

## Estructuras de datos

### Ejercicio 1: Crear una lista de strings y mostrar su contenido

1. Crea una lista de strings llamada `listaStrings`.
2. Agrega tres elementos a la lista: "Hola", "chau", "otro texto".
3. Utiliza un bucle for para recorrer la lista y mostrar cada elemento por pantalla.

### Ejercicio 2: Buscar un elemento en una lista

1. Utiliza el método `contains` para verificar si la lista contiene el elemento "chau".
2. Imprime un mensaje indicando si el elemento fue encontrado o no.

### Ejercicio 3: Eliminar elementos de una lista

1. Elimina el elemento "chau" de la lista utilizando el método `remove`.
2. Muestra la lista actualizada por pantalla.

### Ejercicio 4: Iterar una lista con un Iterator

1. Crea un `Iterator` para la lista.
2. Utiliza un bucle `while` para recorrer la lista e imprimir cada elemento en orden inverso.

## Arrays

### Ejercicio 1: Crear un array de enteros y mostrar la suma de todos los elementos.

Crea un array de enteros del 1 al 5 y luego suma sus elementos.

### Ejercicio 2: Crear un array de cadenas e imprimir la longitud de cada cadena.

Crea una array de String con los valores "Hola", "Mundo", "Java" y "Programación" y luego recorre el contenido del array e imprime el largo de cada palabra.

### Ejercicio 3: Crear un array de enteros e imprimir los números pares.

Crea un array de enteros del 1 al 10 e imprime los números pares.

### Ejercicio 4: Crear un array bidimensional de enteros e imprimir su contenido.



Universidad Tecnológica

Crea un array de 2 dimensiones (matriz) de 3x3 de modo que tenga los valores: 1,2,3 - 4,5,6 y 7,8,9 e imprime sus valores.

## Algoritmos de búsqueda y ordenamiento

### **Ejercicio 1. Ordenamiento Burbuja (Bubble Sort):**

- Implementa el algoritmo de ordenamiento burbuja en Java para ordenar un array de enteros de menor a mayor.

### **Ejercicio 2. Ordenamiento por Selección (Selection Sort):**

- Implementa el algoritmo de ordenamiento por selección en Java para ordenar un array de enteros de mayor a menor.

### **Ejercicio 3. Búsqueda Binaria (Binary Search):**

- Implementa el algoritmo de búsqueda binaria en Java para encontrar la posición de un elemento en un array ordenado de enteros. Asegúrate de manejar el caso en el que el elemento no esté presente en el array.

### **Ejercicio 4. Ordenamiento por Inserción (Insertion Sort):**

- Implementa el algoritmo de ordenamiento por inserción en Java para ordenar un array de cadenas de caracteres en orden alfabético.

### **Ejercicio 5. Búsqueda Secuencial (Sequential Search):**

- Implementa el algoritmo de búsqueda secuencial en Java para buscar un elemento en un array de enteros no ordenado. Devuelve la posición del elemento si se encuentra, o -1 si no está presente.