|  |  |
| --- | --- |
| CLIENTE | Empresa de videojuegos |
| USUARIO | Jugadores |
| REQUERIMIENTOS FUNCIONALES | R1. Registro de jugadores.  R2. Registro de enemigo a un nivel.  R3. Registro de tesoro a un nivel.  R4. Modificar puntaje de jugador  R5. Incrementar nivel del jugador  R6. Informar los tesoros y enemigos de un nivel  R7. Informar cantidad total de un tesoro especifico  R8. Informar cantidad total de un enemigo especifico  R9. Informar el tesoro mas repetido en los niveles  R10. Informar el enemigo que otorga mayor puntaje y en qué nivel se ubica.  R11. Informar la cantidad de consonantes encontradas en los nombres de los enemigos  R12. Informar el top 5 de jugadores de acuerdo con el puntaje |
| CONTEXTO DEL PROBLEMA | La empresa de videojuegos necesita un videojuego con niveles en el que el jugador recoge tesoros y pelea con enemigos. Por ahora el problema solo es para las funcionalidades de el registro de jugadores, niveles y tesoros. |
| REQUERIMIENTOS NO FUNCIONALES | RNF1. Máximo de 20 jugadores, 50 tesoros y 25 enemigos.  RNF2. Los 10 niveles se crean automáticamente  RNF3. La resolución será solamente HD  RNF4. El despliegue de los tesoros y enemigos en la aplicación web no puede tardar más de 2 segundos.  RNF5. El sistema debe funcionar en aplicación web y aplicación para móviles. |

**REQUERIMIENTOS FUNCIONALES**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R1 | | |
| Resumen | Registro de jugadores | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nickname | String | No puede repetirse con otros jugadores |
| name | String |  |
| Actividades generales necesarias para obtener los resultados | 1. Pedirle al usuario la información 2. Chequear que el nickname no se repita con el nickname de otro jugador | | |
| Resultado o postcondición | Se registra un jugador | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R2 | | |
| Resumen | Registro de enemigos a un nivel | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| levelID | int | Debe ser el id de un nivel que exista |
| nameID | String | No se puede repetir con otros enemigos |
| loseScore | double | Debe ser un double valido |
| winScore | double | Debe ser un double valido |
| Actividades generales necesarias para obtener los resultados | 1. Pedirle al usuario todos los datos 2. Verificar que se cumplan las condiciones de selección | | |
| Resultado o postcondición | Se añade un enemigo al nivel | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R3 | | |
| Resumen | Registro de tesoro a un nivel | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| name | String |  |
| treasureID | int | No se puede repetir con otros tesoros |
| levelID | int | Debe ser un id de un level valido |
| imageURL | String |  |
| scoreGranted | double | Debe ser un double valido |
| Actividades generales necesarias para obtener los resultados | 1. Pedirle al usuario todos los datos 2. Verificar que se cumplan las condiciones de selección | | |
| Resultado o postcondición | Se añade un tesoro a un nivel | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R4 | | |
| Resumen | Modificar el puntaje de un jugador | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nickname | String | Debe ser un nickname de un jugador valido |
| scoreToModify | double | Debe ser un double valido |
| Actividades generales necesarias para obtener los resultados | 1. Se le piden los datos al usuario 2. Verificar que se cumplan las condiciones de selección | | |
| Resultado o postcondición | Se modifica el puntaje del jugador correspondiente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R5 | | |
| Resumen | Incrementar el nivel en el que se encuentra el jugador | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nickname | String | Debe ser un nickname de un jugador valido |
| Actividades generales necesarias para obtener los resultados | 1. Se le pide el dato al usuario 2. Se verifica que el nickname sea de un jugador valido 3. Se verifica cuanto puntaje tiene ese jugador 4. Se verifica en que nivel está 5. Se verifica en ese nivel cuantos puntos se requiere para subir de nivel | | |
| Resultado o postcondición | En caso de que el jugador tenga suficientes puntos para subir de nivel, subirá de nivel, de lo contrario, se le avisara cuanto puntaje necesita | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | El jugador no tiene suficiente puntaje para subir de nivel |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R6 | | |
| Resumen | Informar tesoros y enemigos de un nivel | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| levelID | int | Debe ser el ID de un nivel valido |
| Actividades generales necesarias para obtener los resultados | 1. Se le pide el dato al usuario y se comprueba que cumpla la condición 2. Se hace un recorrido total a los arrays de tesoros y enemigos y para cada objeto de cada array se hace un getName() 3. Esos nombres se almacenan en un string | | |
| Resultado o postcondición | Se imprime el string en la pantalla con el nombre de los tesoros y enemigos | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R7 | | |
| Resumen | Informar la cantidad total de un tesoro incluyendo todos los niveles | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| treasureName | String |  |
| Actividades generales necesarias para obtener los resultados | 1. Se le pide el dato al usuario 2. Se hace un recorrido total en todos los niveles 3. En cada nivel se hace un recorrido total de los tesoros de ese nivel buscando tesoros con el nombre ingresado por el usuario 4. Cada tesoro encontrado aumenta un contador de tesoros encontrados | | |
| Resultado o postcondición | Se imprime un mensaje al usuario con ese numero | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | En caso de que el contador sea 0 el mensaje será diferente |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R8 | | |
| Resumen | Informar la cantidad total de un tipo de enemigo incluyendo todos los niveles | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| enemyType | int | Debe ser uno de los 4 tipos de enemigos |
| Actividades generales necesarias para obtener los resultados | 1. Se le pide el dato al usuario y se comprueba que el int cumpla la condición. 2. Se hace un recorrido total en todos los niveles 3. En cada nivel se hace un recorrido total de los enemigos de ese nivel buscando enemigos que sean del tipo ingresado por el usuario 4. Cada enemigo encontrado aumenta un contador de enemigos encontrados | | |
| Resultado o postcondición | Se imprime un mensaje al usuario con ese numero | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | En caso de que el contador sea 0 el mensaje será diferente |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R9 | | |
| Resumen | Informar el tesoro más repetido en todos los niveles | | |
| Actividades generales necesarias para obtener los resultados | 1. Creamos dos arrays, uno con los nombres de los tesoros y otro con la cantidad de veces que se repite ese tesoro 2. Se hace un recorrido total de los niveles 3. Se hace un recorrido total de los tesoros de los niveles 4. Por cada tesoro se hará un chequeo de si el nombre ya está en el array de nombres o no, si está se le sumara 1 a su posición correspondiente en el array de veces que se repite, si no está se añade el nombre al array de nombres y se pone 1 en el array de veces que se repite 5. Al finalizar esto en todos los niveles, se hace un recorrido total en el array de veces que se repite y se busca el numero mas grande 6. Al encontrar la posición del numero mas grande se busca esa posición en el array de nombres | | |
| Resultado o postcondición | Se imprime un mensaje al usuario con el nombre del tesoro que mas se repite | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R10 | | |
| Resumen | Informar que enemigo otorga mayor puntaje y en que nivel se ubica | | |
| Actividades generales necesarias para obtener los resultados | 1. Se crean dos arrays de 10 posiciones, uno de nombres y otro de scores (Cantidad de niveles) 2. Se hace un recorrido total por todos los niveles 3. Por cada nivel se hace un recorrido total por los enemigos 4. Se alla el enemigo que mas puntaje otorga de ese nivel en especifico y se añade a los arrays. 5. Al final de recorrer todos los niveles se chequea cual de los 10 enemigos restantes tiene el mayor puntaje. | | |
| Resultado o postcondición | Al finalizar se muestra un mensaje con el enemigo que otorga mayor puntaje | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R11 | | |
| Resumen | Informar la cantidad de consonantes encontradas en los nombres de los enemigos | | |
| Actividades generales necesarias para obtener los resultados | 1. Se hace un recorrido total por todos los niveles 2. Se hace un recorrido total por los enemigos del nivel 3. Se hace un getName() para cada enemigo y se cuentan las consonantes de ese nombre, la cantidad de consonantes se suman a un contador | | |
| Resultado o postcondición | Al finalizar todos los recorridos se imprime un mensaje con ese numero | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| msj | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R12 | | |
| Resumen | Informar el top 5 de jugadores de acuerdo con el puntaje | | |
| Actividades generales necesarias para obtener los resultados | 1. Se hacen 2 arrays con 5 posiciones, uno de nombres y otro de scores 2. Se inicializa el array de nombres como algún mensaje por defecto 3. Se hace un recorrido total por todos los jugadores 4. Por cada jugador se hace un getScore(), se chequea si ese score es mayor que el puntaje en cada posición del array de socres, si es mayor entonces se moverá a la siguiente posición hasta llegar al final (puesto 1) o no ser mayor 5. Se coloca el puntaje y nickname del jugador en la posición correspondiente y se mueve los otros jugadores del top en el array si es necesario | | |
| Resultado o postcondición | Al finalizar el recorrido se hace un print de una tabla con el top 5 | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String |  |