

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 1

VALOR: 30% (3.0)

II CUATRIMESTRE 2024

# Estructura de Datos- Proyecto 1

## I Objetivo de aprendizaje

- Adquirir los conocimientos básicos sobre las estructuras de datos.
- Analizar la forma en que se implementa la recursión y comprender cuándo se debe utilizar.
- Comprender el funcionamiento de algoritmos de ordenamiento.

## II Temas de estudio

- La API de colecciones y recursividad.
- Algoritmos de ordenamiento.

## 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. El programa debe realizarse en modo gráfico (GUI), es decir, no se permite en modo consola.
3. 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.
4. Los datos deben persistir en memoria en todo momento hasta que se cierre la aplicación.

## IV Instrucciones:

Se deberá implementar un programa para llevar un registro de pedidos de camisetas estampadas para la empresa "XDesign".

El programa deberá contener las siguientes funcionalidades:

### Registro de pedidos:

En esta pantalla se podrá incluir los datos generales de cada pedido de camisetas, con los siguientes campos:

Nombre del campo	Tipo de dato
<b>ID del pedido</b>	Identificador consecutivo numérico generado por el programa.
<b>Tipo de camiseta</b>	Lista desplegable (tipo DropDownList) con las opciones: <ul style="list-style-type: none"><li>• Lisa manga larga</li><li>• Polo manga corta</li><li>• Estampado manga corta</li><li>• Lisa manga corta</li><li>• Deportiva manga larga</li><li>• Deportiva manga corta</li><li>• Deportiva sin mangas</li></ul>
<b>Talla</b>	Lista desplegable (tipo DropDownList) con las opciones: <ul style="list-style-type: none"><li>• S</li><li>• M</li><li>• L</li><li>• XL</li><li>• 2XL</li></ul>
<b>Cantidad</b>	Campo de texto que solo acepte números de 1 o 2 dígitos.
<b>Código de diseño</b>	Código alfanumérico (no se permiten caracteres especiales ni espacios en blanco).
<b>Tipo de pedido</b>	Lista tipo DropDownList con las siguientes opciones: <ul style="list-style-type: none"><li>• Recoger en tienda</li><li>• Envío a domicilio</li></ul>
<b>Dirección de entrega</b>	Campo alfanumérico que se habilita únicamente si el usuario selecciona el tipo de pedido "Envío a domicilio".
<b>Forma de pago</b>	Lista desplegable (tipo DropDownList) con las opciones: <ul style="list-style-type: none"><li>• Efectivo</li><li>• Sinpe móvil</li><li>• Transferencia bancaria</li><li>• Tarjeta de débito/crédito</li></ul>
<b>Número telefónico del cliente</b>	Campo numérico de 8 dígitos sin guiones.
<b>Nombre completo del cliente</b>	Campo alfanumérico.

La información de los pedidos con los campos anteriormente especificados, se guardarán en una clase “Pedido” y cada una de estas se agregará a una única colección utilizando la clase **ArrayList** de Java.

#### *Listado de pedidos:*

En esta pantalla se listan todos los pedidos de camisetas registrados en la colección (con todos sus campos).

En esta opción se debe utilizar una clase de tipo **Iterator** para recorrer la colección. Los datos deben ser mostrados en un componente tipo tabla (un JTable o similar).

#### *Búsqueda y edición de un pedido:*

Para buscar un pedido determinado, el usuario debe digitar el número de teléfono del cliente.

El programa muestra los registros encontrados que correspondan a pedidos con el número de teléfono indicado, desplegándolos en un componente tipo tabla (un JTable o similar).

El usuario seleccionará un único registro para su edición. Con ello se mostrará en pantalla los campos del registro a editar con los valores actuales para poder ser editados.

El campo ID del pedido no podrá ser editado.

Habrà un botón GUARDAR para guardar los cambios.

#### *Comparación de valores:*

Debe utilizar **recursividad** para esta operación.

El usuario deberá ingresar el ID de un pedido registrado en el sistema.

El programa obtendrá el registro correspondiente al ID indicado y tomará el valor del campo “Cantidad”.

Utilizando una función recursiva, se convertirá el valor decimal del campo “Cantidad” a su respectivo valor con base binaria.

El programa mostrará en pantalla el valor decimal de “Cantidad” y su respectivo valor binario de la siguiente manera:

<b>ID del pedido:</b>	4
<b>Cantidad:</b>	79
<b>Valor binario:</b>	1001111

### *Ordenamiento de la lista de pedidos:*

En esta pantalla se mostrarán los pedidos de camisetas ordenados según las siguientes opciones y según lo requiera el usuario:

1. Orden ascendente (de menor a mayor) por el campo “Número telefónico del cliente” utilizando el algoritmo “Por inserción”.
2. Orden descendente por el campo “Código de diseño” (de mayor a menor, donde Z es mayor que A, 9 es mayor que 0 y las letras son mayores que los números) utilizando el algoritmo “Merge sort”.

Previo al ordenamiento, se deben copiar los datos de los elementos existentes en la colección ArrayList a un arreglo primitivo de cadenas de texto, para uso del algoritmo de ordenamiento correspondiente.

Si utiliza el método Sort de la colección ArrayList o cualquier otra, obtendrá un cero en esta funcionalidad automáticamente.

Finalmente se mostrará la lista ordenada de pedidos de camisetas en un componente tipo tabla (JTable u otro similar), según el criterio de ordenamiento indicado.

Debe existir un menú con cada una de las opciones.

Los resultados de las opciones se muestran en la pantalla correspondiente, sin usar mensajes de diálogo para ello.

## V Rúbrica o escala de evaluación

NO.	INDICADORES POR EVALUAR	CUMPLIMIENTO		PUNTOS
		Cumple	No cumple	
REGISTRO DE PEDIDOS				
1.	La pantalla del registro de pedidos incluye los 10 campos indicados en el enunciado con sus correspondientes tipos de dato según lo especificado para cada campo. 1 punto por cada uno.			10
2.	Los 10 campos indicados en el enunciado se almacenan en clases llamadas Pedido y cada una de estas se agregará a una única colección utilizando la clase ArrayList de Java. 1 punto por cada uno.			10
LISTADO DE PEDIDOS				
3.	En esta pantalla se listan todos los pedidos de camisetas registrados en la colección de tipo ArrayList (con todos sus 10 campos según el enunciado en el Registro de pedidos). Los datos se despliegan en un componente tipo tabla (JTable o similar). Se utiliza una clase de tipo <b>Iterator</b> para recorrer la colección ArrayList. Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			10
BÚSQUEDA Y EDICIÓN DE UN PEDIDO				
4.	Para buscar un evento, el usuario digita el Número telefónico del cliente. El programa muestra los registros encontrados para el número indicado, desplegándolos en un componente tipo table (JTable o similar) e incluyendo los 10 campos del pedido según el enunciado en el Registro de Pedidos.			10
5.	Cuando el usuario selecciona un registro para su edición, se muestran en pantalla los 10 campos del registro a editar con los valores actuales y el campo ID del evento no podrá ser editado. 1 punto por cada uno.			10
6.	Al hacer clic en el botón “Guardar”, los 10 campos del registro quedan actualizados.			10
COMPARACIÓN DE VALORES				

7.	El programa obtiene el registro con el ID de pedido indicado por el usuario. Usando recursividad, convierte el valor decimal del campo "Cantidad" a su base binaria. Se muestra en pantalla los valores de Cantidad en su base decimal y binaria.			10
<b>ORDENAMIENTO DE LA LISTA DE EVENTOS</b>				
8.	La pantalla hace ordenamiento ascendente (de menor a mayor) de los registros usando el valor del campo "Número telefónico del cliente" utilizando el algoritmo "Por inserción" (sin usar el método "sort" de la colección ArrayList o alguna otra) y muestra el resultado en un componente tipo tabla (JTable o similar).  Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			10
9.	La pantalla hace ordenamiento descendente por el campo "Código de diseño" (de mayor a menor, donde Z es mayor que A, 9 es mayor que 0 y las letras son mayores que los números) utilizando el algoritmo "Merge sort" (sin usar el método "sort" de la colección ArrayList o alguna otra) y muestra el resultado en un componente tipo tabla (JTable o similar).  Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			10
<b>GENERALIDADES</b>				
10.	Se utiliza un menú con cada una de las opciones.			5
11.	Se utiliza el modo gráfico para el registro de pedidos			5
12.	Se utiliza el modo gráfico para el listado de pedidos.			5
13.	Se utiliza el modo gráfico para la búsqueda y edición de un pedido.			5
14.	Se utiliza el modo gráfico para la comparación de valores.			5
15.	Se utiliza el modo gráfico para el ordenamiento de la lista de pedidos.			5
16.	No utiliza cuadros de diálogo (MessageBox o similares) 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.			5