1. Requerimientos Funcionales

|  |  |
| --- | --- |
| **Nombre:** | R. #1. Permitir leer información |
| **Resumen:** | Permitir leer la información de los posibles participantes desde un archivo de texto delimitado por comas |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Lectura de información de los participantes |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #2. Permitir cargar información |
| **Resumen:** | Permitir cargar la información para que se puedan hacer búsquedas a través del identificador del registro del evento |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Información cargada |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #3. Permitir hacer búsquedas |
| **Resumen:** | Permitir hacer búsquedas por id para cada una de las estructuras creadas. |
| Entradas: | Jugador ID |
|  |  |
| Resultados: | Búsqueda realizada |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #4. Indicar si el espectador se encuentra |
| **Resumen:** | Indicar si es espectador se encuentra entre los buscados y su tiempo que tardo encontrandolo |
| Entradas: | Jugador ID |
|  |  |
| Resultados: | SI: Jugador encontrado, su info y el tiempo en que se tardo |
|  | NO: Jugador no encontrado y tiempo. |

|  |  |
| --- | --- |
| **Nombre:** | R. #5. Generar el máximo de filas |
| **Resumen:** | Se deben generar el máximo de filas para buenas pruebas |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Buena prueba |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #6. Exportar un archivo CSV |
| **Resumen:** | Permitir exportar un archivo CSV el cual delimita los archivos de texto |
| Entradas: | Ninguno |
|  |  |
| Resultados: | Archivo CSV |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #7. Permitir ver estructura |
| **Resumen:** | Permitir ver la estructura del árbol binario en como están organizados los jugadores |
| Entradas: | Ninguno |
|  |  |
| Resultados: | Árbol binario |
|  |  |

1. Diagrama de Clases

Imagen que contiene captura de pantalla

Descripción generada automáticamente

3. Diseños de casos de pruebas unitarias

Configuración de Escenarios

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1() | VoleyEvent | Se crea este escenario con un nuevo objeto tipo Participant para permitir crear los arboles respetivos y hacer pruebas de Agregar, Buscar, Crear y Recorrer |

Diseño de casos de prueba

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar la correcta creación del árbol binario | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| VoleyEvent | Load() | setupScenary1() | Ninguno | True  Se carga correctamente el archivo y permite crear el árbol con los datos contenidos en el CSV |
| VoleyEvent | addParticipantToTree() | setupScenary1() | Id = 0 firstName = “Juan”  LastName = “Hernandez”  Email = “juanher0825@gmail.com”  Gender = “Masculino”  Country = “Colombia”  Photo = null  Birthday = “08/25/2000” | True  Permite agregar correctamente un participante nuevo predeterminado. |
| VoleyEvent | searchParticipant() | setupScenary1() | Ninguna | True  Permite buscar correctamente a un jugador dependiendo del id ingresado |
| VoleyEvent | addParticipantToTreeWithFirst | setupScenary1() | Id = 0 firstName = “Juan”  LastName = “Hernandez”  Email = “juanher0825@gmail.com”  Gender = “Masculino”  Country = “Colombia”  Photo = null  Birthday = “08/25/2000”  Id = 1 firstName = “Daniela”  LastName = “Perez”  Email = “juanher0825@gmail.com”  Gender = “Masculino”  Country = “Colombia”  Photo = null  Birthday = “08/25/2000” | True  Permite agregar correctamente un participante en el árbol binario después de haber agregado uno anteriormente. |