**Documentación**

**Laboratorio 6**

**Requerimientos funcionales:**

|  |  |
| --- | --- |
| Nombre | R0: leer los asistentes desde un archivo .csv |
| Resumen | Los posibles asistentes se leen desde un archivo de texto separado por comas en donde se contiene toda su información |
| Entrada | N/A |
| Salida | Ruta : String |

|  |  |
| --- | --- |
| Nombre | R1: cargar los asistentes a un árbol binario ordenado |
| Resumen | A partir de la ruta del archivo, obtener los datos, crear personas con estos datos e ingresarlos en un árbol binario ordenado |
| Entrada | Ruta : String |
| Salida | Árbol binario ordenado con los asistentes |

|  |  |
| --- | --- |
| Nombre | R2: agregar de forma aleatoria la mitad de los asistentes a una lista que representa los participantes |
| Resumen | En una lista doblemente enlazada se almacenan los participantes, los cuales tomados aleatoriamente de los asistentes |
| Entrada | N/A |
| Salida | Lista doblemente enlazada con los participantes |

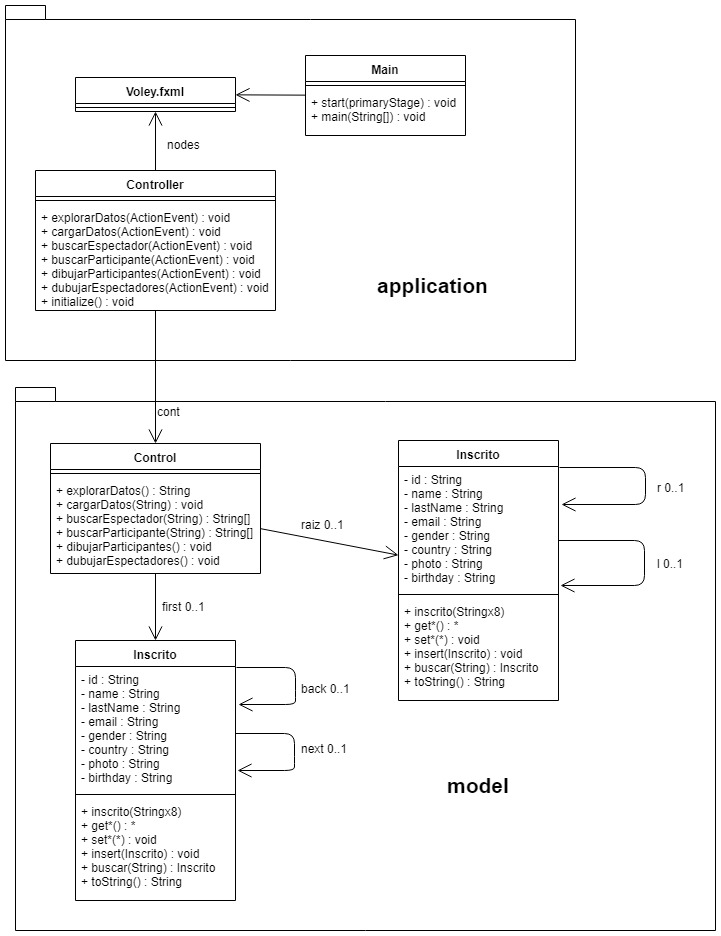
|  |  |
| --- | --- |
| Nombre | R3: Realizar búsquedas de asistentes y visualizar lo encontrado |
| Resumen | A partir del número de identificación personal, realizar una búsqueda en el árbol binario con el fin de encontrar a una persona especifica |
| Entrada | ID : String |
| Salida | Se muestra la información de la persona buscada, en caso de que esta exista en el árbol |

|  |  |
| --- | --- |
| Nombre | R4: realizar búsquedas de participantes y visualizar lo encontrado |
| Resumen | A partir del número de identificación personal, realizar una búsqueda en la lista doblemente enlazada con el fin de encontrar a una persona especifica |
| Entrada | ID : String |
| Salida | Se muestra la información de la persona buscada, en caso de que esta exista en la lista |

|  |  |
| --- | --- |
| Nombre | R5: dibujar, según el país, los asistentes del evento |
| Resumen | En un espacio asignado, se dibuja de forma conceptual el árbol según el país elegido |
| Entrada | Pais : String |
| Salida | Dibujo conceptual del árbol binario ordenado |

|  |  |
| --- | --- |
| Nombre | R6: dibujar, según el país, los participantes del evento |
| Resumen | En un espacio asignado, se dibuja de forma conceptual la lista según el país elegido |
| Entrada | Pais : String |
| Salida | Dibujo conceptual de la lista doblemente enlazada |

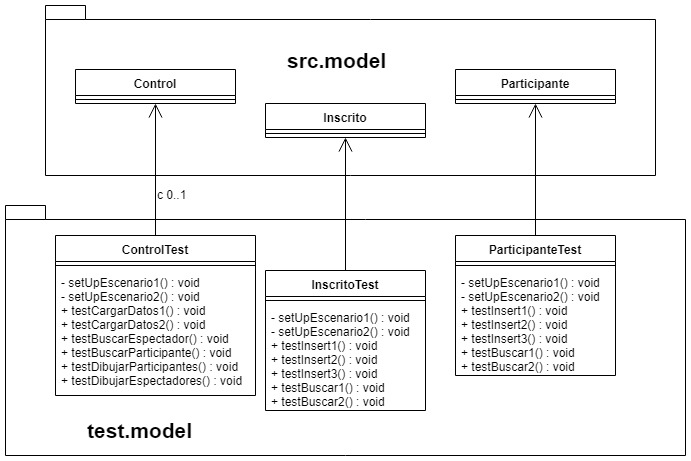
**Diagrama de clases:**

****

**Tabla de trazabilidad del análisis al diseño:**

|  |  |  |
| --- | --- | --- |
| Requerimiento | Clases | Métodos |
| R0: leer los asistentes desde un archivo .csv | Controller | explorarDatos(ActionEvent) : void |
| Control | explorarDatos() : String |
| R1: cargar los asistentes a un árbol binario ordenado | Controller | cargarDatos(ActionEvent) : void |
| Control | cargarDatos() : String |
| Inscrito | Insert(Inscrito) : void |
| R2: agregar de forma aleatoria la mitad de los asistentes a una lista que representa los participantes | Controller | cargarDatos(ActionEvent) : void |
| Control | cargarDatos() : String |
| Participante | Insert(Participante) : void |
| R3: Realizar búsquedas de asistentes y visualizar lo encontrado | Controller | buscarEspectador(ActionEvent) : void |
| Control | buscarEspectador(String) : String[] |
| Inscrito | buscar(String) : Inscrito  toString() : String |
| R4: realizar búsquedas de participantes y visualizar lo encontrado | Controller | buscarParticipante(ActionEvent) : void |
| Control | buscarParticipante(String) : String[] |
| Participante | buscar(String) : Participante  toString() : String |
| R5: dibujar, según el país, los asistentes del evento | Controller | dibujarEspectadores(ActionEvent) : void |
| Control | dibujarEspectadores() : void |
| R6: dibujar, según el país, los participantes del evento | Controller | dibujarParticipantes(ActionEvent) : void |
| Control | dibujarParticipantes() : void |

**Diseño de casos de prueba:**

****