

Estructura de Datos	TRABAJO PRÁCTICO N° 1 - Ciclo 2025 Tema: POO
INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY	

El trabajo práctico permite que el estudiante conozca y aplique conceptos básicos de la programación orientada a objetos.

Para la resolución de los ejercicios se requiere del desarrollo de clases sencillas y la manipulación de objetos de esas clases.

Ejercicios propuestos

- 1) Diseñar una clase llamada Rectangulo que represente una figura geométrica. Esta clase debe tener los atributos ancho y alto. La clase Rectangulo debe incluir:
 - a) Constructor predeterminado que inicializa ambos valores en cero.
 - b) Un constructor con argumentos para establecer las dimensiones (ancho y alto) del rectángulo.
 - c) Un método que devuelva el área del rectángulo.
 - d) Un método que devuelva el perímetro del rectángulo.

En el programa principal (main) se debe:

- e) Crear una única instancia de la clase Rectangulo.
- f) Solicitar al usuario que ingrese el ancho y el alto del rectángulo verificando que los valores sean positivos.
- g) Utilizar los métodos de la clase Rectangulo para establecer estas dimensiones.
- h) Calcular y mostrar el área y el perímetro del rectángulo creado.

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) Crear una clase llamada Producto que modele un artículo de inventario. Esta clase debe contener los siguientes atributos: nombre, precio y stock. La clase Producto debe implementar:
 - a) Constructor predeterminado que inicializa el nombre con "Desconocido", precio y stock en 0.
 - b) Constructor con argumentos que permite establecer el nombre, precio y stock del producto mediante argumentos.
 - c) Un método vender que simule la venta de una cierta cantidad del producto. Debe disminuir el stock y retornar el monto total de la venta. Si la cantidad solicitada excede el stock disponible, no debe realizar la venta y debe retornar el valor 0, además debe mostrar un mensaje de error.

Estructura de Datos	TRABAJO PRÁCTICO N° 1 - Ciclo 2025 Tema: POO
INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY	

- d) Un método reponer que simule la reposición de stock, aumentando la cantidad disponible.
- e) El método toString() que imprima en consola el nombre, precio y stock actual del producto.

En el programa principal (main) se debe:

- f) Crear una instancia de Producto.
- g) Inicializar el producto con algunos valores (ej. "Notebook", 1500.0, 20).
- h) Permitir al usuario, a través de un menú sencillo, realizar operaciones de venta o reposición para este producto, mostrando los detalles actualizados después de cada operación. El menú debe repetirse hasta que el usuario decida salir.

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**.

- 3) Desarrollar un programa que simule la gestión de licencias con fecha de emisión y fecha de vencimiento, incorporando la funcionalidad de renovación.

El programa debe incluir:

- a) Una Clase Licencia con los atributos: numeroidentificacion único para la licencia, tipoLicencia que podría ser "Conducir", "Porte de Armas" o "Habilitación Comercial", fechaEmision y fechaVencimiento.
- b) Constructor predeterminado que inicializa numeroidentificacion en 000, tipoLicencia "a definir", fechaEmision con la fecha actual y fechaVencimiento igual a un mes después de la fecha actual.
- c) Constructor parametrizado que permite establecer el numeroidentificacion, tipoLicencia, fechaEmision y fechaVencimiento mediante argumentos.
- d) Método toString() para retornar una representación textual completa y legible del objeto Licencia.
- e) Método auxiliar para validar fechas que comprueba si las fechas de inicio y fin son válidas (la fecha de inicio debe ser anterior a la fecha de fin y ambas deben estar dentro de los próximos 30 días). Si las fechas no son válidas, se ajustan automáticamente del mismo modo que en el constructor predeterminado.
- f) Un método para determinar si la licencia está vigente en una fechaConsulta dada por el usuario. Debe retornar true si la fechaConsulta es igual o posterior a fechaEmision y es igual o anterior a fechaVencimiento. De lo contrario, retorna false.
- g) Método para calcular la cantidad de días restantes para el vencimiento de la licencia desde una fechaConsulta, o la cantidad de días que lleva vencida si ya caducó. Si la licencia aún no ha vencido,

Estructura de Datos	TRABAJO PRÁCTICO N° 1 - Ciclo 2025 Tema: POO
INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY	

debe retornar un valor positivo que represente los días restantes hasta fechaVencimiento desde fechaConsulta. Si la licencia ya está vencida, debe retornar un valor negativo que represente los días que han transcurrido desde fechaVencimiento hasta fechaConsulta. Si la fecha de vencimiento es igual a la fecha de consulta, debe retornar 0.

- h) Método `simularRenovacion(int cantidadAnios)` que calcula la nueva fecha de vencimiento si la licencia se renovara por una cantidad específica de años a partir de su fecha de vencimiento actual.

En el programa principal (main) se debe:

- i) Solicitar al usuario los datos para crear al menos dos objetos Licencia diferentes, por ejemplo: una licencia vigente y una licencia vencida. Se debe realizar las validaciones correspondientes.
- j) Después de crear las licencias, el programa debe solicitar al usuario una "fecha de consulta" (día, mes, año) para verificar el estado de las licencias. Esta fecha también debe ser validada.

Para cada objeto Licencia creado, el programa debe:

- k) Mostrar sus detalles utilizando el método `toString()`.
- l) Llamar al método `estaVigente()` pasándole la "fecha de consulta" y mostrar el resultado.
- m) Llamar al método `calcularDiasRestantesOVencidos()` pasándole la misma "fecha de consulta" y mostrar la cantidad de días restantes o vencidos.
- n) Solicitar al usuario una cantidad de años para simular una renovación, llamar al método `simularRenovacion()` y mostrar la nueva fecha de vencimiento calculada (sumar fechas).

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) Desarrollar una clase llamada Estudiante para gestionar información académica básica. Esta clase debe tener los siguientes atributos: nombre, lu, notaPrimerParcial, notaSegundoParcial. La clase Estudiante debe incluir los siguientes métodos:
 - a) Un método para establecer el nombre, la lu y las notas del estudiante.
 - b) Un método que devuelva el promedio de las dos notas.
 - c) Un método que devuelva el String "Promoción", si el promedio es mayor o igual a 7, "Regular" si el promedio esta entre 4 y 7, "Libre" si el promedio es menor a 4.

Estructura de Datos	TRABAJO PRÁCTICO N° 1 - Ciclo 2025 Tema: POO
INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY	

En el programa principal (main) se debe:

- d) Crear una instancia de la clase Estudiante.
- e) Solicitar al usuario que ingrese el nombre, la lu del estudiante y sus dos notas.
- f) Utilizar los métodos de la clase Estudiante para establecer estos valores.
- g) Mostrar el nombre del estudiante, la lu, el promedio y su estado de aprobación.

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**.

- 5) Crear un programa para gestionar un inventario de vehículos aplicando el concepto de herencia. Definir una clase base Vehiculo con atributos comunes como marca, modelo, año de fabricación y precio. Esta clase debe contener el método llamado obtenerDescripcionBasica() que retornará un String con la marca, modelo y año de fabricación del vehículo. El método calcularAntigüedad(int anioActual) que recibirá el año actual como parámetro y retornará la antigüedad del vehículo en años (calculada como anioActual - anioFabricacion).

Crear las siguientes clases derivadas

- a) Moto: con los atributos cilindradaCC y tipoManillar (Deportivo, Cruiser, Enduro, etc.). Método obtenerClasificacionCilindrada(), devuelve "Baja" para cilindradas hasta 250cc, "Media" para cilindradas entre 251cc y 600cc, y "Alta" para cilindradas superiores a 600cc.
- b) Auto: con los atributos numeroPuertas, tipoCombustible (Gasolina, Diesel, Eléctrico), capacidadBaulLitros. Método esFamiliar() devuelve true si numeroPuertas es 4 o 5, y false en cualquier otro caso. Método esEcoAmigable() devuelve true si tipoCombustible es "Eléctrico" o "Híbrido", de lo contrario false.
- c) Camion: con los atributos capacidadCarga, numeroEjes, tipoCarroceria (Volquete, Caja Seca, Plataforma). Método requiereLicenciaEspecial() devuelve true si el numeroEjes es mayor a 2, y false en caso contrario. Método describirCapacidadesEspeciales() que devuelve un String diferente según el valor de tipoCarroceria, si es "Volquete", devuelve "Ideal para movimiento de tierras y materiales a granel"; si es "Caja Seca", devuelve "Apto para transporte de mercancías protegidas", en cualquier otro casos devuelve "Útil para todo tipo de carga".

En el programa principal (main) se debe permitir al usuario:

Estructura de Datos	TRABAJO PRÁCTICO N° 1 - Ciclo 2025 Tema: POO
INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY	

- d) Crear una instancia de Auto: El programa debe solicitar al usuario todos los datos necesarios para inicializar tanto los atributos heredados de Vehiculo como los atributos propios de Auto.
- e) Crear una instancia de Moto: De manera similar, solicitará todos los datos para sus atributos heredados y propios.
- f) Crear una instancia de Camion: Solicitará todos los datos para sus atributos heredados y propios.

Para cada uno de los vehículos creados, el programa debe realizar las siguientes acciones y mostrar los resultados:

- g) Mostrar la descripción básica del vehículo utilizando el método obtenerDescripcionBasica() que ha sido heredado de la clase Vehiculo.
- h) Calcular y mostrar la antigüedad del vehículo utilizando el método calcularAntigüedad() que ha sido heredado de la clase Vehiculo. Para esto, el anioActual puede ser un valor fijo (ej. 2025) o solicitado al usuario.
- i) Finalmente, mostrar por pantalla todos los detalles completos de cada vehículo de manera formateada, incluyendo tanto los atributos heredados (marca, modelo, año, precio) como los atributos propios de su respectiva subclase.

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) Escribir un programa que calcule el área y el perímetro de diferentes figuras geométricas utilizando herencia y demostrando polimorfismo.

Definir una clase base FiguraGeometrica con los atributos color y nombre, y métodos para:

- a) Mostrar la información básica de la figura (color y nombre).
- b) Calcular el área (calcularArea), que será implementado en las clases derivadas. Este método debe devolver 0, ya que será sobrescrito en la clase derivada.
- c) Calcular el perímetro (calcularPerimetro), que también será implementado en las clases derivadas. Al igual que el ítem anterior, este método debe devolver 0, ya que será sobrescrito en la clase derivada.
- d) Crear las clases derivadas Circulo y Rectangulo que hereden de FiguraGeometrica e implementen los métodos de cálculo de área y perímetro de acuerdo a sus propiedades específicas (radio para el círculo; ancho y alto para el rectángulo).

Estructura de Datos	TRABAJO PRÁCTICO N° 1 - Ciclo 2025 Tema: POO
INGENIERÍA INFORMÁTICA – LICENCIATURA EN SISTEMAS FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE JUJUY	

El programa debe permitir al usuario:

- e) Crear una instancia de Circulo ingresando sus dimensiones y mostrar su área y perímetro.
- f) Crear una instancia de Rectangulo ingresando sus dimensiones y mostrar su área y perímetro.

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.