LABORATORIO 6

1. Requerimientos funcionales

|  |
| --- |
| *RF #1. Leer la información de los posibles asistentes al evento desde un archivo de texto* |
| Resumen: la información de los asistentes deben leerse desde un archivo de texto y el programa al cargar esta información debe estar delimitada por comas para la identificación de cada asistente. |
| Entradas: no tiene entradas. |
| Salidas: el programa carga la información de los asistentes desde el archivo de texto. |

|  |
| --- |
| *RF #2. La información se carga en una estructura de datos de tipo árbol binario de búsqueda (ABB)* |
| Resumen: a partir de la lectura de los datos, el programa debe organizar a cada uno de los posibles asistentes al evento en una estructura de datos de tipo ABB cuyo criterio de orden estará dado por el id de registro al evento. |
| Entradas: información de los posibles asistentes. |
| Salidas: el programa carga la información en una estructura de datos tipo ABB. |

|  |
| --- |
| *RF #3. Realizar búsquedas a través del identificador del registro del evento en las dos estructuras de datos creadas (ABB y lista doblemente enlazada).* |
| Resumen: el programa indica, en cualquier caso, si el espectador se encuentra o no entre los buscados en su respectiva estructura de datos; así como el tiempo que tardó la búsqueda. Si el espectador fue encontrado, deben visualizarse sus datos en un panel a la izquierda. |
| Entradas: no tiene entradas. |
| Salidas: el programa indica si el espectador fue encontrado o no, junto al tiempo que duró la búsqueda, en un panel. |

|  |
| --- |
| *RF #4. Seleccionar aleatoriamente a los asistentes y agregarlos en una lista doblemente enlazada.* |
| Resumen: una vez cargada la información en el árbol de posibles asistentes, se seleccionará aleatoriamente el 50% de esos espectadores para agregarlos a una lista doblemente enlazada, que será la estructura de datos que almacenará los inscritos al evento o participantes al evento. |
| Entradas: no tiene entradas. |
| Salidas: el programa agrega a los asistentes escogidos aleatoriamente en una lista doblemente enlazada. |

|  |
| --- |
| *RF #5. Visualizar los datos de las dos estructuras de un país determinado* |
| Resumen: el programa también contará con una opción que podrá visualizar los datos de las dos estructuras de un país determinado, donde la estructura del árbol binario de búsqueda se verá como un árbol y la lista se visualizará en forma consecutiva. |
| Entradas: no tiene entradas. |
| Salidas: el programa permite que el usuario visualice a los asistentes y espectadores, según un país determinado y la estructura de datos indicada. |

1. Tabla de trazabilidad del diseño

|  |  |
| --- | --- |
| Requerimiento | Método(s) con la clase que lo solucionan |
| *RF #1. Leer la información de los posibles asistentes al evento desde un archivo de texto* |  |
| *RF #2. La información se carga en una estructura de datos de tipo árbol binario de búsqueda (ABB)* |  |
| *RF #3. Realizar búsquedas a través del identificador del registro del evento en las dos estructuras de datos creadas (ABB y lista doblemente enlazada)..* |  |
| *RF #4. Seleccionar aleatoriamente a los asistentes y agregarlos en una lista doblemente enlazada.* |  |
| *RF #5. Visualizar los datos de las dos estructuras de un país determinado* |  |