

EJERCICIOS ARREGLOS

1. Realizar una aplicación en JAVA que solicite al usuario un numero entero N, cree un arreglo de String de tamaño N, llene el arreglo con datos pedidos al usuario y muestre el arreglo en pantalla.
2. Realizar una aplicación en JAVA que solicite al usuario un numero entero N, cree un arreglo de String de tamaño N, llene el arreglo con datos pedidos al usuario, muestre el arreglo en pantalla.

Después de mostrar el arreglo en pantalla se debe solicitar al usuario ingresar un nombre y la aplicación debe mostrar en pantalla la posición donde se encuentra exactamente ese nombre en arreglo.

Para comparar variables de tipo String, se deben tener en cuenta que no son tipos de datos primitivos, por lo que no funciona la comparación de tipos primitivos (Ejemplo: `a == b`). Para comparar variables de tipo String se debe usar el método interno `.equals()` o `.equalsIgnoreCase()`

```
String nombre_1 = "Andres";
String nombre_2 = "Juan";

if ( nombre_1.equals( nombre_2 ) ){
    System.out.println( "Los nombres son iguales" );
}else{
    System.out.println( "Los nombres son diferentes" );
}
```

3. Realizar una aplicación en JAVA que muestre en la pantalla un menú con la siguiente forma:

```

                                Memoria disponible: 100 Registros.
1: Registrar cliente.
2: Ver lista de clientes.
3: Editar cliente.
4: Eliminar cliente.
5: Salir
```

Consideraciones:

- Cada vez que se visualice el menú, se debe mostrar en pantalla la cantidad de memoria disponible para almacenar registros.
- Registrar un cliente solo consiste en solicitar el nombre completo del cliente al usuario y almacenarlo en una posición dentro del arreglo.
- No se permite registrar clientes con nombres repetidos.
- Para editar un cliente, el sistema debe solicitarle al usuario ingresar el nombre del cliente que este almacenado, realiza la búsqueda de ese cliente en el sistema y en caso de encontrarlo se le solicita al usuario el nuevo nombre que tendrá.
- Para eliminar un cliente se debe tener en cuenta que todos los clientes están almacenados en un arreglo. En el arreglo no deben existir posiciones intermedias con datos vacíos, es decir que la ubicación de los datos debe encontrarse de forma secuencial, los espacios vacíos del arreglo van quedando al final del mismo.