



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador ColaPrioridad<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas

2014



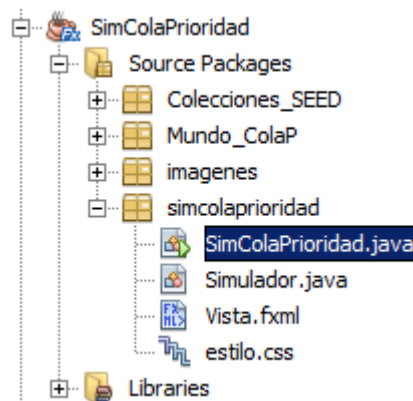
MANUAL DE USUARIO: Simulador “ColaP<T>”

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 ColaP<T>.

Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Cola de Prioridad: enColar (insertar) y deColar(Eliminar) datos en la estructura.

Adicionalmente el estudiante podrá conocer de la Cola de Prioridad, el tamaño y los datos almacenados dentro de la estructura. 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 ColaP<T>”

A continuación se presenta la interface principal del simulador para “Cola de Prioridad”. El simulador para ColaP<T> permite al Estudiante crear un Cola de Prioridad con solo correr la aplicación SimCola de Prioridad, aunque esta estructura es dinámica “*El tamaño de la Cola de Prioridad es limitado por cuestiones de simulación a un valor mayor a cero (0) y menor a quince (15) posiciones*”.



“Interface principal del Simulador para ColaP<T>”

Descripción de las Funcionalidades del Simulador

1. enColar Datos:

Se debe ingresar el valor del dato a insertar en la Cola de Prioridad, el cual no puede ser menor a **-99** ni mayor a **999**, rango seleccionado por cuestiones de que no se desborde el número del nodo gráfico, adicional al dato ingresado en la caja de Texto Dato a Encolar, se debe ingresar en la caja de texto Prioridad la prioridad del dato que se desea insertar. Una vez insertado el dato, este será mostrado a continuación en la Cola de Prioridad.



"Cola de Prioridad después de insertar los datos: 23, 8, 34, 89, 45 y 11".

2. DeColar Datos:

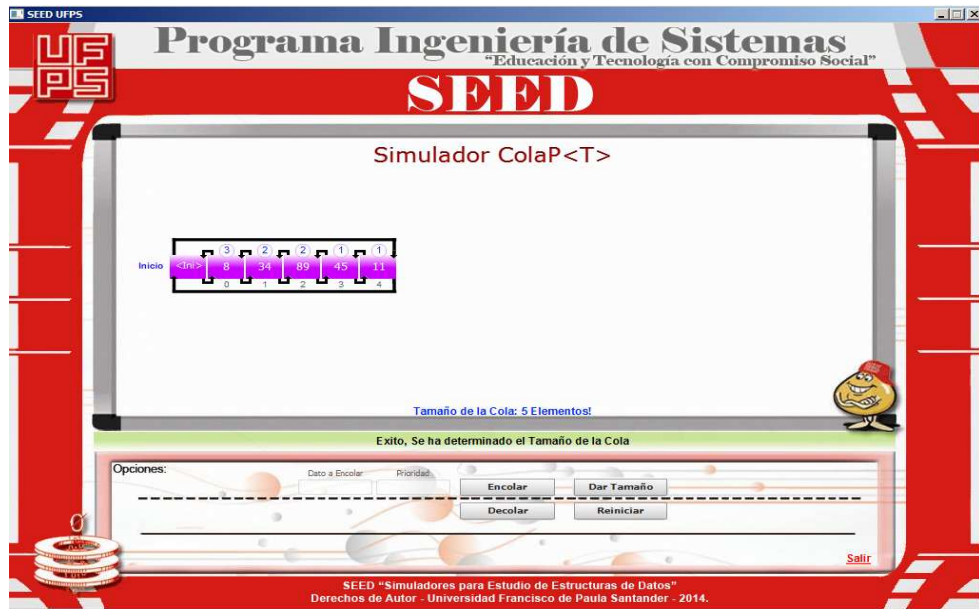
Para Decolar (Eliminar) un Dato de la Cola de Prioridad el estudiante deberá dar clic en el botón "Decolar", en donde la estructura debe poseer datos. Una vez eliminado el dato de la Cola de Prioridad, este no será pintado.



"Cola de Prioridad después de decolar el dato".

3. Conocer el Tamaño de la Cola de Prioridad:

Para conocer el tamaño de la Cola de Prioridad, el estudiante podrá dar clic el botón **“Dar Tamaño”** dependiendo del valor de la Cola de Prioridad que desee conocer en su momento.



“Tamaño de la Cola de Prioridad determinado”.

4. Reiniciar La Cola de Prioridad:

Para reiniciar la Cola de Prioridad, el estudiante podrá dar clic el botón **“Reiniciar”**, esta acción elimina todos los datos de la Cola de Prioridad dejándola vacía para que el estudiante comience a ilustrar de nuevo las funciones básicas.



“Reinicio de la Cola de Prioridad”.

5. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Cola Prioridad, 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 gráfica del Simulador.

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

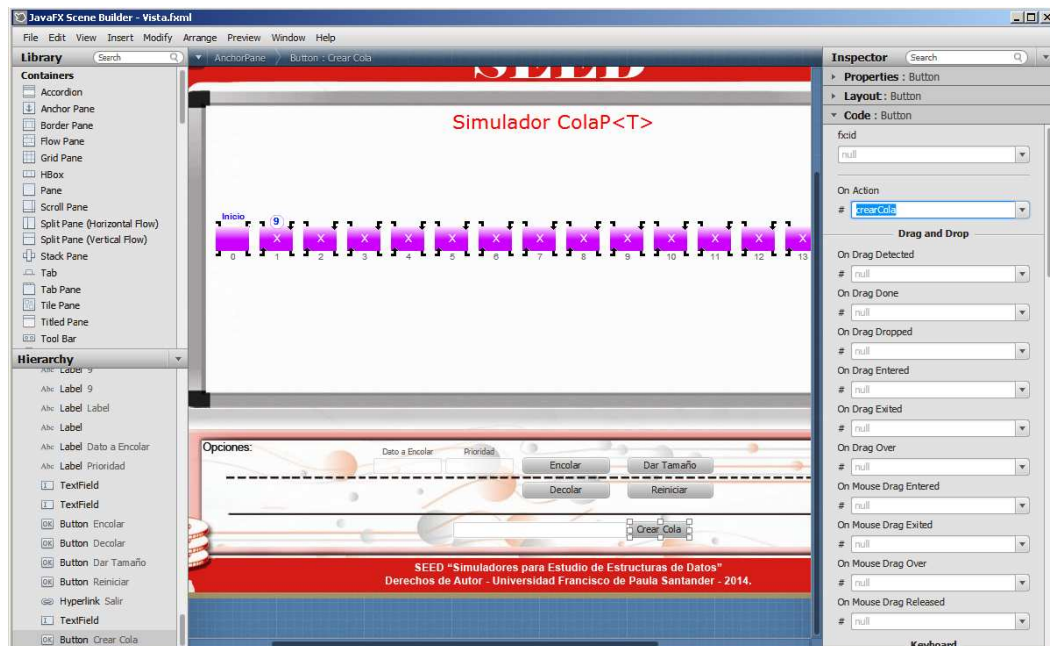
5.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **ColaP**, presente en el paquete **SEED_Colecciones** y que desea adicionar a la funcionalidad del Simulador. (Para el ejemplo, se creará un algoritmo que permita **crear una nueva Cola de prioridad** a partir de una cadena de datos separada por “-“ unos de otros. Se utilizará el método **encolar** ya existente):

5.2. A continuación el Estudiante debe generar un Método en la clase **SimuladorColaP** del paquete **Mundo_ColaP**, que realice el llamado al Método con el nuevo algoritmo creado en la Estructura de Datos **ColaP**. Para el llamado debe utilizar el objeto creado en el Mundo **miCola**.

```
public void crearCola(String pila){
    String v[] = pila.split("-");
    for(int i=0; i<v.length; i++){
        String dato[] = v[i].split(",");
        this.miCola.encolar(Integer.parseInt(dato[0]), (Integer.parseInt(dato[1])));
    }
}
```

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

5.3.1. Utilizando JavaFX SceneBuilder



5.3.2. Insertando directamente el elemento en Vista.fxml

```
<TextField fx:id="txtCrear" layoutX="306.0" layoutY="608.0" prefWidth="200.0" />
<Button layoutX="515.0" layoutY="608.0" mnemonicParsing="false" onAction="#crearCola" text="Crear Cola" />
```

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

A su vez, debe ser declarado el objeto **TextField** utilizado para capturar los datos con los que se desea crear la Pila.

```
@FXML private TextField txtCrear;
```

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

```
@FXML
private void crearCola(){
    String pila = this.txtCrear.getText();
    this.simulador.crearCola(pila);
    this.pintarTDA();
    this.impNota("La cola ha sido creada!", 0);
}
```

Además del llamado al Método creado en **SimuladorColaP**, el estudiante deberá invocar el método que le permita volver pintar la Cola, 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:

Primeramente se ingresan a la caja de Texto la cadena con los datos que se desea crear la nueva Cola:



“ingresando los datos de la nueva cola”

Al ejecutar la nueva función el Simulador debe crear una Cola con los datos ingresados en la cadena:



“ingresando los datos de la nueva cola”