**Juan José Barrera Gracia – A00394876**

**Algoritmos & Programación I**

**Tarea Integradora II (Diseño)**

**Universidad ICESI**

**Nicolás Javier Salazar Echeverry**

**2022-2**

|  |  |
| --- | --- |
| CLIENTE | Empresa de videojuegos |
| USUARIO | Jugadores del videojuego |
| REQUERIMIENTOS FUNCIONALES | R1: Registro de jugadores  R2: Registro de niveles  R3: Registro de tesoros  R4: Registro de enemigos  R5: Modificar puntaje del jugador  R6: Incrementar nivel de un jugador  R7: Informar tesoros y enemigos de un nivel  R8: Informar la cantidad de tesoros en todos los niveles.  R9: Informar la cantidad de enemigos en todos los niveles.  R10: Informar el tesoro más repetido en todos los niveles.  R11: Informar el enemigo que otorga más puntaje y el nivel donde se ubica.  R12: Informar la cantidad de consonantes encontradas en los nombres de los enemigos.  R13: Informar el top 5 de jugadores de acuerdo al puntaje. |
| CONTEXTO DEL PROBLEMA | |  | | --- | | Para mejorar el funcionamiento del sistema y tener claridades frente al mismo es pertinente hacerse las siguientes preguntas:   * En ningún momento se menciona el tiempo de desarrollo ¿El cliente desea la aplicación rápidamente? * ¿Será solo trabajo de los ingenieros telemáticos y de sistemas la creación de videojuego? O ¿Existen más partes involucradas? * ¿El juego desea proyectarse a futuro?, si es así, ¿Cuántos enemigos, jugadores y tesoros debería soportar a largo plazo? * ¿Cuál es puntaje necesario para pasar de un nivel a otro?, ¿será siempre el mismo o variará? | |
| REQUERIMIENTOS NO FUNCIONALES | \*El sistema debe tener un rendimiento óptimo.  \*El sistema debe desplegar los enemigos rápidamente (en un tiempo máximo de 2 segundos).  \*El sistema debe funcionar tanto para aplicaciones web como para dispositivos móviles.  \*Para esta versión, el juego debe soportar 20 jugadores a la vez. |

**\*Verde es prerrequisito de amarillo. Ya que es necesario la creación de al menos un jugador para cumplir con los requerimientos posteriores.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R1: Registrar jugadores | | |
| Resumen | Se registra un jugador, pidiéndole distintos datos por pantalla. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| id | String | No, solo es ingresado una vez por el usuario. |
| name | String | No, solo es ingresado una vez por usuario. |
|  |  |  |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Una vez el usuario haya ingresado sus respectivos datos, se registrará el jugador, se le asignará un puntaje inicial (10) y un número de vidas (5). | | |
| Resultado o postcondición | El jugador ha sido registrado. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj: El jugador <<name>> ha sido registrado**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre o identificador | R2: Registro de niveles | | | |
| Resumen | Se registran todos los niveles pidiéndole al usuario distintos datos por pantalla. | | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| numberOfLevel | int | Si el nivel ingresado por el usuario ya existe, no le permitirá al usuario agregar el nivel, la entrada no puede ser un carácter. |
| scoreRequiredToPassLevel | int | Si la entrada es un carácter se le volverá a pedir al usuario. |
|  |  |  |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Una vez el usuario haya ingresado los respectivos datos, el nivel se añadirá a la primera posición disponible del arreglo de niveles (es un arreglo de tamaño 20, en el cual 10 posiciones se encuentran inicializadas). Será decisión del usuario ingresar o no enemigos y tesoros a este nuevo nivel. | | | |
| Resultado o postcondición | El nivel ha sido registrado. | | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj=Es un mensaje que se le muestra al usuario notificándole el registro del nivel.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R3: Registra de tesoros a un nivel | | |
| Resumen | Se registran los tesoros de un nivel pidiéndole distintos datos al usuario. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| numberLevel | int | Si el nivel no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un numero de nivel valido. |
| name | String | No, solo se le pedirá una vez al usuario. |
| url | String | No, solo se le pedirá una vez al usuario. |
| score | int | Si la entrada es un carácter, se le volverá a pedir al usuario. |
| positionX | int | Se repetirá la entrada en caso de que el usuario ingrese un carácter o salga de las márgenes permitidas de la pantalla HD. |
| positionY | int | Se repetirá la entrada en caso de que el usuario ingrese un carácter o salga de las márgenes permitidas de la pantalla HD. |
| Actividades generales necesarias para obtener los resultados | Al existir un arreglo de niveles, el número ingresado por el usuario significará una posición en el arreglo y significará un nivel en específico, en esa posición del arreglo, se añadirá el tesoro con sus distintas características. 50 tesoros se encontrarán inicializados en los diferentes niveles y el tamaño máximo del arreglo de tesoros será de 75. | | |
| Resultado o postcondición | Un tesoro ha sido asociado a un nivel. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |

**\*msj= El tesoro <<name>> se ha agregado al nivel <<numberLevel>>**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R4: Registrar Enemigos | | |
| Resumen | Se registran los enemigos pidiéndole al usuario distintos datos. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| numberLevel | int | Si el nivel no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un número de nivel valido, sucederá lo mismo en caso de ingrese un carácter. |
| id | String | No, solo se le pedirá una vez al usuario. |
| type | String | Se repetirá en caso de que lo digitado por el usuario no corresponda a un tipo de enemigo. |
| scoreWin | int | Si la entrada es un carácter, se le volverá a pedir al usuario. |
| scoreLose | int | Si la entrada es un carácter, se le volverá a pedir al usuario. |
| positionX | double | Se repetirá la entrada en caso de que el usuario ingrese un carácter o salga de las márgenes permitidas de la pantalla HD. |
| positionY | double | Se repetirá la entrada en caso de que el usuario ingrese un carácter o salga de las márgenes permitidas de la pantalla HD. |
| Actividades generales necesarias para obtener los resultados | Al existir un arreglo de niveles, el número ingresado por el usuario significará una posición en el arreglo y significará un nivel en específico, en esa posición del arreglo se añadirá el enemigo con sus distintas características. 25 enemigos se encontrarán inicializados, mientras que el tamaño del arreglo será de 35. | | |
| Resultado o postcondición | Un enemigo ha sido asociado a un nivel. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= Un enemigo de tipo <<type>> se ha agregado al nivel <<numberLevel>>**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R5: Modificar el puntaje de un jugador | | |
| Resumen | Se modificará el puntaje del jugador a partir de un name. | | |
| Entradas (No hay entradas, no se requiere de ellas) | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| namePlayer | String | Si el jugador no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un “name” que se encuentre en el sistema. |
| newScore | int | Si la entrada es un carácter, se le volverá a pedir al usuario. |
|  |  |  |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Se deberá hacerle un set al atributo del puntaje del jugador, cambiándole el puntaje actual por el nuevo ingresado por el usuario. | | |
| Resultado o postcondición | El puntaje se ha modificado. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= El nuevo puntaje del jugador <<namePlayer>> es <<newScore>>.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R6: Incrementar el nivel de un jugador. | | |
| Resumen | Se incrementa o no el nivel de un jugador teniendo en cuenta su puntaje. | | |
| Entradas (No hay entradas, no se requiere de ellas) | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| namePlayer | String | Si el jugador no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un “name” que se encuentre en el sistema. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | El programa buscará el puntaje asociado al name ingresado y lo comparará con el puntaje requerido para subir de nivel. El nivel de un jugador solo se incrementará si el puntaje que tiene es mayor o igual al requerido. | | |
| Resultado o postcondición | El nivel de un jugador se ha incrementado o no. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj=El jugador <<namePlayer>> ha incrementado su nivel.**

**\*msj=El jugador <<namePlayer>> no cuenta con un puntaje suficiente para incrementar su nivel.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R7: Informar los enemigos y tesoros presentes en un nivel. | | |
| Resumen | Dado un nivel, se le muestran al usuario todos los enemigos y tesoros presentes en él. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| numberLevel | int | Si el nivel no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un número de nivel que se encuentre inicializado, sucederá lo mismo en caso de ingrese un carácter. |
| Actividades generales necesarias para obtener los resultados | Al existir un arreglo de niveles, el número ingresado por el usuario significará una posición en el arreglo y significará un nivel en específico. Mediante los “gets” presentes en las clases se obtendrán tanto los tipos de los enemigos como los nombres de los tesoros que se encuentren en esa posición del arreglo. | | |
| Resultado o postcondición | Los enemigos y tesoros han sido mostrados por pantalla. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | La salida se repetirá hasta que hayan sido encontrados todos los enemigos y tesoros de un nivel. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**msj= Un enemigo <<typeEnemy>>, Un tesoro de <<nameTreasure>>**

**Un enemigo <<typeEnemy>>, Un tesoro de <<nameTreasure>>**

**(……………………………………………………………………..)**

**han sido encontrados en el nivel <<numberLevel>>.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R8: A partir de un nombre, informar los tesoros presentes en todos los niveles. | | |
| Resumen | Dado un nombre de tesoro, el sistema busca y muestra la cantidad de tesoros asociados a ese nombre. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| nameTreasure | String | Si el tesoro no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un nombre de tesoro que se encuentre en el sistema. |
| Actividades generales necesarias para obtener los resultados | Mediante un recorrido total por todos los niveles, el sistema irá guardando todos los tesoros que concuerden con el nombre ingresado por el usuario. Al final, se le muestra al usuario el número de tesoros asociado a ese nombre. | | |
| Resultado o postcondición | Todos los tesoros han sido mostrados por pantalla. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= La cantidad de tesoroes <<nameTreasure>> presentes en todos los niveles son <<quantyOfTreasure>>**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R9: A partir de un tipo, informar los enemigos presentes en todos los niveles. | | |
| Resumen | Dado un tipo de enemigo, el sistema busca y muestra la cantidad de tesoros asociados a ese nombre. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| typeEnemy | String | Si el enemigo no existe, el programa le mostrará un mensaje al usuario hasta que ingrese un tipo de enemigo que se encuentre en el sistema. |
| Actividades generales necesarias para obtener los resultados | Mediante un recorrido total por todos los niveles, el sistema irá guardando todos los enemigos que concuerden con el tipo ingresado por el usuario. Al final se le muestra al usuario el número de enemigos encontrados de ese tipo durante todo el recorrido. | | |
| Resultado o postcondición | Todos los enemigos han sido mostrados por pantalla. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | No. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= La cantidad de enemigos de tipo <<typeEnemy>> presentes en todos los niveles son <<quantyOfEnemy>>**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R10: Informar el tesoro más repetido en todos los niveles. | | |
| Resumen | Se le muestra al usuario cual es el tesoro más repetido en todo los niveles. | | |
| Entradas (No hay entradas, el sistema hace un proceso interno) | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Mediante un recorrido total por todos los niveles, el sistema irá guardando la cantidad de veces que se repitan cada uno de los diferentes tesoros. Después de ello, se le muestra al usuario el o los tesoros más repetidos. | | |
| Resultado o postcondición | Se muestra al usuario el tesoro más repetido en todos los niveles. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | La salida se puede repetir en caso de que sean varios los tesoros que más se repiten. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= El tesoro <<nameTreasure>>**

**(……………………………..)**

**Es o son los más repetidos en todos los niveles.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R11: Informar el enemigo que otorga más puntaje y el nivel donde se ubica. | | |
| Resumen | Se le muestra al usuario cual es el enemigo más valioso de todo el juego y el nivel donde se ubica. | | |
| Entradas (No hay entradas, el sistema hace un proceso interno) | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Teniendo en cuenta las características de puntaje de cada enemigo, se realizará un recorrido total por todos los niveles, el sistema irá guardando y rescribiendo el enemigo si encuentra uno más “valioso” que el anterior, una vez terminado el recorrido, se le hará un “get” al nivel donde se aloje. Se le mostrará toda esta información al usuario. | | |
| Resultado o postcondición | Se muestra al usuario el enemigo más “valioso” de todo el juego. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | La salida se puede repetir en caso de que sean varios los enemigos que otorguen un mayor puntaje. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**msj= El enemigo <<typeEnemy>> se encuentra en el nivel <<numberLevel>>**

**El enemigo <<typeEnemy>> se encuentra en el nivel <<numberLevel>>**

**(………………………………………………………………………………)**

**Es o son los que otorgan más puntaje en todo el juego.**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R12: Informar la cantidad de consonantes presentes en los nombres de los enemigos. | | |
| Resumen | Se le muestra al usuario la cantidad de consonantes presentes en los nombres de los enemigos presentes en el juego. | | |
| Entradas (No hay entradas, el sistema hace un proceso interno) | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Mediante un recorrido total por la String de los nombres, el sistema irá guardando, aquellas que no son vocales. Al final se le muestra al usuario la cantidad de consonantes encontradas en el nombre de un enemigo. | | |
| Resultado o postcondición | Se muestra al usuario las consonantes presentes en todos los enemigos del juego. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | La salida se repetirá hasta que se le realice el mismo proceso con todos los enemigos. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= El enemigo <<typeEnemy>> tiene <<quantyConsonts>> en su nombre.**

**El enemigo <<typeEnemy>> tiene <<quantyConsonts>> en su nombre.**

**(……………………………………………………………………………..)**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | R13: Informar el top 5 de jugadores de acuerdo al puntaje. | | |
| Resumen | Se le muestra al usuario los 5 mejores jugadores de acuerdo a su puntaje. | | |
| Entradas (No hay entradas, el sistema hace un proceso interno) | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Mediante un recorrido total por el arreglo de jugadores se obtendrán los 5 mejores puntajes que existan, así como como el respectivo nombre de los jugadores que se encuentren asociados a cada puntaje, en caso de que existan más puntajes, el sistema obviaría los que se encuentren por debajo de ese top. | | |
| Resultado o postcondición | Se muestra al usuario los mejores 5 puntajes de todo el juego. | | |
| Salidas | Nombre salida | Tipo de dato | Condición de selección o repetición |
| msj | String | La salida se repetirá 5 veces, para mostrar el top 5 de jugadores. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\*msj= El jugador <<namePlayer>> tiene un puntaje de <<Score>> y es el top 1.**

**El jugador <<namePlayer>> tiene un puntaje de <<Score>> y es el top 2.**

**(……………………………………………………………………………….)**

**El jugador <<namePlayer>> tiene un puntaje de <<Score>> y es el top 5.**

**Diagramas de trazabilidad:**

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Nombre de la Clase** | **Nombre del método** |
| R1:  Registrar jugador | Clase Main | ExecuteOption(option) |
| Clase Videogame | addPlayer(id, nickname, score, numberOfLives): String |
| Clase Player | Player(…) |
| R2:  Registrar niveles | Clase Main | ExecuteOption(option) |
| Clase Videogame | addLevel(number, requiredScore, dificulty) |
| Clase Level | Level(…) |
| R3:  Registrar tesoros a un nivel | Clase Main | ExecuteOption(option) |
| Clase Videogame | addTreasureToLevel(numberLevel, name, url, score, positionX, positionY) |
| Clase Level | addTreasure(Treasure) |
| Clase Treasure | Treasure(…) |
| R4:  Registrar enemigos a un nivel | Clase Main | ExecuteOption(option) |
| Clase Videogame | addEnemyToLevel(numberLevel, id, type, scoreIfWin, scoreIfLose, positionX, positionY) |
| Clase Level | addEnemy(Enemy) |
| Clase Enemy | Enemy(…) |
| R5:  Modificar el puntaje de un jugador | Clase Main | ExecuteOption(option) |
| Clase Videogame | setPlayerScore(namePlayer, newScorePlayer) |
| Clase Player | setScore(newScore) |
| R6:  Incrementar el nivel de un jugador | Clase Main | ExecuteOption(option) |
| Clase Videogame | LevelUpPlayer(namePlayer) |
| Clase Level | getRequiredScore() |
| Clase Player | getScore() |
| setNumberLevel() |
| R7: Informar los enemigos y tesoros presentes en un nivel. | Clase Main | executeOption(option) |
| Clase Videogame | showTreasuresAndEnemies(numberLevel) |
| Clase Level | searchTreasuresAndEnemies() |
|  | Clase Enemy | getType() |
|  | Clase Treasure | getName() |
| R8: A partir de un nombre, informar la cantidad de un tesoro en todos los niveles. | Clase Main | executeOption(option) |
|  | Clase Videogame | showTreasuresInGame(nameTreasure) |
|  | Clase Level | countTreasures(nameTreasure) |
|  | Clase Treasure | getName() |
| R9:  A partir de un tipo, informar la cantidad de un enemigo encontrado en todos los niveles. | Clase Main | executeOption(option) |
| Clase Videogame | showEnemiesInGame(typeEnemy) |
| Clase Level | countEnemy(typeEnemy) |
| Clase Enemy | getType() |
| R10:  Mostrar el tesoro más repetido en todos los niveles | Clase Main | executeOption(option) |
| Clase Videogame | searchMostRepeatenTreasureInGame() |
| Clase Level | searchMostRepeatenTreasure() |
| Clase Treasure | getName() |
| R11:  Mostrar el enemigo que otorgue más puntaje en todo el juego | Clase Main | executeOption(option) |
| Clase Videogame | searchMostValuableEnemyInGame() |
| Clase Level | searchMostValuableEnemy() |
| Clase Enemy | getScoreIfLose() |
| getType() |
| R12:  Mostrar las consonantes presentes en los nombres de los enemigos | Clase Main | executeOption(option) |
| Clase Videogame | showCountConsonantsEnemies() |
| Clase Level | countConsonantsEnemies() |
| Clase Enemy | getName() |
| R13:  Mostrar el top 5 de jugadores de acuerdo al puntaje | Clase Main | executeOption(option) |
| Clase Videogame | showTopFivePlayers() |
| Clase Player | getName() |
|  | getScore() |