

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 2 - Ciclo 2025</b>  <b>Tema: Arreglo</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

El trabajo práctico permite que cada estudiante conozca, distinga, desarrolle y utilice el tipo de dato “arreglo” desde el punto de vista de la programación orientada a objetos.

También se prevé la utilización de arreglos dinámicos, objetos y colecciones de objetos. Se requiere de la realización de una clase sencilla y la manipulación de objetos de esa clase.

### Ejercicios propuestos

- 1) Un negocio desea almacenar los precios de los productos que tiene a la venta. Para lograrlo, primero se debe solicitar la cantidad de precios a ingresar, y luego, realizar la carga. A través de un menú de opciones, el programa debe presentar las siguientes operaciones:
  - a) Calcular y mostrar la suma de todos los precios de los productos existentes.
  - b) Devolver otro arreglo en el que se aplique un descuento del 15% a los precios de los productos cuyo valor supere un monto ingresado por el usuario.
  - c) Devolver el promedio de los precios haciendo uso de un método cuyo parámetro es el arreglo original.
  - d) Obtener la cantidad de productos que tienen un precio inferior a un valor ingresado por el usuario.

#### Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 2) Desarrollar un programa que solicite al usuario un valor entero N. Generar N – 1 números aleatorios y cargarlos en un arreglo. Luego el programa debe implementar lo siguiente:
  - a) Un método que reciba como parámetro el arreglo inicial y devuelva otro arreglo que contenga los elementos iniciales pero ubicados una posición hacia la derecha.
  - b) Un método que muestre la cantidad de números pares e impares que contiene el arreglo inicial.
  - c) Un método para revertir los elementos del arreglo (el primer elemento pasa a la última posición y el último a la primera, el segundo elemento a la penúltima posición y viceversa, etc.).

#### Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**. También necesita del objeto random para generar valores de manera aleatoria.

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 2 - Ciclo 2025</b>  <b>Tema: Arreglo</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

- 3) Crear un programa para la gestión de pedidos. En primer lugar, solicitar la cantidad de pedidos a gestionar. Para cada pedido, el usuario debe ingresar un código de pedido y su estado actual, el cual puede ser: "Pendiente", "Procesando", "Enviado" o "Entregado". Estos datos deben almacenarse en arreglos paralelos (uno para el código y otro para el estado). El programa debe:
- Buscar un código de pedido específico y, si lo encuentra, mostrar su estado actual y su posición en el arreglo.
  - Obtener la cantidad de pedidos que se encuentran en un estado particular (ej. "Pendiente"), ingresado por el usuario.
  - Solicitar al usuario el ingreso de un estado y devolver un arreglo que contenga los códigos de pedidos que corresponden a ese estado.
  - Reorganizar el arreglo (intercambiando elementos) de manera que todos los pedidos "Pendientes" se muevan al inicio del arreglo, manteniendo su orden relativo. Mostrar el arreglo de códigos de pedidos y sus estados después de esta operación.

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso** y **validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 4) Realizar un programa que, en primer lugar, solicite al usuario una cantidad de caracteres a ingresar y luego haga el ingreso de cada caracter en un arreglo. Una vez cargado el arreglo, el programa debe:
- Obtener la cantidad de caracteres que solo aparecen una vez en el arreglo, es decir, los que no se repiten. Asimismo, estos caracteres deben mostrarse en pantalla.
  - Buscar y mostrar la posición de la primera y última aparición de un carácter específico ingresado por el usuario.
  - Obtener un nuevo arreglo que contenga únicamente los caracteres del arreglo inicial que son vocales. Luego, mostrar el arreglo resultante.
  - Intercambiar la posición de la primera vocal encontrada con la posición de la última consonante encontrada en el arreglo. Luego, mostrar el arreglo modificado. Si el arreglo no tiene vocales o consonantes, se debe generar un número aleatorio (entre 0 y la longitud del arreglo) y usar el caracter que se encuentra en esa posición para el intercambio.

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código controle los problemas que normalmente ocurren al operar con la consola o teclado.

Se espera una correcta modularización entre el código que realiza el ingreso y validación de los datos respecto del código que hace lo que se solicita en el ejercicio.

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 2 - Ciclo 2025</b>  <b>Tema: Arreglo</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

- 5) Crear la clase Componente con los atributos: codigo, nombre, precioUnitario y cantidadStock. El programa debe solicitar al usuario la cantidad de componentes a manejar. Luego, permitir el ingreso de diferentes componentes hasta llenar un arreglo. Además, el programa debe permitir:
- Actualizar la cantidad en stock de un componente dado su codigo (sumar o restar unidades).
  - Calcular y mostrar el stock total disponible en el arreglo.
  - Calcular y mostrar el monto total que se obtendría si se venden todos los componentes.
  - Identificar y mostrar el componente con la menor cantidad en stock.
  - Listar todos los componentes que tienen un stock inferior a un valor ingresado por el usuario. Mostrar un mensaje para indicar que necesitan ser reabastecidos

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 6) Crear un programa que simule una agenda telefónica, para lo cual se debe crear la clase Contacto con atributos como nombre, teléfono y correo electrónico. El usuario debe poder guardar cualquier cantidad de contactos. El programa debe permitir:
- Agregar nuevos contactos. Los nombres ingresados no se pueden repetir.
  - Buscar un contacto por su nombre y mostrar su información.
  - Solicitar el nombre de un contacto y modificar el número de teléfono o el correo electrónico.
  - Eliminar un contacto existente de la agenda

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso y validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.

- 7) Desarrollar un programa que permita organizar y gestionar diferentes eventos, controlando los participantes inscriptos en cada uno. Para ello, se debe crear la clase Participante con los atributos: idParticipante, nombre, apellido y correo electrónico. Además, se requiere otra clase llamada Evento que representa un acontecimiento. Esta clase debe tener los atributos: idEvento, nombreEvento, fecha, lugar, y un atributo para guardar varios objetos de tipo Participante llamado participantesInscriptos. El programa principal debe gestionar un arreglo de objetos Evento. A través de un menú, el usuario debe poder realizar las siguientes operaciones:

<b>Estructura de Datos</b>	<b>TRABAJO PRÁCTICO N° 2 - Ciclo 2025</b>  <b>Tema: Arreglo</b>
<b>INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS</b>  <b>FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY</b>	

- a) Crear un nuevo evento solicitando los datos de ingreso al usuario.
  - b) Inscribir un participante en un evento, para lo cual debe existir previamente un evento.
  - c) Solicitar al usuario el ingreso del idEvento y mostrar todos los participantes que están inscriptos en el mismo.
  - d) Solicitar idEvento e idParticipante para eliminar a dicho participante del evento.
  - e) Obtener la cantidad de participantes en un evento dado, ingresando el nombreEvento.
  - f) Buscar eventos por fecha: Mostrar los detalles de los eventos que coincidan.
- 8) Se desea cargar, en una matriz de dos dimensiones, las notas correspondientes a varias materias que han sido cursadas por un grupo de estudiantes. Para ello, a través de un programa, se debe solicitar al usuario el número de estudiantes y el número de materias. A continuación, ingresar las notas para cada estudiante en cada materia. Una vez cargadas, el programa debe:
- a) Calcular y mostrar el promedio de calificaciones de cada estudiante.
  - b) Calcular y mostrar el promedio de calificaciones de cada asignatura.
  - c) Encontrar y mostrar la calificación más alta y más baja de toda la matriz, indicando la posición (fila y columna) en la que se encuentran.
  - d) Obtener la cantidad de estudiantes que aprobaron (nota  $\geq 6.0$ ) todas las asignaturas.
  - e) Obtener la cantidad de estudiantes que aprobaron (nota  $\geq 6.0$ ) una asignatura indicada por el usuario.
  - f) Crear un arreglo que contenga las mayores notas obtenidas por cada alumno.

Indicaciones:

Este ejercicio necesita del objeto scanner para ingresar datos por la consola o teclado, se espera que el código **controle los problemas** que normalmente ocurren al operar con la consola o teclado.

Se espera una **correcta modularización** entre el código que realiza el **ingreso** y **validación** de los datos respecto del código que hace lo que se **solicita en el ejercicio**.