|  |  |
| --- | --- |
| CLIENTE | Una empresa de videojuegos |
| USUARIO | Los administradores del videojuego |
| REQUERIMIENTOS FUNCIONALES | RF1-Registrar jugadores  RF2-Registrar niveles  RF3-Registrar tesoros en un nivel  RF4-Registrar enemigos en un nivel  RF5-Modificar el puntaje de un jugador.  RF6-Incrementar nivel para un jugador  RF7-Informar los tesoros y enemigos de un nivel  RF8-Informar la cantidad encontrada de un tesoro en todos los niveles  RF9-Informar la cantidad encontrada de un tipo de enemigo en todos los niveles.  RF10-Informar el tesoro más repetido en todos los niveles.  RF11-Informar el enemigo que otorga mayor puntaje y el nivel donde se ubica.  RF12-Informar la cantidad de consonantes encontradas en los nombres de los enemigos del juego.  RF13-Informar el top 5 de los jugadores de acuerdo con el puntaje.  RF14-Generar la posición al azar de un enemigo y un tesoro. |
| CONTEXTO DEL PROBLEMA | Para esta versión del juego, se tendrán un máximo de 20 jugadores, 50 tesoros y 25 enemigos. Es importante tener en cuenta que, al momento de iniciar el juego se crearán automáticamente los 10 niveles y se debe preguntar al usuario cual es la resolución sobre la cual se ejecutará el juego. Ejemplo (El usuario escoge la resolución QHD) y esta información será utilizada al momento de generar randómicamente las posiciones en pantalla. |
| REQUERIMIENTOS NO FUNCIONALES | RNF1- el  despliegue de los tesoros y enemigos de un nivel, en la aplicación web no debe tardar más de 2 seg.  RNF2-El sistema debe funcionar tanto en aplicaciones web como en app para móviles. |

**Especificación de requerimientos funcionales**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | RF1-Registrar jugadores. | | |
| Resumen | Dados los datos de los jugadores (nickname y nombre) el sistema debe verificar que el nickname no exista para poder guardarlo con un puntaje y numero de vidas iniciales por defecto;10 y 5, respectivamente. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nickName | String | -- |
| name | String | -- |
|  |  |  |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | El sistema verifica que el nickname no haya sido registrado previamente y que haya espacio disponible para guardar al usuario | | |
| Resultado o postcondición | El jugador registrado en el sistema. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | RF2-Registrar niveles | | |
| Resumen | El sistema registra automáticamente los niveles del juego y les asigna un puntaje de avance por defecto. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- | -- | -- |
| -- | -- | -- |
|  |  |  |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | El juego debe estar creado previamente | | |
| Resultado o postcondición | El nivel registrado en el videojuego. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | RF3-Registrar tesoros en un nivel | | |
| Resumen | Dados los datos de un tesoro (nombre, URL, puntos, posición), el número de tesoros que debe haber y el número del nivel en el cual se debe registrar, el sistema deberá verificar que: exista el nivel al cual se quiere registrar el tesoro, que el número de tesoros no exceda el máximo establecido (50), para poder registrar el tesoro en el nivel indicado, la cantidad de veces indicadas. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| name | String | -- |
| url | String | -- |
| points | int | -- |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que haya espacio para agregar la cantidad de tesoros indicados por el usuario, posteriormente registra la cantidad de tesoros que fue indicada | | |
| Resultado o postcondición | La cantidad de tesoros que el usuario indico registrada en el sistema | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | RF4-Registrar enemigos en un nivel | | |
| Resumen | Dados los datos de un enemigo(nombre,tipo, puntaje que resta, puntaje que suma), el sistema deberá verificar que el enemigo aun no haya sido registrado en el nivel al que se quiere registrar para de esta manera evitar que el enemigo sea registrado varias veces en un mismo nivel.Ademas, también se debe verificar que haya un espacio disponible en la cual registrar al enemigo. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| name | String | -- |
| type | String | -- |
| substracScore | int | -- |
| addScore | int | -- |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que el enemigo no haya sido registrado previamente en el nivel y que hayan espacios disponibles para registrar al enemigo. Para de esta manera, registrar el enemigo al nivel indicado. | | |
| Resultado o postcondición | El enemigo registrado en el nivel indicado | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- |  |  |
|  |  |  |
|  |  |  |
| Nombre o identificador | RF5- Modificar el puntaje de un jugador. | | |
| Resumen | Dado el nickname de un jugador, el sistema debe confirmar que el jugador existe, para posteriormente cambiar el puntaje de este | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nickname | String | -- |
| score | int | -- |
| Actividades generales necesarias para obtener los resultados | El sistema verifica que el nickname ingresado corresponde a un jugador registrado, para posteriormente cambiar el puntaje actual de ese jugador por uno nuevo digitado por el usuario. | | |
| Resultado o postcondición | El nuevo puntaje del enemigo. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| score | int | Si el jugador existe. |
| Nombre o identificador | RF6-Incrementar nivel para un jugador | | |
| Resumen | Dado el nickname de un jugador, el sistema deberá verificar que el jugador exista, para después verificar que el puntaje que este tiene es suficiente, de ser así, incrementara el nivel del jugador de no ser así, informa al usuario que nivel requiere para ascender de nivel. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nickname | String | -- |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que el jugador exista, para posteriormente acceder al puntaje que tiene el jugador y ver si es suficiente para avanzar de nivel, de ser así, el jugador será ascendido de nivel, de no serlo se informara el puntaje que necesita para avanzar. | | |
| Resultado o postcondición | Mensaje con la información de ascenso de nivel del jugador. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| Message | String | Si el jugador existe. |
| Nombre o identificador | RF7-Informar los tesoros y enemigos de un nivel | | |
| Resumen | Dado el numero de un nivel, ingresado por un usuario, el sistema debe informar cuantos tesoros y enemigos hay registrados en este | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| levelNumber | String | -- |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que el nivel, este registrado para posteriormente acceder a la información sobre el numero de tesoros y enemigos registrados en el. | | |
| Resultado o postcondición | Mensaje con la información solicitada. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| Message | String | Si el nivel existe. |
| Nombre o identificador | RF8-Informar la cantidad encontrada de un tesoro en todos los niveles | | |
| Resumen | Dado el nombre de un tesoro, el sistema debe verificar que este exista. Posteriormente, debe saber en que niveles se encuentra, contar la cantidad de tesoros en cada nivel para posteriormente sumarlos | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| treasureName | String | -- |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que el tesoro haya sido registrado previamente. Luego, el sistema comprueba en que niveles esta registrado y cuantas veces en cada uno para de esta manera sumarlos entre si y obtener la cantidad de ese tesoro en todos los niveles | | |
| Resultado o postcondición | Cantidad de veces que el tesoro se encuentra en todos los niveles | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| treasureQuantity | int | Si el tesoro existe por lo menos en un nivel |
| Nombre o identificador | RF9-Informar la cantidad encontrada de un tipo de enemigo en todos los niveles. | | |
| Resumen | Dado un tipo de enemigo, el sistema debe verificar que exista por lo menos un enemigo de ese tipo en el juego, para posteriormente contar cuantos enemigos de este tipo existen en cada nivel del juego. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| enemyType | String | -- |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que al menos un enemigo del tipo ingresado exista, para posteriormente buscar en cada nivel que enemigos son de este tipo y sumar su cantidad | | |
| Resultado o postcondición | Cantidad de enemigos de un tipo especifico | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| enemyTypeQuantity | int | Si al menos existe un enemigo del tipo solicitado |
| Nombre o identificador | RF10-Informar el tesoro más repetido en todos los niveles. | | |
| Resumen | El sistema busca en todos los niveles que tesoros se repiten y cuantas veces lo hacen, posteriormente compara estas cantidades para buscar el tesoro más repetido | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- |  |  |
| Actividades generales necesarias para obtener los resultados | El sistema verifica que exista por lo menos un tesoro para después buscar que tesoros se repiten en varios niveles y cuantas veces lo hacen, para posteriormente sumar la cantidad de veces que se repite cada uno de forma individual y compararlas entre si para de esta manera obtener el tesoro más repetido. | | |
| Resultado o postcondición | El tesoro mas repetido en todos los niveles del juego. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| treasureName | String | Si al menos un tesoro ha sido registrado. |
| Nombre o identificador | RF11-Informar el enemigo que otorga mayor puntaje y el nivel donde se ubica. | | |
| Resumen | El sistema busca entre los enemigos existentes el que mayor puntaje otorgue al jugador, para posteriormente ubicar en que nivel se encuentra e informarlo junto con su nombre | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- | -- | -- |
| Actividades generales necesarias para obtener los resultados | El sistema verifica que por lo menos exista un enemigo, para después entre los enemigos existentes verificar cual otorga mas puntaje y en que nivel se ubica. Estos datos deben ser informados al usuario. | | |
| Resultado o postcondición | Mensaje con la información solicitada | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| message | String | Si existe al menos un enemigo. |
| Nombre o identificador | RF12- Informar la cantidad de consonantes encontradas en los nombres de los enemigos del juego. | | |
| Resumen | El sistema cuenta entre los enemigos existentes cuantas de las letras en sus nombres corresponden a una consonante, posteriormente suma esta cantidad. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- | -- | -- |
| Actividades generales necesarias para obtener los resultados | El sistema verifica que exista por lo menos un enemigo registrado, cuenta en su nombre que letras corresponden a vocales y suma la cantidad total en todos los enemigos. | | |
| Resultado o postcondición | Cantidad de consonantes en las letras de los enemigos existentes en el juego. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| consonantsQuantity | int | Si existe por lo menos un enemigo |
| Nombre o identificador | RF13- Informar el top 5 de los jugadores de acuerdo con el puntaje. | | |
| Resumen | Dados los datos de los jugadores, el sistema comprueba el puntaje de cada uno y le informa al usuario el nombre de los cinco con mas puntaje. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- | -- | -- |
| Actividades generales necesarias para obtener los resultados | El sistema debe verificar que exista por lo menos un jugador registrado, después de esto, adquiere la información sobre el puntaje de estos y coteja los 5 jugadores con la mayor puntuacion para informarla al usuario. | | |
| Resultado o postcondición | La información de los cinco jugadores con mayor puntaje | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| Message | String | Si existe por lo menos un jugador. |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | RF14- Generar la posición al azar de un enemigo y un tesoro. | | |
| Resumen | Dependiendo de la resolución de pantalla del videojuego, se debe generar un numero al azar que se encuentre en el rango de esta | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| -- | -- | -- |
| Actividades generales necesarias para obtener los resultados | El sistema verifica la resolución de pantalla del videojuego en uno de sus dos ejes y genera una posición al azar para ese eje. | | |
| Resultado o postcondición | La coordenada de una posición en uno de los dos ejes. | | |
| Salidas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| coordinate | int | La resolución de pantalla debe existir. |
|  |  |  |
|  |  |  |

**TABLA DE TRAZABILIDAD:**

|  |  |  |  |
| --- | --- | --- | --- |
| RF1  Registrar jugadores | Class: Player | Player(....) | Model |
| Class: VideoGame | addPlayer() | Model |
| Class: VideoGame | findPlayer() | Model |
| Class: VideoGame | isAvailablePlayer() | Model |
| Class: VideoGameApp | addPlayer() | ui |
| RF2  Registrar niveles | Class: Level | Level(…) | Model |
| Class: Videogame | fillLevels() | Model |
| RF3  Registrar tesoros a un nivel | Class: Treasure | Treasure () | Model |
| Class: Level | addTreasure () | Model |
| Class: Level | isAvailableAmount() | Model |
| Class: VideoGame | addTreasure() | Model |
| Class: VideoGameApp | addTreasure() | Ui |
|  | VideoGame | searchLevel | Model |

|  |  |  |  |
| --- | --- | --- | --- |
| RF4  Registrar enemigos en un nivel | Class: Enemy | Enemy() | Model |
| Class: Level | searchEnemy() | Model |
| Class: Level | isAvailableEnemy() | Model |
| Class: Level | addEnemy() | Model |
| Class: VideoGame | addEnemy() | Model |
| Class: VideoGameApp | addEnemy() | Ui |
| Class: VideoGame | searchLevel() | Model |

|  |  |  |  |
| --- | --- | --- | --- |
| RF5  Cambiar puntaje de un jugador | Class: VideoGame | searchPlayer() | Model |
| Class: Player | setScore() | Model |
| Class: VideoGame | changePlayerScore() | Model |
| Class: VideoGameApp | changePlayerScore() | Ui |
| RF6  Incrementar nivel para un jugador | Class:Player | setCurrentLevel() | Model |
| Class: VideoGame | searchPlayer() | Model |
| Class: VideoGame | changePlayerLevel() | Model |
| Class: VideoGameApp | ChangePlayerLevel() | Ui |
| Class:VideoGame | Index() | Model |
| RF7  Informar los tesoros y enemigos de un nivel | Class: Enemy | getName() | Model |
| Class: Treasure | getName() | Model |
| Class: Level | showEnemies() | Model |
| Class: Level | showTreasures() | Model |
| Class: VideoGame | searchLevel() | Model |
| Class: VideoGame | EnemiesAndTreasures() | Model |
| Class: VideoGameApp | showTreasuresAndEnemies() | Ui |
|  | Class: Level | countEnemies() | Model |
| Class:Level | countTreasures() | Model |
| RF8  Informar la cantidad encontrada de un tesoro en todos los niveles | Class Level | treasureAmount() | Model |
| Class Treasure | getName() | Model |
| Class Videogame | treasureAmount() | Model |
| Class VideoGameApp | showTreasuresQuantity() | Ui |
| RF9  Informar la cantidad encontrada de un tipo de enemigo en todos los niveles. | Class Level | totalEnemiesPerType() | Model |
| Class VideoGame | totalEnemiesPerType() | Model |
| Class VideoGameApp | totalEnemiesPerType() | ui |
| Class:Enemy | CalculateEnemyType() | Model |
| RF10  Informar el tesoro más repetido en todos los niveles. | Class Level | treasuresNames() | Model |
| Class Level | countTreasureNames() | Model |
| Class Level | countTreasures() | Model |
| Class VideoGame | mostCommonTreasure() | Model |
| Clase VideoGame | totalTreasures() | Model |
| Class VideoGameApp | mostCommonTreasure() | ui |
| RF11  Informar el enemigo que otorga mayor puntaje y el nivel donde se ubica. | Class Level | biggestEnemy() | Model |
| Class VideoGame | biggestEnemy() | Model |
| Class VideoGame | index() | Model |
| Class Enemy | getScore() | Model |
| Class VideoGameApp | biggestEnemy() | ui |
| RF12  informar la cantidad de consonantes encontradas en los nombres de los enemigos del juego. | Class Enemy | calculateConsonants() | Model |
| Class Level | countEnemyConsonants() | Model |
| Class VideoGame | countConsonants() | Model |
| Class VideoGameApp | countConsonants() | ui |
| RF13  Informar el top 5 de los jugadores de acuerdo con el puntaje. | Class Player | getScore() | Model |
| Class Player | getName() | Model |
| Class VideoGame | topFive() | Model |
| Class VideoGameApp | topFive() | ui |
|  |  |  |
| RF14  Generar la posición al azar de un enemigo y un tesoro. | Class: VideoGame | calculateRandomPosition() | Model |
| enum: Resolution | Get weidth | model |
| Enum:Resolution | Get heigth | model |
| Class: Level | searchLevel() | Model |
| Class: Level | addTreasure() | Model |
| Class: Level | addEnemy() | Model |
| Class: VideoGame | addEnemy() | Model |
| Class: VideoGame | addTreasure() | Model |
| Class: VideoGameApp | addEnemy() | ui |
| Class: VideoGameApp | addTreasure() | ui |
| Class:Treasure | Treasure() | Model |
| Class: Enemy | Enemy() | Model |