





ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES

Nombre: <i>RF1 – Cargar información</i>	
Resumen:	<i>El sistema permite leer y cargar la información de los espectadores del evento (La información se encontrará en un archivo de texto). Una vez cargado los espectadores, estos se almacenarán en estructura de datos (ABB). Después, se seleccionará aleatoriamente al 50% de los espectadores, y se guardarán en otra estructura de datos (Lista doblemente enlazada).</i>
Entrada:	<i>La ruta relativa del archivo de texto con la información.</i>
Salida:	<i>La información se ha cargado satisfactoriamente.</i>

Nombre: <i>RF2 – Buscar a un espectador</i>	
Resumen:	<i>El sistema permite buscar a un espectador por el id e indicarle al usuario si se encontró o no al espectador, en caso de ser encontrado se mostrará su información. Además, se mostrará el tiempo que haya tardado la búsqueda (se halla encontrado o no).</i>
Entrada:	<i>El id del espectador.</i>
Salida:	<i>La información del espectador y el tiempo que tardó la búsqueda.</i>

Nombre:	RF3 – Buscar a un participante
Resumen:	<i>El sistema permite buscar a un espectador por el id e indicarle al usuario si se encontró o no al espectador, en caso de ser encontrado se mostrará su información. Además, se mostrará el tiempo que haya tardado la búsqueda (se halla encontrado o no).</i>
Entrada:	<i>El id del participante.</i>
Salida:	<i>La información del participante y el tiempo que tardó la búsqueda.</i>

Nombre:	RF4 – Mostrar espectadores de un país
Resumen:	<i>El sistema permite mostrar todos los espectadores de un país determinado</i>
Entrada:	<i>El país del que se desea ver sus espectadores</i>
Salida:	<i>Los espectadores del país</i>

Nombre:	RF5 – Mostrar participantes de un país
Resumen:	<i>El sistema permite mostrar todos los participantes de un país determinado</i>
Entrada:	<i>El país del que se desea ver sus participantes</i>
Salida:	<i>Los participantes del país</i>

ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES

<i>Nombre:</i>	<i>RNF1 – Usar ABB</i>
<i>Resumen:</i>	<i>Los espectadores deben ser almacenados en un ABB cuyo criterio de ordenamiento será el id.</i>

<i>Nombre:</i>	<i>RNF – Usar listas doblemente enlazadas</i>
<i>Resumen:</i>	<i>Los competidores deben ser almacenados en un una doblemente enlazada</i>

<i>Nombre:</i>	<i>RNF – Usar recursividad</i>
<i>Resumen:</i>	<i>Todos los métodos que administren los ABB, deben ser implementados usando recursividad</i>

<i>Nombre:</i>	<i>RNF – Cargar de un archivo plano</i>
<i>Resumen:</i>	<i>La información del programa debe ser cargada desde un archivo plano</i>

CASOS DE PRUEBA

Caso de prueba 1

Clase	Método	Escenario	Valores de entrada	Resultado
Event	Event()	Un ABB con 9 espectadores y otro ABB vacío	La ruta del archivo con la información de los espectadores	Verdadero, cuando se introduce una ruta válida se crea el objeto evento, por otro lado, si no es así, no se crea el objeto

Caso de prueba 2

Clase	Método	Escenario	Valores de entrada	Resultado
Event	getSpectatorById()	Un ABB con 9 espectadores y otro ABB vacío	El id del espectador que se desea obtener del ABB	Verdadero, se retorna el espectador esperado

Caso de prueba 3

Clase	Método	Escenario	Valores de entrada	Resultado
Event	getSizeBST()	Un ABB con 9 espectadores y un ABB vacío al cual	Los atributos de los espectadores que se	Verdadero, el método retorna el número esperado.

		se le agregarán espectadores	agregarán al ABB	
--	--	------------------------------	------------------	--

Caso de prueba 4				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	getFather()	Un ABB con 9 espectadores y otro vacío	Los espectadores a los que les queremos hallar sus padres en el ABB	Verdadero, el método retorna el padre del espectador

Caso de prueba 5				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	deleteSpectator()	Un ABB con 9 espectadores	Los espectadores que se van a borrar del ABB	Verdadero, los espectadores fueron eliminados correctamente

Caso de prueba 6				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	addCompetitors()	Un ABB con 10 espectadores y otro con 9 espectadores	Ninguno	Verdadero, el 50% seleccionado aleatoriamente

				de los espectadores fueron agregados a la lista doblemente enlazada de competidores
--	--	--	--	---

Caso de prueba 7				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	GetCompetitorById()	Una lista doblemente enlazada con 4 competidores	Los id de los competidores	Verdadero, el método retorna los competidores esperados

Caso de prueba 8				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	searchSpectator()	Un ABB con 9 espectadores	Los id de los espectadores	Verdadero, el método retorna el mensaje esperado

Caso de prueba 9				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	searchCompetitor()	Una lista doblemente	Los id de los competidores	Verdadero, el método retorna

		enlazada con 4 competidores	que desea buscar	el mensaje esperado
--	--	-----------------------------	------------------	---------------------

Caso de prueba 10				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	showCompetitorsByCountry()	Una lista doblemente enlazada con 9 competidores	El país del que se desea mostrar los competidores	Verdadero, el método retorna el mensaje esperado

Caso de prueba 11				
Clase	Método	Escenario	Valores de entrada	Resultado
Event	ShowSpectatorsByCountry()	Un ABB con 9 espectadores	El país del cual se desea hallar sus espectadores	Verdadero, el árbol se imprime como se esperaba

TRAZABILIDAD

RF1 – Cargar información

<i>CLASE</i>	<i>MÉTODO</i>
<i>Event</i>	loadData()
<i>Event</i>	addCompetitors()
<i>Event</i>	selectRandomSpectator()
<i>Event</i>	deleteSpectator()
<i>Event</i>	addCompetitor()
<i>Spectator</i>	deleteSpectator()
<i>Spectator</i>	reAddSpectator()
<i>Spectator</i>	compareTo()
<i>Spectator</i>	hasChildren()
<i>Spectator</i>	getFather()
<i>Competitor</i>	addCompetitor()

RF2 – Buscar a un espectador

<i>CLASE</i>	<i>MÉTODO</i>
<i>Event</i>	searchSpectator()

<i>Event</i>	getSpectatorById()
<i>Spectator</i>	getSpectatorById()
<i>Spectator</i>	toString()

<i>RF3 – Buscar a un participante</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Event</i>	searchCompetitor()
<i>Event</i>	getCompetitorById()
<i>Competitor</i>	getCompetitorById()
<i>Competitor</i>	toString()

<i>RF4 – Mostrar espectadores de un país</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Event</i>	showSpectatorsByCountry()
<i>Event</i>	getSpectatorsByCountry()
<i>Event</i>	createBSTByCountry()
<i>Spectator</i>	createBSTByCountry()
<i>Spectator</i>	reAddSpectator()

<i>Spectator</i>	<u>showSpectators()</u>
<i>Spectator</i>	<u>showSpectator()</u>

<i>RF5 – Mostrar participantes de un país</i>	
<i>CLASE</i>	<i>MÉTODO</i>
<i>Event</i>	showCompetitorsByCountry()
<i>Competitor</i>	showCompetitorsByCountry()
<i>Competitor</i>	showCompetitor()