

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)

I CUATRIMESTRE 2025

# 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. 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 llevar un registro de los eventos que se realizarán en los próximos Juegos Olímpicos Los Ángeles 2028.

El programa deberá contener las siguientes funcionalidades:

### Registro de eventos:

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

Nombre del campo	Tipo de dato
<b>ID del evento</b>	Identificador consecutivo numérico generado por el programa.
<b>Fecha</b>	Campo para seleccionar una fecha. La librería utilizada para este campo debe ser incluida en el proyecto al entregarlo.
<b>Hora</b>	Campo para digitar la hora y minutos del evento. El formato deberá ser la hora militar. Por ejemplo: "13:15".
<b>Deporte</b>	Texto con el nombre del deporte al que corresponde el evento.
<b>Género</b>	Lista desplegable (tipo DropDownList) con las opciones: <ul style="list-style-type: none"><li>• Femenino</li><li>• Masculino</li></ul>
<b>Individual</b>	Campo tipo CheckBox. Por default estará marcado para cada registro.
<b>País A</b>	Texto con el nombre del país competidor A.
<b>País B</b>	Texto con el nombre del país competidor B.
<b>Atleta ganador o MVP</b>	Texto con el nombre del atleta ganador del evento, o en el caso de equipo, nombre del atleta más valioso.

La información de los eventos con los campos anteriormente especificados, se guardarán en una clase "Evento" y cada una de estas se agregará a una única colección utilizando la clase **LinkedList** de Java.

### Listado de eventos:

En esta pantalla se listan todos los eventos deportivos 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 usando un componente tipo tabla (un JTable o similar).

### Búsqueda y edición de un evento:

Para buscar un evento determinado, el usuario debe seleccionar una fecha utilizando un componente de fecha (como el JDatePicker o similar) y hacer clic en un botón Buscar.

El programa muestra los registros encontrados (con todos sus campos) que correspondan a eventos con la fecha igual a la seleccionada, desplegándolos en un componente tipo tabla (un JTable o similar).

De los registros desplegados, el usuario podrá seleccionar un único registro para su edición. Al seleccionar un registro, se mostrarán en pantalla los campos del registro a editar con los valores actuales para poder ser editados.

La edición no será dentro de la misma tabla que contiene los registros encontrados, sino que en un bloque aparte se mostrarán cada uno de los campos a editar similar a la pantalla de registro de eventos, considerando lo especificado en la columna de tipo de dato. Por ejemplo, en el caso del Género, se mostrará una lista desplegable igual a la mostrada en la pantalla de registro, mostrando como seleccionado el valor actual del registro.

El campo ID del evento no podrá ser editado.

Habrà un botón GUARDAR para guardar los cambios.

### *Búsqueda de palíndromos:*

Debe utilizar **recursividad** para esta operación.

En esta opción, el programa buscará para todos los registros de eventos, el texto del campo “Atleta ganador o MVP” que corresponda a un palíndromo.

Un palíndromo es una palabra o frase cuyas letras están dispuestas de manera que se lee igual de izquierda a derecha o viceversa.

Ejemplos de palíndromos: “A mi loca Colima”, “Amad a la dama”, “Ana”, “Siris”.

Los textos encontrados deben ser listados en pantalla.

### *Ordenamiento de la lista de eventos:*

En esta pantalla se mostrarán los eventos deportivos ordenados según los siguientes criterios según lo requiera el usuario:

1. Orden ascendente o descendente por el campo “Deporte” utilizando el algoritmo “Selección”.
2. Orden ascendente o descendente por el campo “Fecha” (considerando el formato año-mes-día) utilizando el algoritmo “Quick sort”.

No se permite utilizar el método Sort (o cualquier otro método para ordenar existente en la colección) del LinkedList o cualquier otra colección.

Los registros ordenados (con todos sus campos) se mostrarán en un componente tipo tabla (JTable u otro similar), según el criterio de ordenamiento indicado por el usuario.

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 EVENTOS				
1.	La pantalla del registro de eventos incluye los 9 campos indicados en el enunciado con sus correspondientes tipos de dato según lo especificado para cada campo. Los 9 campos se almacenan en clases llamadas Evento y cada una de estas se agrega a una única colección utilizando la clase LinkedList de Java. 1 punto por cada uno.			9
LISTADO DE EVENTOS				
2.	En esta pantalla se listan todos los eventos deportivos registrados en la colección de tipo LinkedList (con todos sus 9 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 LinkedList. Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			9
BÚSQUEDA Y EDICIÓN DE UN EVENTO				
3.	Para buscar eventos, el usuario selecciona la fecha utilizando un componente tipo Date como JDatePicker o similar. La librería del componente Fecha se incluye en el proyecto entregado. El programa muestra los registros encontrados correspondientes a la fecha indicada, desplegándolos en un componente tipo table (JTable o similar) e incluyendo los 9 campos del pedido según el enunciado en el Registro de Pedidos. Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			9
4.	Cuando el usuario selecciona un registro para su edición, se muestran en pantalla los 9 campos del registro a editar con los valores actuales. El campo ID del evento no podrá ser editado. La edición no se hace en el componente tipo tabla, sino que los campos se muestran en un bloque separado			9

	donde cada campo cumple con el tipo de dato según lo indicado en el registro de eventos. 1 punto por cada uno.			
5.	Al hacer clic en el botón “Guardar”, los 9 campos del registro quedan actualizados.			9
<b>BÚSQUEDA DE PALÍNDROMOS</b>				
6.	Utiliza recursividad para esta operación. El programa busca para todos los registros de eventos, el texto del campo “Atleta ganador o MVP” que corresponda a un palíndromo. Los textos encontrados se listan en pantalla.			9
<b>ORDENAMIENTO DE LA LISTA DE EVENTOS</b>				
7.	El programa hace ordenamiento ascendente o descendente por el campo “Deporte” utilizando el algoritmo “Selección” (sin usar el método “sort” de la colección LinkedList o alguna otra) y muestra los registros ordenados con todos sus campos en un componente tipo tabla (JTable o similar).  Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			9
8.	El programa hace ordenamiento ascendente o descendente por el campo “Fecha” (considerando el formato año-mes-día) utilizando el algoritmo “Quick sort” (sin usar el método “sort” de la colección LinkedList o alguna otra) y muestra los registros ordenados con todos sus campos en un componente tipo tabla (JTable o similar).  Nota: debe cumplir con todos los enunciados anteriores, sino no, la nota será 0.			9
<b>GENERALIDADES</b>				
9.	Utiliza modo gráfico para todas las funcionalidades sin excepción.			25
16.	No utiliza cuadros de diálogo (MessageBox o similares) para dar o solicitar información al usuario, ni para indicar el resultado de una operación exitosa. Solo se usan a menos que sean excepciones del programa o que el usuario omitió algo necesario para la operación que quiere realizar.			3