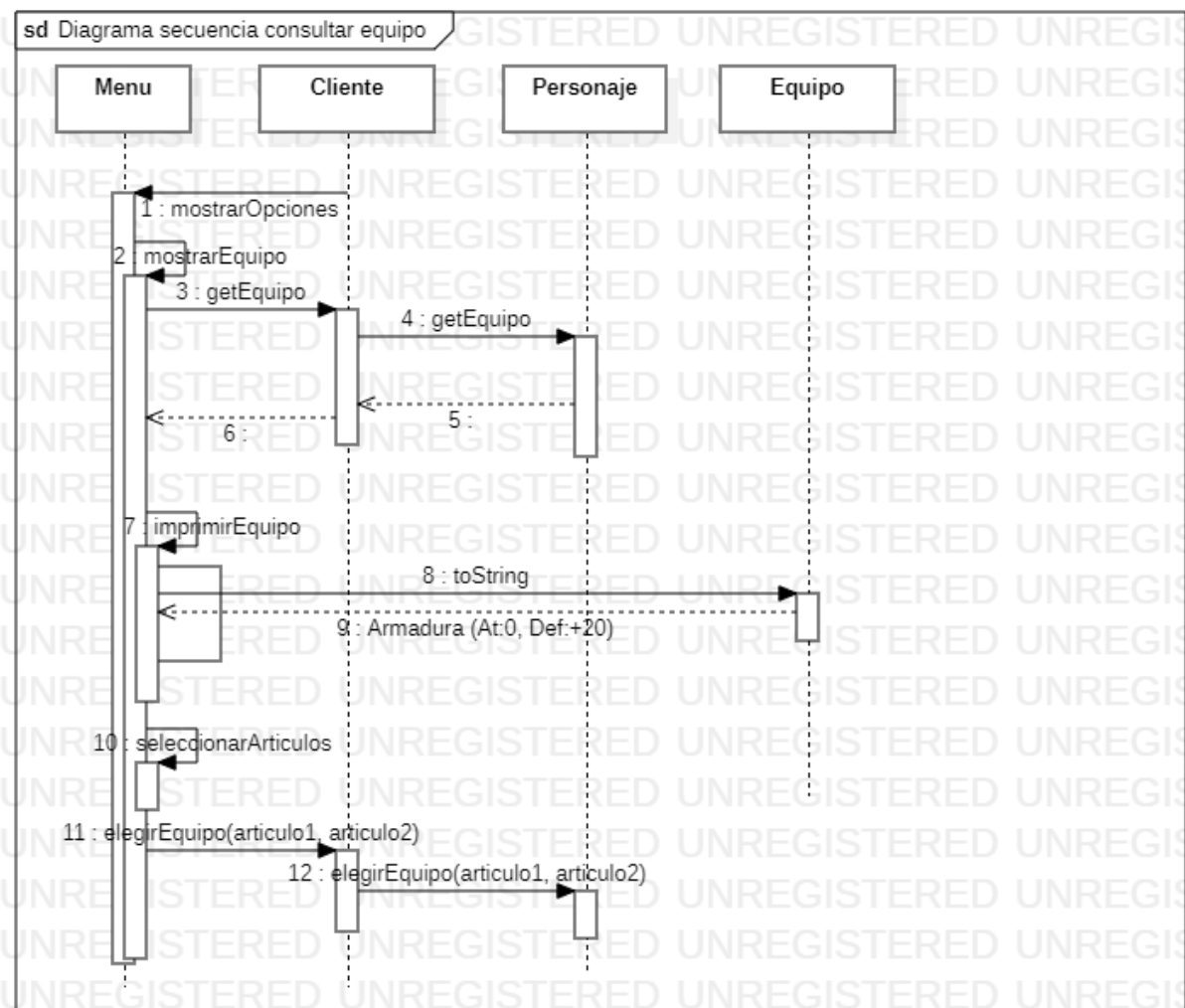
El programa comienza con la llamada al main. Seguidamente se procede a cargar la estructura del programa. Esto incluye Usuarios, Personajes, Ranking y Desafíos. Para ello primero se crean los objetos y después se procede al registro de usuario. Una vez tengamos un usuario válido, se carga la estructura ranking y se busca en él al usuario para completar la información del objeto Usuario y asociados.

Verificación del desafío



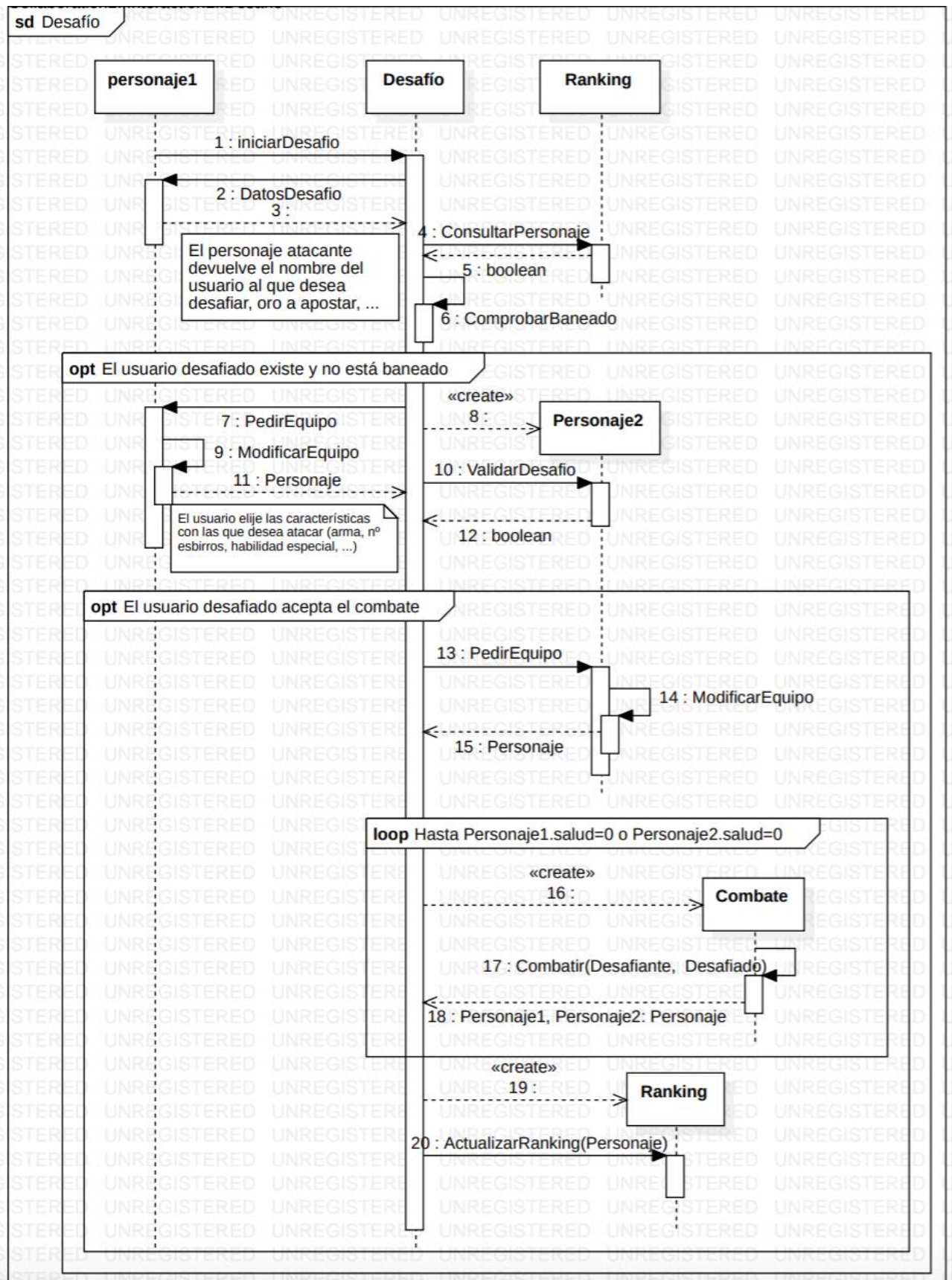
Un desafío comienza con la llamada a `desafiar()` por parte de un cliente. Se comienza llamando a un `OperadorSistema` para que comunique si el desafío es válido, comprobando si los usuarios implicados cumplen las condiciones. Si se cumplen, el usuario llama a los métodos que permiten elegir los artículos y habilidades para usar y se comprueba si la selección es válida. Finalmente se espera respuesta al desafío por parte del contrincante, que deberá enfrentarse al mismo procedimiento. Si ambos aceptan se llama al objeto `Desafio` para que ejecute la disputa y comunique el resultado.

Consultar equipo



La consulta y modificación del equipo del personaje se inicia con la llamada por parte del cliente a través de las opciones del menú. Cuando se elige mostrar el equipo, el menú toma del cliente que le está dando uso el equipo de su personaje a través del método `getEquipo`. Después se procede a mostrarlo por pantalla con `imprimirEquipo`. Finalmente, el cambio de artículos de equipo se realiza si el cliente selecciona en el menú dos artículos que quiere intercambiar. En este caso Menú llama a `elegirEquipo` de cliente con los artículos seleccionados para realizar el cambio en su personaje.

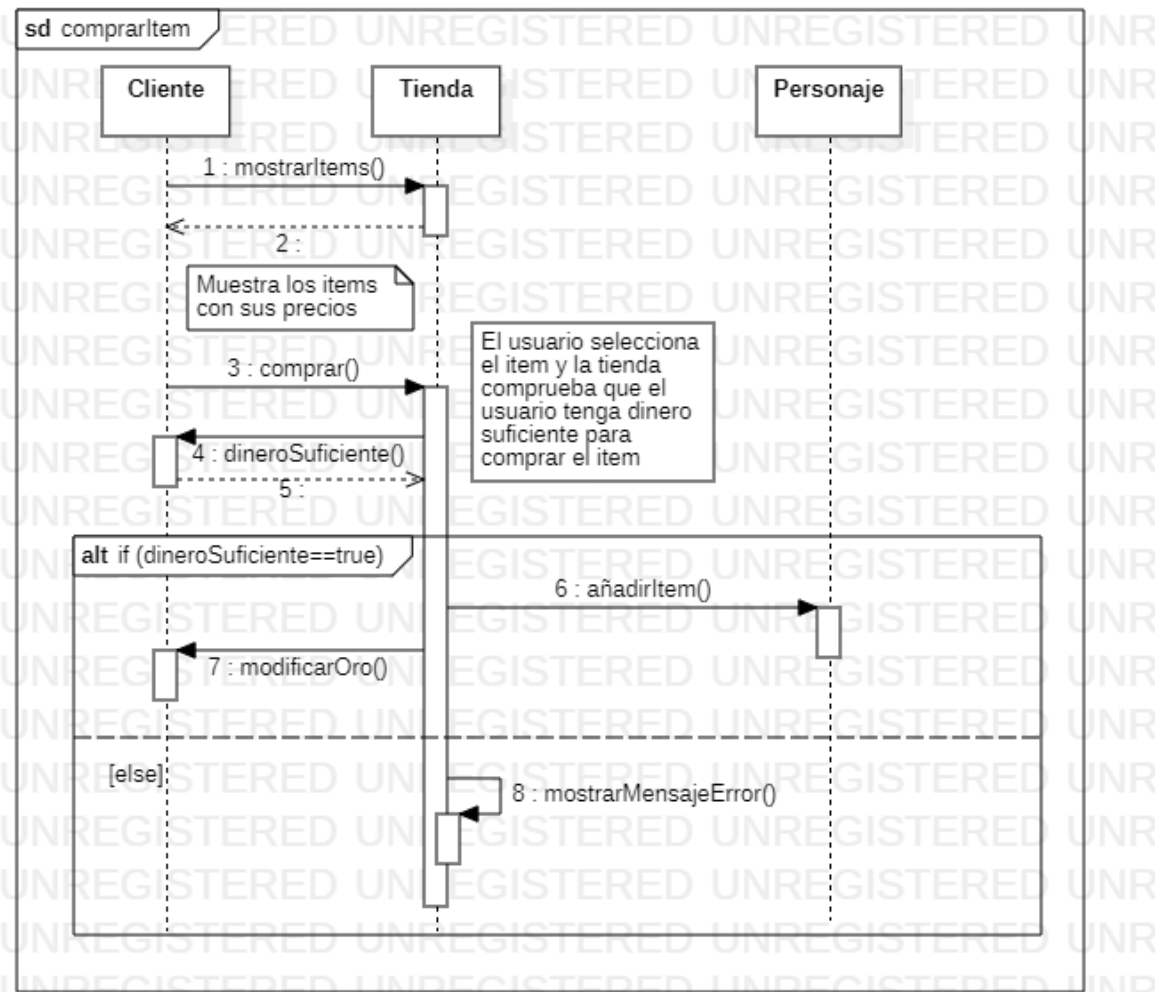
Desarrollo del desafío



Al iniciar un desafío, previamente un cliente tiene que haber retado a otro. Esto hace que necesariamente el programa tenga que comprobar que dicho jugador no está baneado y además exista. Una vez comprobado lo anterior, el usuario que desafió debe configurar el personaje que usará en el combate. Al mismo tiempo, el usuario desafiado debe aceptar o rechazar el desafío, y en el caso de que se haya aceptado, el usuario desafiado debe configurar las características de su personaje para usar en el combate.

A partir de este momento, ambos personajes se enfrentan en una serie de combates, hasta que la salud de alguno de ellos sea igual a cero. Una vez se llegue a cero en la salud de alguno, se guardará el resultado del desafío en cada uno de los personajes y en consecuencia, se modificará el ranking del programa.

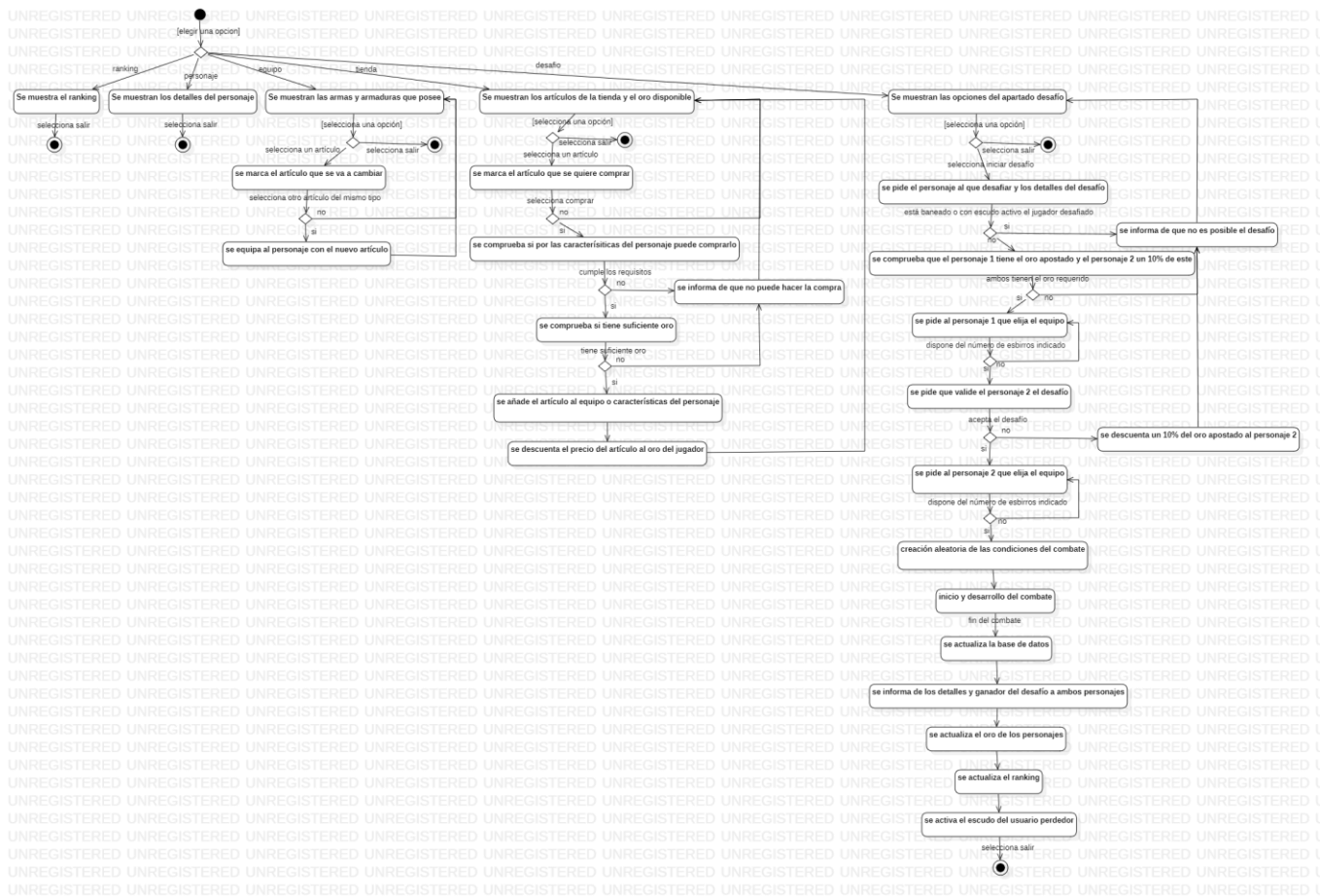
Comprar item



Si el cliente elige la opción de comprar items en el menú, la clase Tienda le mostrará los items con sus precios. El usuario seleccionará el item que desea comprar y se comprobará que tenga dinero suficiente para comprarlo. Si lo tiene, se añadirá el item a su personaje y se descontará el oro que costaba del total. Si no, se mostrará un mensaje de error, en el que se le indicará que no dispone de oro suficiente para comprar el item.

Diagramas de estados

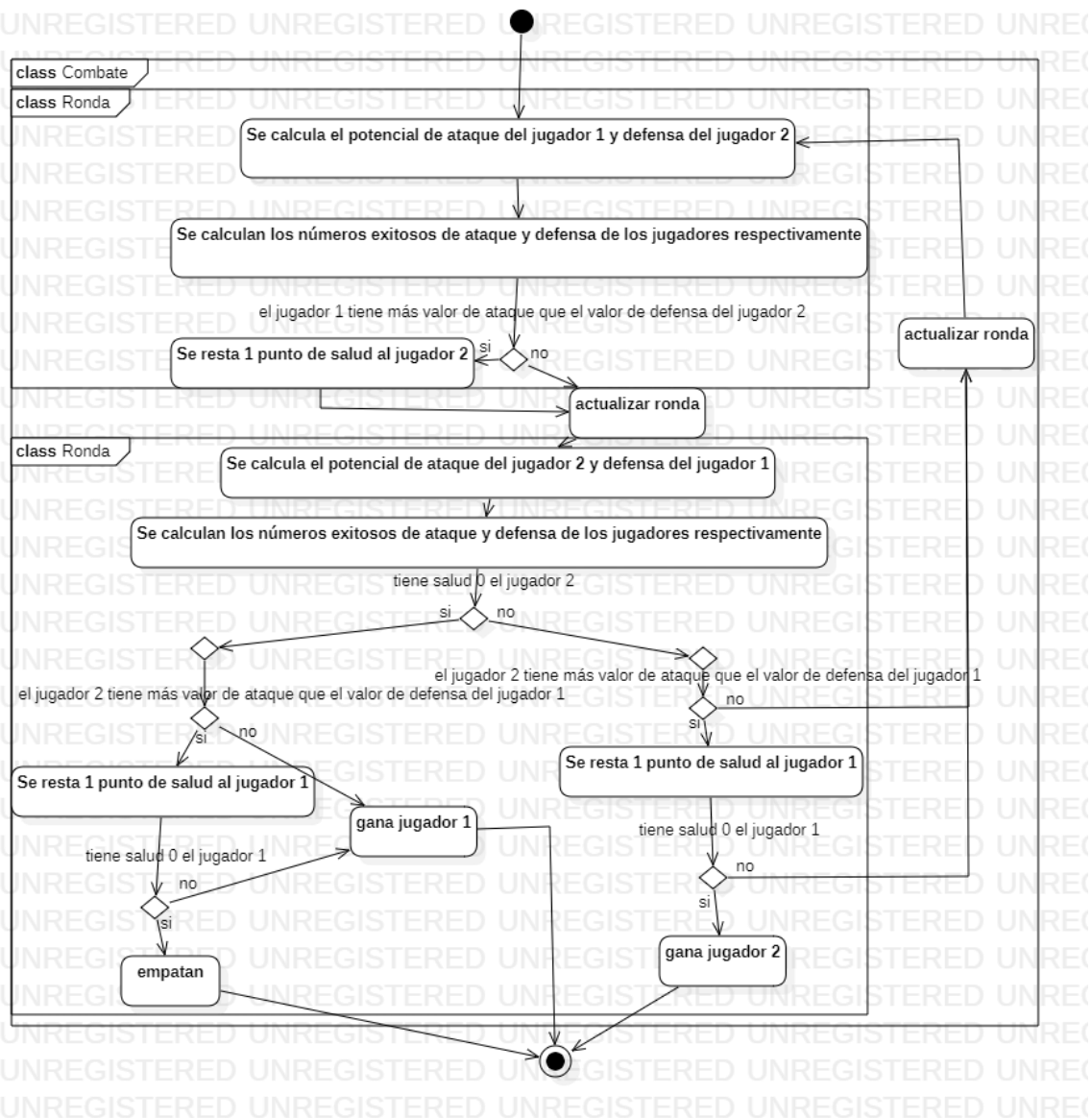
Menú



El funcionamiento del menú se representa en el diagrama superior. Comenzamos con 5 opciones iniciales:

- Ranking: da paso al listado de usuarios ordenado por victorias
- Personaje: muestra las características del personaje asociado al usuario
- Equipo: muestra los artículos del equipo y sus características, y permite equipar al personaje con algunos artículos
- Tienda: muestra los artículos en venta y el oro del que se dispone, y permite seleccionar artículos para comprarlos si se cumplen las condiciones necesarias.
- Desafío: pide la información necesaria para iniciar un desafío. Comienza solicitando el usuario a desafiar, y sigue con la elección del equipo y el desarrollo del combate si es posible.

Combate



En el diagrama se explica el desarrollo del combate. En cada ronda se irá alternando el rol de atacante y defensor de cada personaje, de manera que se calcularán sus potenciales de ataque (jugador 1) y defensa (jugador 2) y en la siguiente ronda se alternará el cálculo de estos (potencial de ataque del jugador 2 y de defensa del jugador 1). Con estos valores se conseguirá calcular el valor de ataque y defensa, a través de la obtención de números exitosos. De esta manera, si el valor de ataque, del jugador que ataca, es mayor que el valor de defensa, del jugador que defiende, se restará 1 punto de salud al defensor. Esto se repetirá varias rondas. Si después del ataque de ambos jugadores, uno de ellos se queda sin salud, el otro jugador gana, sin embargo, si los dos jugadores se quedan sin salud, empatan.