



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador Secuencia<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas
2014



MANUAL DE USUARIO: Simulador “Secuencia<T>”

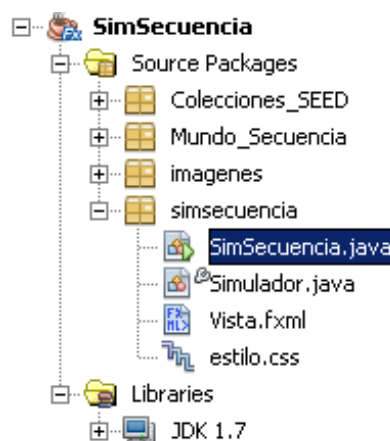
Descripción General

El presente Manual de usuario pretende describir en detalle el conjunto de funcionalidades de la Aplicación desarrollada para la Simulación del comportamiento de la Estructura de Datos Secuencia<T>.

Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Secuencia: Insertar, Eliminar, Buscar y Editar datos dentro de cada una de las posiciones de la misma.

Adicionalmente el estudiante podrá conocer las propiedades más importantes de la Secuencia, como lo son su capacidad de almacenar datos y la cantidad de datos (tamaño) almacenados dentro de la Secuencia en un momento determinado.

Para la implementación de este Simulador se ha determinado la siguiente distribución de paquetes, ya conocida por el Estudiante, de forma que sea fácilmente apropiable a futuras modificaciones con el fin de hacer buen uso de esta aplicación.



“Directorio del Simulador para Secuencia<T>”

A continuación se presenta la interface principal del simulador para “Secuencias”.



"Interface principal del Simulador para Secuencia<T>"

Descripción de las Funcionalidades del Simulador

El simulador para Secuencia<T> permite al Estudiante:

1. Crear una nueva Secuencia:

Para Crear una nueva Secuencia el estudiante deberá oprimir el Botón "**Crear**", ingresando previamente en la caja de texto el **Tamaño** (cantidad de posiciones) que desea tener en la Secuencia, debido a su característica de ser una Estructura Estática. *"El tamaño de la secuencia viene limitado a un valor mayor a cero (0) y menor a veinticinco (25) posiciones"*.



"Creación de una Secuencia de Tamaño igual a 19 posiciones"

2. Insertar Datos:

Para insertar datos el estudiante previamente debió haber creado una nueva Secuencia. Se debe ingresar el valor del dato a insertar en la Secuencia, el cual no puede ser menor a **-99** ni mayor a **999**. Una vez insertado el dato, este será mostrado a continuación en la Secuencia.



"Secuencia después de insertar los datos: 18, 1,-55, 66, 17,19 y 22"

3. Eliminar Datos:

Para Eliminar un Dato de la Secuencia el estudiante deberá ingresar a la caja de texto el valor del dato que desea eliminar, el cual debe cumplir las mismas características de los valores insertados. Una vez eliminado el dato de la Secuencia, este no será pintado.



"Secuencia después de eliminar el dato: 66".

4. Editar posición de la Secuencia:

Para editar una posición de la Secuencia el estudiante deberá insertar una **posición** valida (en la cual exista un dato) en donde se realizara la edición, y adicionalmente a esto deberá ingresar el **dato** con el cual desea reemplazar el valor encontrado en dicha posición. La Secuencia será pintada a continuación con la posición editada.



"Secuencia después de editar la posición '2' con el dato: '455'"

5. Conocer el Tamaño y la Capacidad de la Secuencia:

Para conocer el tamaño y la capacidad de la Secuencia, que son valores totalmente diferentes, el estudiante podrá oprimir el botón **"Dar Tamaño"** o **"Dar Capacidad"** dependiendo del valor de la Secuencia que desee conocer en su momento.



"Tamaño de la Secuencia determinado"



"Capacidad de la Secuencia determinada"

6. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Secuencia, existe la posibilidad de que el estudiante pueda **"adicionar nuevas funcionalidades"** a la aplicación, de acuerdo a las actividades asignadas por los docentes o el interés propio de generar nuevos algoritmos en cada estructura y poder simularlos gracias a la herramienta grafica del Simulador.

A continuación se presentan los pasos que deberá seguir el estudiante para crear una nueva funcionalidad dentro del Simulador de Secuencia:

- 6.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **Secuencia**, presente en el paquete **SEED_Colecciones** y que desea adicionar a la funcionalidad del Simulador. (Para el ejemplo, se creará un algoritmo que permita ordenar los datos dentro de la Secuencia por el Algoritmo de Ordenamiento **"Burbuja"**).

```

/**
 * Metodo que permite ordenar los datos de una Secuencia utilizando el
 * Algoritmo de Ordenamiento Burbuja.
 */
public void ordenarPorBurbuja(){
    for(int i=0; i<this.vector.length; i++){
        for(int j=i+1; j<this.vector.length; j++){
            int compara = ((Comparable)vector[i]).compareTo(vector[j]);
            if(compara>0){
                T aux = this.vector[i];
                this.vector[i] = this.vector[j];
                this.vector[j] = aux;
            }
        }
    }
}

```

6.2. A continuación el Estudiante debe generar un Método en la clase **SimuladorSecuencia** del paquete **Mundo_Secuencia**, que realice el **llamado** al Método con el nuevo algoritmo creado en la Estructura de Datos **Secuencia**. Para el llamado debe utilizar el objeto creado en el Mundo **miSecuencia**.

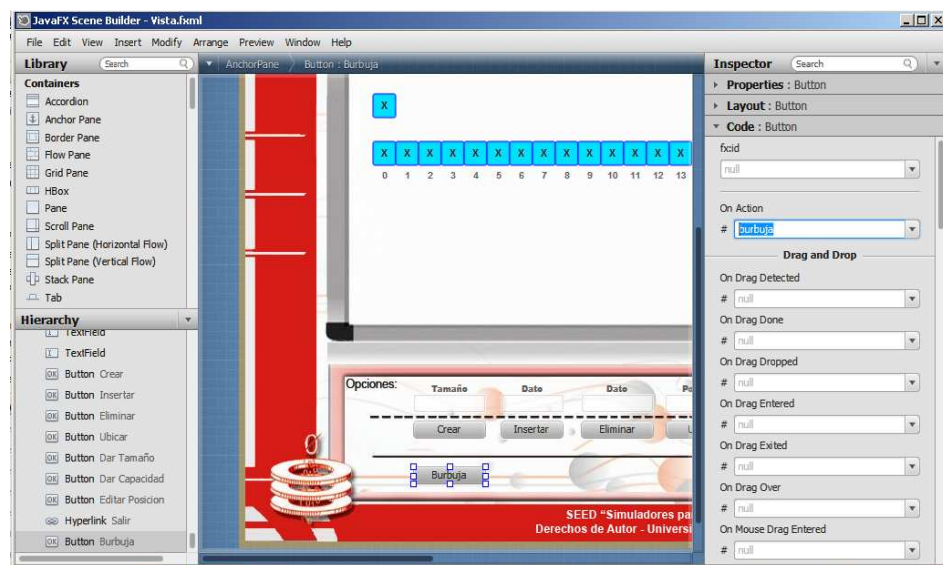
```

public void ordenarPorBurbuja() {
    this.miSecuencia.ordenarPorBurbuja();
}

```

6.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **Simulador_Secuencia**. Existe dos posibilidades para ello: Utilizar la herramienta “**JavaFX SceneBuilder**” para insertarlo, o agregar el código del Button en el Archivo **Vista.fxml**.

6.3.1. Utilizando JavaFX SceneBuilder



“Inserción de un nuevo Button con la herramienta JavaFX SceneBuilder”

6.3.2. Insertando directamente el elemento en Vista.fxml

```
<Button layoutX="184.0" layoutY="611.0" mnemonicParsing="false" onAction="#burbuja" prefWidth="80.0" text="Burbuja" />
```

“Inserción de un nuevo Button por medio de código FXML en el archivo Vista.fxml”

Es importante resaltar, para ambos casos, que se debe asignar el evento **“OnAction”** del Button, para el ejemplo **“burbuja”**, el cual será el nombre del **Método** dentro de la clase **Controlador** que permite realizar la nueva funcionalidad del Simulador.

A continuación el Método dentro del paquete **simsecuencia** en la clase **Simulador** que permite realizar el llamado a la nueva funcionalidad.

```
@FXML
private void burbuja(ActionEvent event) {
    this.simulador.ordenarPorBurbuja();
    this.pintarTDA();
    this.impNota("Secuencia ordenada por burbuja", 0);
}
```

Además del llamado al Método creado en **SimuladorSecuencia**, el estudiante deberá invocar el método que le permita volver pintar la Secuencia, el cual siempre será **“pintarTDA()”**. Opcionalmente se recomienda enviar un mensaje con la respuesta a la operación realizada utilizando **“impNota(“Mensaje a enviar “ , tipo)”** donde tipo es cero (0) si en un mensaje Exitoso y uno (1) en caso de ser un mensaje erróneo.

A continuación se comprueba el funcionamiento del Algoritmo realizado:



“Secuencia con 25 datos insertados de manera desordenada”



"Secuencia después de aplicar **burbuja()** a los 25 datos insertados previamente"