**1.**

**Requerimientos Funcionales:**

|  |  |
| --- | --- |
| **Nombre** | RF1: Cargar los espectadores desde un archivo |
| **Resumen** | Permite cargar todos los candidatos del evento mediante la ruta de un archivo. La búsqueda del archivo se podrá realizar tanto con su ruta relativa, como con su ruta absoluta. Si no se encuentra el archivo indicara que no se pudieron cargar los datos. |
| **Entrada** | La ruta del archivo. |
| **Salida** | Se han cargado los espectadores del evento. |

|  |  |
| --- | --- |
| **Nombre** | RF2: Cargar aleatoriamente los participantes |
| **Resumen** | Carga todos los participantes del evento mediante la selecciona aleatoria de algunos espectadores. Se seleccionará aproximadamente el 50% de los espectadores para ser los competidores del evento. Cada que se escoge un participante, este se eliminara de los espectadores. |
| **Entrada** | Ninguna. |
| **Salida** | Se han seleccionado los participantes del evento. |

|  |  |
| --- | --- |
| **Nombre** | RF3: Buscar espectador por id |
| **Resumen** | Permite buscar a cualquier espectador y mostrar su información mediante su id. Si este no existe se indicará su inexistencia. |
| **Entrada** | El id del espectador. |
| **Salida** | La información del espectador. |

|  |  |
| --- | --- |
| **Nombre** | RF4: Buscar participante por id |
| **Resumen** | Permite buscar a cualquier participante y mostrar su información mediante su id. Si este no existe se indicará su inexistencia. |
| **Entrada** | El id del participante. |
| **Salida** | La información del participante. |

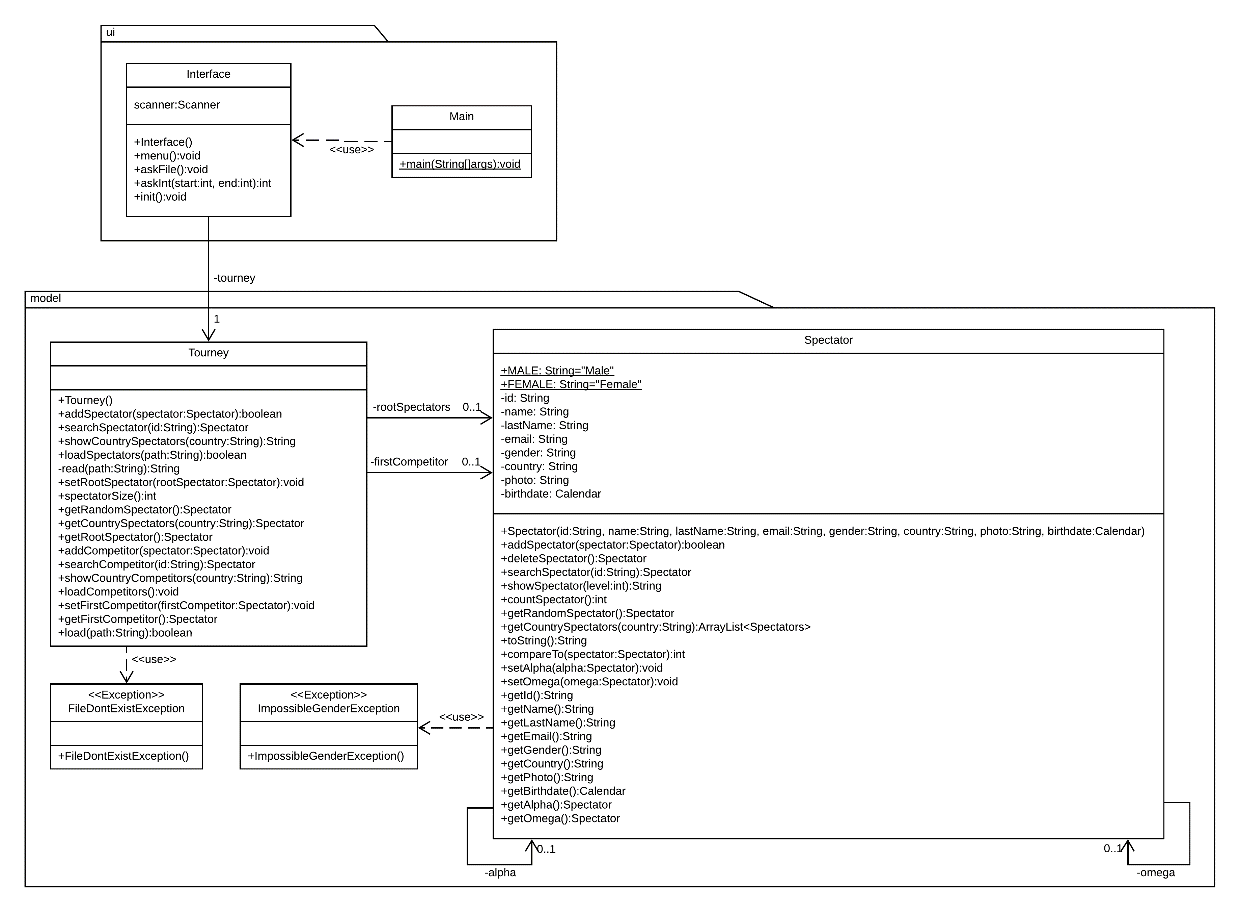
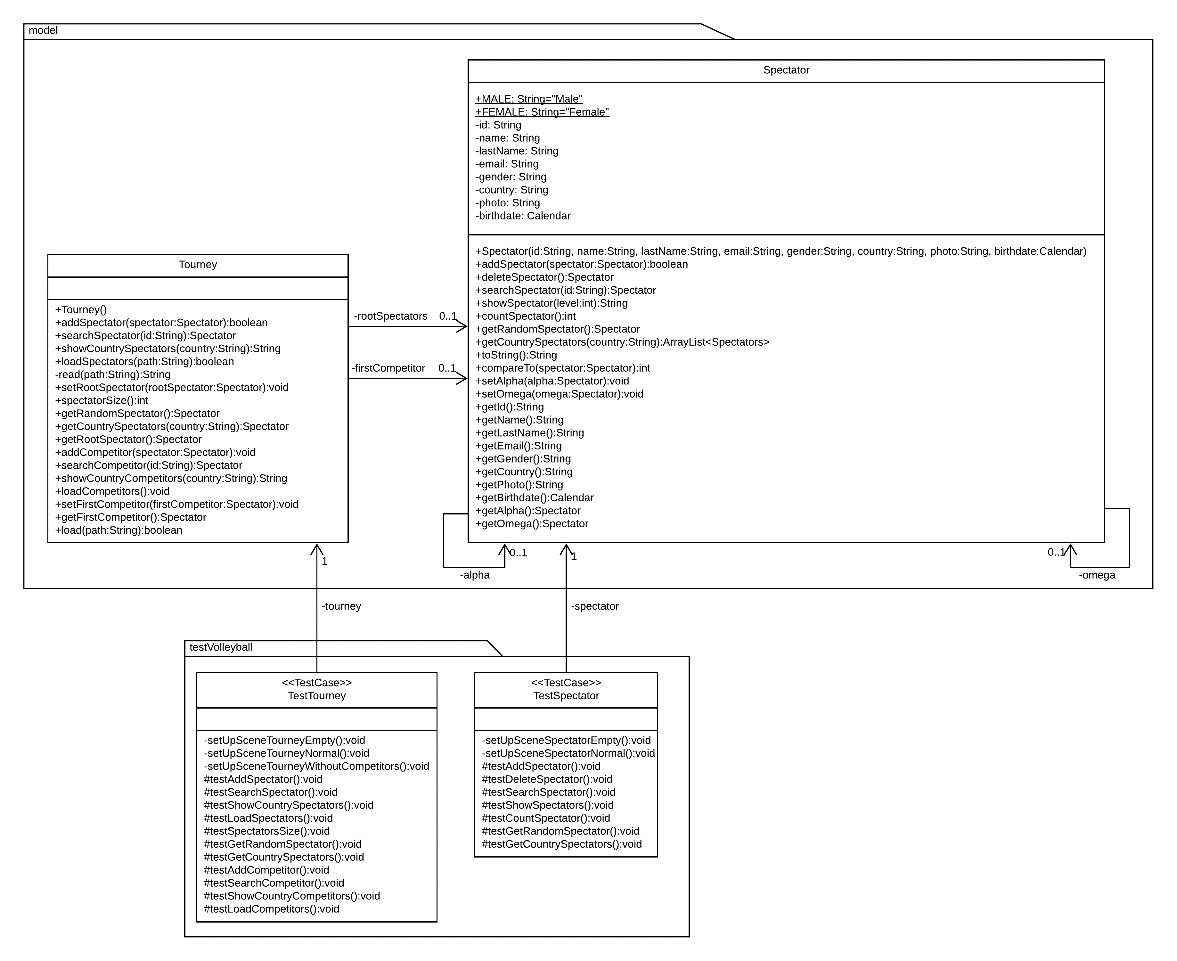
|  |  |
| --- | --- |
| **Nombre** | RF5: Mostrar los espectadores de un país |
| **Resumen** | Permite mostrar todos los participantes de un país. Estos se mostrarán en su debida estructura. |
| **Entrada** | El país que se desea mostrar. |
| **Salida** | Los espectadores de ese país. |

|  |  |
| --- | --- |
| **Nombre** | RF5: Mostrar los participantes de un país |
| **Resumen** | Permite mostrar todos los participantes de un país. Estos se mostrarán en su debida estructura. |
| **Entrada** | El país que se desea mostrar. |
| **Salida** | Los espectadores de ese país. |

**Requerimientos No Funcionales:**

* La información se cargará mediante archivos planos.
* Los espectadores deben ser implementado en un árbol binario de búsqueda (ABB).
* Los competidores deben ser implementados en una lista doblemente enlazada.
* Mostrar los tiempos del sistema de los métodos de búsqueda.
* Todos los métodos de ABB deben ser implementados recursivamente.

**2.**

****

**4.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | | **Valores entrada** | **Resultado** | | |
| Tourney | addSpectator | Un evento sin espectadores, ni competidores. | | Se añadió un espectador con id “3”. Luego, otro espectador con id ”1”. Luego, un espectador con id ”5”.  Finalmente, un espectador con id “3”. | Se pudieron añadir exitosamente todos los espectadores excepto el ultimo. El espectador con id “3” tiene a su lado izquierdo el espectador con id “1” y a su lado derecho el de id “5”. | | |
| Tourney | searchSpectator | Un evento sin espectadores, ni competidores. | Se busca el espectador con id “ola”. | | | | No se encuentra el espectador con ese id. |
| Un evento con 3 espectadores de id ”3”, “1” y “5”. | Se buscan los espectadores con id “3”, “1”, “5” y “2”. | | | | Se encuentran todos los espectadores con ese id, excepto el de id ”2”. |
| Tourney | showCountrySpectators | Un evento con 3 espectadores de nombres “Esteban”, que tiene a su izquierda a “Mateo” y que tiene a su derecha a “Johan”. Todos pertenecen a “Colombia”. | | Se muestra el árbol de “Colombia”. | Primero sale “Esteban”. Luego, (Un poco más al lado derecho) sale “Mateo”. Finalmente, (Justo debajo del anterior) sale “Johan”. | | |
| Tourney | loadSpectators | Un evento sin espectadores, ni competidores. | | Primero, se trata cargar un archivo de ruta “LoLXd”. Luego, se trata cargar un archivo de ruta “data/test.txt”. | La primera carga no va a ser posible, porque el archivo no existe. La segunda carga se realizará; el primer espectador tendrá id “6” y habrá un total de 7 espectadores. | | |
| Tourney | spectatorSize | Un evento sin espectadores, ni competidores. | |  | | Hay 0 espectadores en el evento. | |
| Un evento con un total de 3 espectadores. | |  | | Hay 3 espectadores en el evento. | |
| Tourney | getRandomSpectator | Un evento con 3 espectadores de id ”3”, que tiene a su izquierda a un espectador de id “1” y a su derecha uno de id “5”. | | Se seleccionan aleatoriamente a dos de los espectadores. | Los dos espectadores van a ser diferentes y solo quedara un espectador en el evento. | | |
| Tourney | getCountrySpectators | Un evento con 3 espectadores de id ”3”, que tiene a su izquierda a un espectador de id “1” y a su derecha uno de id “5”. Todos pertenecen a “Colombia”. | | Se pide el árbol de “Colombia”. | La raíz del árbol va a ser el espectador de id “3”. A su izquierda estará el espectador de id “1” y a su derecha el de id “5”. | | |
| .Tourney | addCompetitor | Un evento sin espectadores, ni competidores. | | Se añadió un competidor con id “3”. Luego, otro competidor con id ”1”. Finalmente, un competidor con id ”5”. | Los 3 competidores se añadieron exitosamente en el orden: “5”, “1” y “3”. | | |
| Tourney | searchCompetitor | Un evento con 3 competidores de id ”3”, “1” y “5”. | | Se buscan los competidores con id “3”, “1”, “5” y “2”. | Se encuentran todos los competidores con ese id, excepto el de id ”2”. | | |
| Tourney | showCountryCompetitors | Un evento con 3 competidores de nombre “Esteban”, “Mateo” y “Johan”. Se encuentran en ese respectivo orden. | | Se muestra la lista de “Colombia”. | Primero sale “Esteban”. Luego, sale “Mateo”. Finalmente, sale “Johan”. | | |
| Tourney | loadCompetitors | Un evento con 5 espectadores. | | Se seleccionan aleatoriamente los competidores del evento. | Ahora hay 3 espectadores y 2 competidores en el evento. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Spectator | addSpectator | Un espectador de id “6” sin más espectadores. | Se añadió un espectador con id “7”. Luego, otro espectador con id ”5”.  Finalmente, un espectador con id “6”. | Se añadió el espectador de id “7” a la derecha, el espectador de id “5” a la izquierda y el espectador de id “6” no se pudo añadir. |
| Spectator | deleteSpectator | Un espectador sin más espectadores. |  | Retorna un valor nulo. |
| Un espectador con dos espectadores; uno de id “5” y otro de id “7”. |  | Retorna a uno de los hijos que añade ordenadamente al otro de los hijos. |
| Spectator | searchSpectator | Un espectador de id “6” con dos espectadores; uno de id “5” y otro de id “7”. | Se buscan los espectadores con id “6”, “5”, “7” y “4”. | Se encuentran todos los espectadores con ese id, excepto el de id ”4”. |
| Spectator | showSpectators | Un espectador de nombre “Esteban” con dos espectadores; “Mateo” se encuentra a la izquierda y “Johan” se encuentra a la derecha. |  | Primero sale “Esteban”. Luego, (Un poco más al lado derecho) sale “Mateo”. Finalmente, (Justo debajo del anterior) sale “Johan”. |
| Spectator | countSpectator | Un espectador sin más espectadores. |  | Hay 1 espectadores. |
| Un espectador con 2 espectadores. |  | Hay 3 espectadores. |
| Spectator | getRandomSpectator | Un espectador con 2 espectadores. | Se seleccionan aleatoriamente a dos de los espectadores. | Los 2 espectadores van a ser diferentes. |
| Spectator | getCountrySpectators | Un espectador de id “6” con dos espectadores; uno de id “5” y otro de id “7”. Todos pertenecen a “Colombia”. | Se pide la lista de espectadores “Colombia”. Luego, Se pide la lista de espectadores “Japón”. | En la primera lista estarán 3 espectadores de id “6”, “5” y “7” en ese respectivo orden. Mientras que la segunda lista se encontrara vacía. |