

UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORÍA ACADÉMICA ESCUELA DE
CIENCIAS EXACTAS Y NATURALES CARRERA
INGENIERÍA INFORMÁTICA



CÁTEDRA INGENIERÍA DE SOFTWARE

ASIGNATURA

00825 ESTRUCTURA DE DATOS

PROYECTO 2

VALOR: 20% (2.0)

I CUATRIMESTRE 2025

Estructura de Datos- Proyecto 2

I Objetivo de aprendizaje

- Analizar el funcionamiento de las pilas y colas y explicar su implementación.

II Temas de estudio

- Pilas y colas.

III Descripción del trabajo

Precondiciones:

1. Como herramienta de desarrollo se deberá utilizar el Netbeans en su versión más reciente. Se debe descargar el último release de la página [Apache Netbeans](#).
2. También se debe utilizar el Java JDK más reciente. Pueden descargarlo de la página [Oracle](#).
3. El programa debe realizarse en modo gráfico (GUI), es decir, no se permite en modo consola.
4. No se permite el uso de cuadros de diálogo tipo MessageBox para solicitar o para mostrar datos. Para ello se pueden utilizar cajas de texto, etiquetas o listas gráficas según sea. La única excepción es para mostrar excepciones de la aplicación o para dar un mensaje al usuario por operaciones incorrectas en el sistema resultado de validaciones. Si la operación se realiza exitosamente, no se deben mostrar mensajes utilizando estos cuadros de diálogo.
5. Los datos deben persistir en memoria en todo momento hasta que se cierre la aplicación.

IV Instrucciones:

Se deberá implementar un programa para gestionar Problemas e Incidentes para una mesa de ayuda.

El programa contendrá las siguientes pantallas o funcionalidades:

Registro de problemas

Pantalla para incluir los problemas detectados.

Cada problema incluido se almacenará en un arreglo de objetos de la clase "Problema", la cual funcionará como una cola (primer elemento en entrar, primero en salir). Un atributo del elemento "Problema" será un arreglo de objetos de la clase "Incidente", el cual se describe en la siguiente funcionalidad (Registro de incidentes).

No se podrán usar colecciones o interfaces de Java como Queue, Stack, ArrayList o ninguna otra colección ya implementada, sino que se deben utilizar **arreglos de objetos**.

En esta pantalla se podrá incluir registros de problemas con los siguientes campos:

Nombre del campo	Tipo de dato
ID del problema	Número entero. Debe ser un consecutivo único generado por el sistema.
Descripción breve	Texto.

Se debe mostrar una lista con los problemas registrados, la cual se actualiza con cada inserción.

La lista debe ser un control tipo tabla (JTable u otro similar).

Registro de incidentes:

En esta pantalla se debe mostrar una lista gráfica con los diferentes problemas registrados y sus diferentes campos. Esta lista debe ser un control tipo tabla (JTable u otro similar).

El usuario deberá seleccionar uno de los problemas y a continuación ingresar los datos del incidente, el cual se va a relacionar con el problema previamente seleccionado.

Los campos por ingresar para un incidente serán los siguientes:

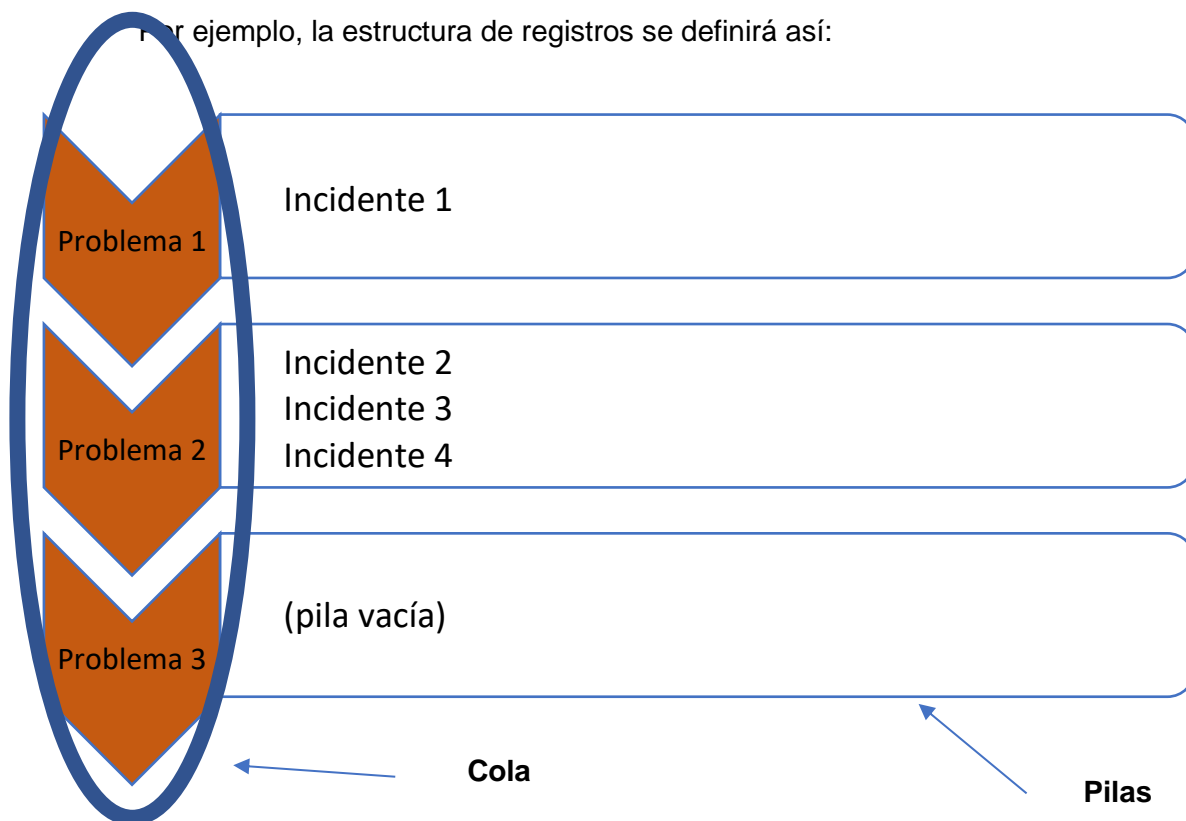
Nombre del campo	Tipo de dato
ID del incidente	Número entero. Debe ser un consecutivo generado por el sistema. Un solo consecutivo para todos los incidentes de los diferentes problemas.
Usuario solicitante	Texto.
Categoría	Se selecciona un valor de una lista (control tipo DropDownList) con las siguientes opciones: <ul style="list-style-type: none">• Software• Grupos y cuentas de usuario• Herramientas colaborativas• Seguridad Informática• Redes• Monitoreo
Detalle	Texto con la descripción del incidente.
Fecha	Campo para seleccionar una fecha. La librería utilizada para este campo debe ser incluida en el proyecto al entregarlo.

El almacenamiento de los incidentes por problema se realizará mediante un arreglo de objetos de la clase "Incidente", el cual funcionará como una pila (primero en entrar, último en salir).

No podrá utilizarse ninguna colección de Java como Queue, Stack, ArrayList ni ninguna otra colección. Debe trabajarse con **arreglos de objetos**.

Se deberá utilizar una pila **independiente** con los diferentes registros de incidentes, dentro de cada elemento de la cola principal de problemas.

Por ejemplo, la estructura de registros se definirá así:



Al inicio la Cola de Problemas estará vacía. Al incluir un primer problema, éste contendrá una pila de incidentes vacía, hasta que el usuario incluya los incidentes de ese problema.

Eliminación de incidentes

En esta pantalla el usuario deberá seleccionar uno de los problemas registrados y a continuación se mostrarán los incidentes registrados pertenecientes al problema seleccionado (se deben mostrar todos los campos de los incidentes).

Al hacer clic en un botón “Eliminar incidente”, se eliminará el incidente que se registró de último en el problema seleccionado (recordemos que se trata de una pila y por este motivo no se debe seleccionar un elemento a eliminar). Posteriormente se refrescará la lista de incidentes en la pantalla.

Traslado de incidentes

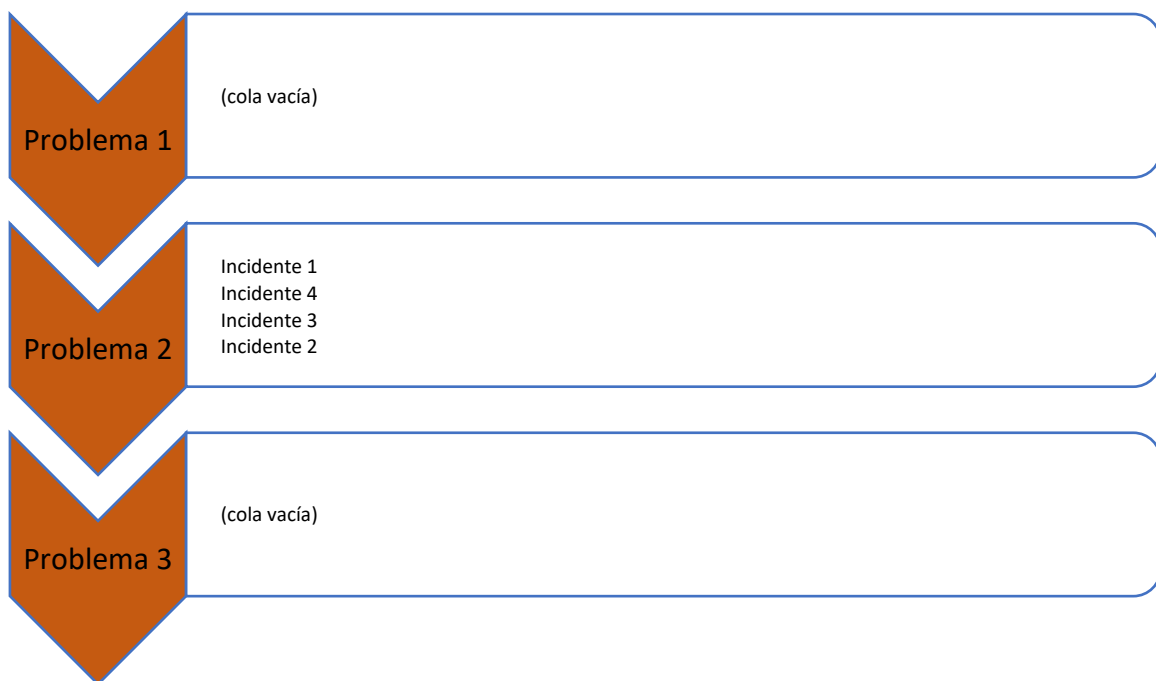
En esta pantalla el usuario debe poder trasladar una pila de incidentes de un problema a otro.

Para ello deberá seleccionar el problema origen y el problema destino.

Adicionalmente debe cumplir con las siguientes consideraciones:

1. La cola de problemas debe tener al menos 2 elementos.
2. El problema origen no debe estar vacío.
3. El problema origen y el destino no pueden ser el mismo.
4. Cada incidente por trasladar se saca de la pila origen (POP) y se incluye en la pila destino (PUSH).

Por ejemplo, basándonos en la estructura de registros del ejemplo anterior, si el usuario traslada los incidentes del “Problema 1” al “Problema 2”, el resultado final sería el siguiente:



Eliminación de problemas

La eliminación de problemas consiste en una acción que intentará eliminar de la cola principal el primer elemento registrado (como en cualquier cola, por lo tanto, no se debe seleccionar el problema a eliminar).

La eliminación se hará efectiva únicamente si el primer elemento de la cola no contiene ningún incidente asociado (es decir, su cola de incidentes está vacía). En caso de existir al menos un incidente, se mostrará un mensaje indicando que no se puede eliminar el problema debido a que tiene incidentes relacionados.

Si el problema es eliminado, se refrescará la lista de problema con los elementos existentes.

Deberá existir un menú con cada una de las opciones.

V Rúbrica o escala de evaluación

NO.	INDICADORES POR EVALUAR	CUMPLIMIENTO		PUNTOS
		Cumple	No cumple	
REGISTRO DE PROBLEMAS				
1.	<p>La pantalla del registro de problemas incluye los 2 campos indicados en el enunciado con sus correspondientes tipos de dato según lo especificado para cada campo.</p> <p>Cada problema incluido se almacena en un arreglo de objetos de la clase “Problema”, el cual funciona como una cola (primer elemento en entrar, primero en salir).</p> <p>No se utilizan colecciones o interfaces de Java como Queue, Stack, ArrayList o ninguna otra colección ya implementada.</p> <p>Un atributo de la clase “Problema” es un arreglo de objetos de la clase “Incidente”.</p> <p>Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.</p>			2
2.	<p>Se muestra una lista con los problemas registrados, la cual se actualiza con cada inserción.</p> <p>La lista debe ser un control de tipo tabla (JTable o similar).</p>			1
REGISTRO DE INCIDENTES				
3.	<p>En esta pantalla se debe mostrar una lista gráfica con los diferentes problemas registrados. La lista debe ser un control de tipo tabla (JTable o similar).</p> <p>El usuario puede seleccionar una de las productoras registradas.</p>			1
4.	<p>El usuario puede seleccionar uno de los problemas y a continuación ingresar los datos del incidente, el cual se va a relacionar con el problema previamente seleccionado.</p> <p>Los datos de ingreso incluyen los 5 campos indicados en el enunciado correspondiente a este apartado, considerando los tipos de dato señalados.</p> <p>El almacenamiento de los incidentes por problema (incluyendo los 5 campos especificados en el enunciado) se realizará mediante un arreglo de objetos de la clase “Incidente”, el cual funcionará como una pila (último en entrar, primero en salir).</p> <p>No podrá utilizarse ninguna colección de Java como Queue, Stack, ArrayList ni ninguna otra colección.</p> <p>Tampoco se podrá implementar usando listas enlazadas con nodos.</p> <p>Se deberá utilizar una pila independiente con los diferentes registros de incidentes, dentro de cada elemento de la cola principal de problemas.</p> <p>Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.</p>			5

ELIMINACIÓN DE INCIDENTES				
5.	<p>En esta pantalla se muestra una lista gráfica con los diferentes incidentes registrados.</p> <p>El usuario puede seleccionar un problema y se refresca la lista de incidentes que tiene relacionados (se deben mostrar todos los campos de los incidentes).</p> <p>Al hacer clic en un botón “Eliminar incidente”, se elimina el incidente que se registró de último (usando una estructura de pila, por lo tanto, el usuario no selecciona el incidente a eliminar). Luego, se refresca la lista de incidentes en la pantalla.</p>			5
TRASLADO DE INCIDENTES				
6.	<p>El usuario puede seleccionar el problema origen y el destino.</p> <p>Se valida que la cola de problemas tenga al menos 2 elementos.</p> <p>Se valida que el problema origen no esté vacío.</p> <p>Cada incidente por trasladar se saca de la pila origen (POP) y se incluye en la pila destino (PUSH).</p>			4
ELIMINACIÓN DE PROBLEMAS				
7.	<p>La eliminación se realiza únicamente si el primer elemento de la cola no contiene ningún incidente asociado (es decir, su pila de incidentes está vacía). Luego de eliminado el problema, se refresca la lista de problemas.</p> <p>En caso de existir al menos un incidente, se muestra un mensaje indicando que no se puede eliminar problema debido a que tiene incidentes relacionados.</p>			2
GENERALIDADES				
8.	Utiliza modo gráfico para todas las funcionalidades sin excepción.			9
9.	No utiliza cuadros de diálogo (tipo MessageBox u otro control similar) para dar ni para solicitar información al usuario, a menos que sean excepciones del programa o que el usuario omitió algo necesario para la operación que quiere realizar.			1