



# SEED

“Simuladores para Estudio de Estructuras de Datos”

## Manual de Usuario

*Simulador Pila<T>*

*Versión: 1.0*

Universidad Francisco de Paula Santander  
Programa Ingeniería de Sistemas

2014



## MANUAL DE USUARIO: Simulador “Pila<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 Pila<T>.

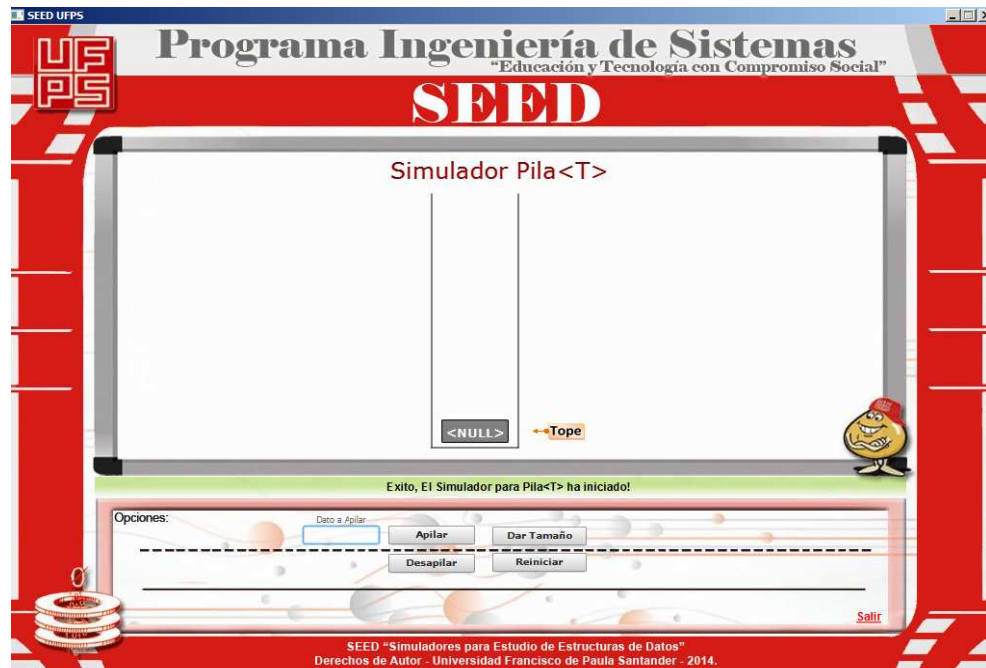
Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Pila: Apilar (insertar) y Desapilar (eliminar) datos en la estructura.

Adicionalmente el estudiante podrá conocer de la Pila, 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 Pila<T>”

A continuación se presenta la interface principal del simulador para “Pila”. El simulador para Pila<T> permite al Estudiante crear un Pila con solo correr la aplicación SimPila, aunque esta estructura es dinámica “*El tamaño de la Pila es limitado por cuestiones de simulación a un valor mayor a cero (0) y menor a once (11) posiciones*”.

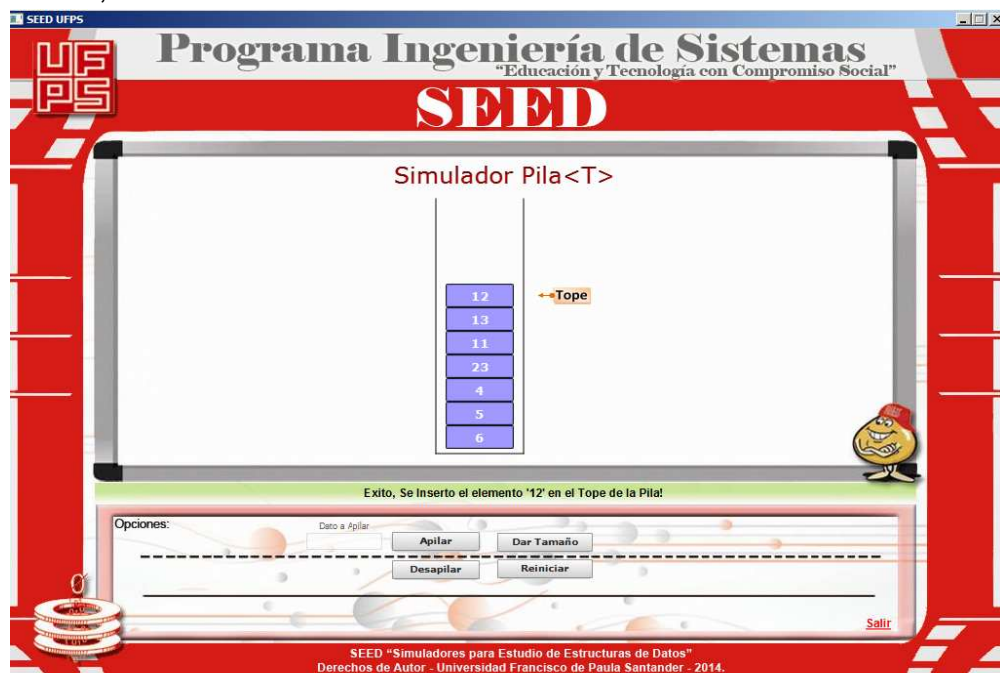


*“Interface principal del Simulador para Pila<T>”*

## Descripción de las Funcionalidades del Simulador

### 1. Apilar Datos:

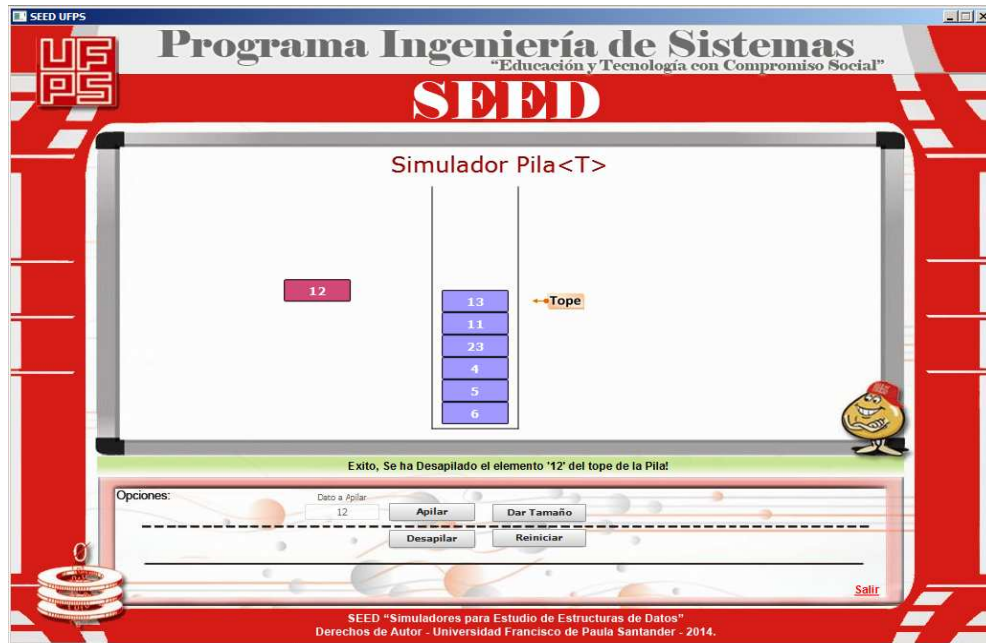
Se debe ingresar el valor del dato a insertar en la Pila, el cual no puede ser menor a **-99** ni mayor a **999**, rango seleccionado por cuestiones de que no se desborde el número del nodo gráfico. Una vez insertado el dato, este será mostrado a continuación en la Pila.



*“Pila después de insertar los datos: 6, 5, 4, 23, 11, 13 y 12”.*

## 2. Desapilar Datos:

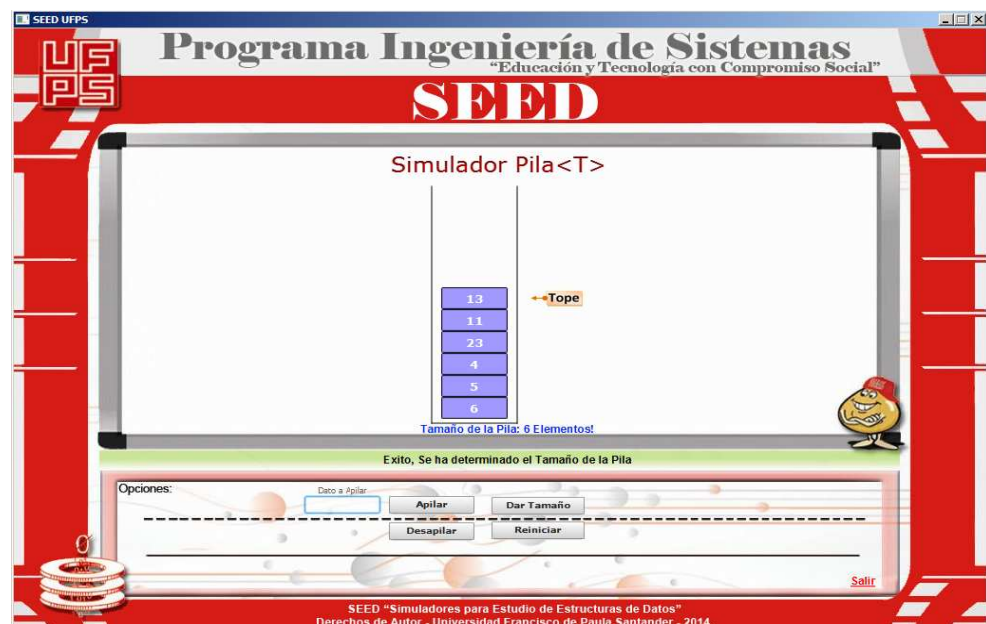
Para Desapilar (Eliminar) un Dato de la Pila el estudiante deberá dar clic en el botón “Desapilar”, en donde la estructura debe poseer datos. Una vez eliminado el dato de la Pila, este no será pintado.



*“Pila después de eliminar el dato: 12”.*

## 3. Conocer el Tamaño de la Pila:

Para conocer el tamaño de la Pila, el estudiante podrá oprimir el botón “Dar Tamaño” dependiendo del valor de la Pila que desee conocer en su momento.



*“Tamaño de la Pila determinado”.*

#### 4. Reiniciar La Pila:

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



*“Reinicio de la Pila”.*

#### 5. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Pila, 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 Pila:

**5.1.** El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **Pila**, 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 Pila** a partir de una cadena de datos separada por “-“unos de otros. Se utilizará el método **apilar** ya existente):

```

/**
 * Metodo que inserta un elemento en la Pila. <br>
 * <b>post: </b> Se inserto un elemento dentro de la Pila.<br>
 * @param info es de tipo T y contiene la información a insertar en la pila.
 */
public void apilar(T info){
    if(this.esVacia()){
        this.tope = new Nodo<T>(info, null);
    }
    else
        this.tope=new Nodo<T>(info, this.tope);
    this.tamanio++;
}

```

- 5.2. A continuación el Estudiante debe generar un Método en la clase **SimuladorPila** del paquete **Mundo\_Pila**, que realice el **llamado** al Método con el nuevo algoritmo creado en la Estructura de Datos **Pila**. Para el llamado debe utilizar el objeto creado en el Mundo **miPila**.

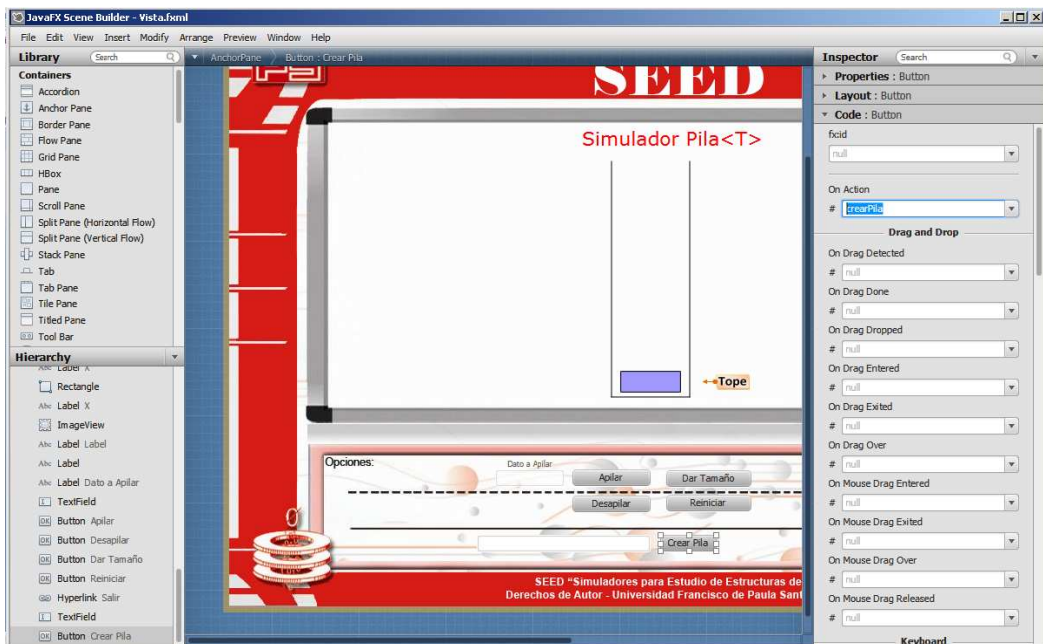
```

public void crearPila(String pila){
    String v[] = pila.split("-");
    for(int i=0; i<v.length; i++){
        this.miPila.apilar(Integer.parseInt(v[i]));
    }
}

```

- 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 **SimuladorPila**. 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 id="txtCrearPila" fx:id="txtCrear" layoutX="285.0" layoutY="608.0" prefWidth="200.0" />
<Button layoutX="496.0" layoutY="608.0" mnemonicParsing="false" onAction="#crearPila" text="Crear Pila" />
```

Es importante resaltar, para ambos casos, que se debe asignar el evento **“OnAction”** del Button, para el ejemplo **“crearPila”**, 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 **simpila** en la clase **Simulador** que permite realizar el llamado a la nueva funcionalidad.

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

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



*“ingresando los datos de la nueva pila”*

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



*“Lista creada a partir de una lista de datos”*