



**Universidad Nacional de San Luis**  
**Área de Programación y Metodologías de Desarrollo de Software**  
**Ingeniería en Informática – Ingeniería en Computación**  
**Programación II**  
**Práctico N° 2**  
*Lenguaje Java: Clases*

**Ejercicio 1:** define los siguientes conceptos: a) Clase, b) Objeto, c) Constructor, d) Estado del objeto, e) Variables de Instancia y f) Método. ¿Cómo se implementan los destructores en Java?

**Ejercicio 2:** explica cada uno de los modificadores de acceso que provee Java.

**Ejercicio 3:** define la clase auto, la misma posee las siguientes variables de instancia: nombre, marca, tipo, color. Define los métodos observadores (get) y modificadores (set) para cada una de las variables de instancia.

**Ejercicio 4:** crea la clase Fracción con métodos para sumar, restar multiplicar y dividir fracciones. Tenga en cuenta de agregar las variables de instancia necesarias.

**Ejercicio 5:** crear una clase Libro que contenga los siguientes atributos:

- ISBN
- Título
- Autor
- Número de páginas

Crear sus respectivos métodos get y set correspondientes para cada atributo.

Crear el método toString() para mostrar la información relativa al libro con el siguiente formato: “El libro *título* con *ISBN* creado por *autor* tiene *número de páginas* páginas”

**Ejercicio 6:** implementa la clase Fecha que permite almacenar y manipular una fecha. La clase posee las siguientes variables de instancia: día, mes y año. Define los siguientes métodos:

- constructores (por defecto e inicializar fecha),
- observadores y modificadores para las variables de instancia,
- cantDias(Fecha d) que retorna la cantidad de días entre la fecha que envía el mensaje y una fecha pasada como parámetro.
- mostrarFecha() muestra la fecha por pantalla (en formato fecha “dd/mm/aaaa”).

**Ejercicio 7:** define la clase Persona, la misma posee las siguientes variables de instancia: nombre, tipo de documento, numero de documento, estado civil y fechaNacimiento. El atributo fechaNacimiento es un objeto de la clase Fecha creada en el ejercicio anterior. Define los métodos observadores y modificadores para cada una de las variables de instancia. Define el método mostrarPersona(), que muestra por pantalla los datos de una persona. ¿Qué harían los métodos “public String datosPersona()” y “... clonar()”?



**Ejercicio 8:** desarrolla la clase Materia, la cual posee las siguientes variables de instancia: nombre, año del curso (1°, 2°, 3°, 4° o 5°), cantidad de estudiantes y cuatrimestre. Define los métodos necesarios para la manipulación de una materia. ¿Es posible mantener la cantidad de materias que se han creado en total?.

**Ejercicio 9:** desarrolla un programa que permita al usuario ingresar una cantidad n de materias y completar todos sus campos. Una vez finalizada la carga de materias, debe imprimir todas las materias del año que el usuario indique.

**Ejercicio 10:** implemente la clase PilaPersonas usando un arreglo como estructura base. Define los métodos y variables de instancia que considere necesarios. ¿Puede utilizar alguna clase de algún ejercicio anterior?

**Ejercicio 11:** realiza un programa que provea un menú de opciones para:

- cargar n materias.
- modificar los datos de una materia.
- cargar cantidad de estudiantes a una materia.
- mostrar todas las materias que posean más de 20 estudiantes.

¿Es posible que la materia contenga el listado de estudiantes que cursan la misma? ¿Se puede usar la clase Persona para esto? Defina/reutilice las clases que considere necesarias.

**Ejercicio 12:** define la clase Password que contenga las variables de instancia longitud y contraseña. Por defecto la longitud será de 8 caracteres. Define los siguientes métodos:

- un constructor por defecto.
- un constructor que inicialice la longitud con un parámetro.
- métodos get y set para cada atributo.
- esSegura() que devuelve verdadero si la clave es segura (tiene 2 mayúsculas, más de una minúscula y más de 3 números).
- generarContraseña(...) genera la contraseña del objeto con la longitud que el mismo posea.

**Ejercicio 13:** Lleva a cabo las siguientes tareas:

- implementa la clase Serie, la cual permite almacenar los datos de una serie de televisión (o web). La clase deberá almacenar los siguientes datos: título de la serie, género y temporadas. Para las temporadas se almacenan los siguientes datos: año, título y capítulos (número y título de cada capítulo en un string). Además de todos los métodos que considere necesarios para la implementación, debes implementar los siguientes métodos:
  - los constructores que crea convenientes para la serie,
  - agregarTemporada(...) que permite agregar una nueva temporada a la serie,
  - eliminarTemporada(...) que permite eliminar una temporada indicada por el usuario (con el título),
  - buscarCapítulo(...) permite consultar el título de un capítulo.
- Desarrolla un programa que permita al usuario definir y operar con una Serie, haciendo uso de TODOS los métodos implementados. Muestre los resultados por pantalla.

**Ejercicio 14:** una empresa de construcción posee dos tipos de trabajadores el constructor y el consultor. El primero es aquel que se encarga de construir de realizar tareas de construcción. El segundo tiene



como principal objetivo interactuar con el cliente y convencerlo de llevar a cabo una operación de construcción. La empresa registra para todos los trabajadores los siguientes datos: Nombre, Domicilio, Teléfono, Fecha de Nacimiento y Fecha de Inicio de Contrato. Además de esos datos en común, la empresa registra para los trabajadores que son constructores el sueldo y los impuestos y para los trabajadores que son consultores, las horas trabajadas y la tarifa. Se pide, construye un programa en Java que:

- permita el ingreso de  $n$  trabajadores de la construcción a la empresa,
- permita el ingreso de  $k$  consultores a la empresa,
- calcule el monto mensual que la empresa le tiene que pagar a los constructores y
- informe el consultor que trabajo más horas.

*Nota:* Define las clases que creas convenientes para la solución de esta realidad.

### EJERCICIOS COMPLEMENTARIOS

**Ejercicio 1:** explica porque en algunas situaciones es importante implementar el método `clonar()` para una clase determinada.

**Ejercicio 2:** ¿Cómo se identifica un constructor en Java? ¿Cuántos constructores se pueden definir?.

**Ejercicio 3:** implementa la clase `NumeroComplejo`, que permite definir y manipular número complejos. Además de definir las variables de instancia y métodos que consideres necesarios. También, debes implementar los siguientes métodos:

- un constructor que inicialice un número complejo.
- `sumaComplejo()` que permite sumar dos complejos y devolver el resultado.
- `restaComplejo()` que permite restar dos complejos y devolver el resultado.
- `clonar()` que clonar el objeto que envíe el mensaje.
- `mostrarComplejo()` que imprime por pantalla el objeto.

Desarrolla un programa que permita al usuario definir y operar con números complejos, mostrando los resultados por pantalla.

**Ejercicio 4:** Realiza las siguientes actividades:

- implementa la clase `ConjuntoEnteros` la cual permite almacenar un conjunto de enteros. Además de todos los métodos que consideres necesarios para la implementación de esta clase, debe implementar los siguientes métodos:
  - un constructor que reciba como parámetro un arreglo de enteros y genere el conjunto (CUIDADO: en este arreglo puede haber objetos repetidos),
  - `agregarEntero(...)` que permite agregar entero al conjunto.
  - `eliminarEntero(...)` que permite eliminar un entero del conjunto.
  - `consultarEntero(...)` permite consultar si un entero existe en el conjunto.
  - `interseccion(...)` toma como argumento otro conjunto, y devuelve un nuevo conjunto con la intersección de los dos.
  - `union(...)` toma como argumento otro conjunto, y devuelve un nuevo conjunto con la unión de ambos.
- Desarrolle un programa que permita al usuario definir y operar con conjuntos de enteros,



haciendo uso de TODOS los métodos implementados. Muestre los resultados por pantalla.

*Nota 1:* en un conjunto, no puede haber objetos repetidos y los mismos no se encuentran ordenados.

**Ejercicio 5:** implementa la clase FilaMateria usando un arreglo como estructura base. Define los métodos y variables de instancia que considere necesarios.

**Ejercicio 6:** implementa un sistema simple de emisión de facturas para un comercio. Cada factura posee las siguientes variables de instancia: numero, tipo (A, B o C), comprador y total. Además, una factura se compone de “items” que poseen las siguientes variables de instancia: código de producto, denominación, cantidad y precio. Permite al usuario ingresar los campos de una factura e imprimirla por pantalla. Ten en cuenta la visualización de la factura. ¿Cambia algo si la factura es de tipo A?

**Ejercicio 7:** teniendo en cuenta el ejercicio de los billetes (ejercicio h, práctico 1), implementa el mismo programa del cajero considerando una capacidad limitada de billetes. Es decir, el cajero debe considerar la cantidad de billetes que posee para poder expender las sumas solicitadas.

**Ejercicio 8:** en base al ejercicio 12 cree un programa que:

- permita al usuario ingresar y almacenar n contraseñas. Posteriormente se deben imprimir todas las contraseñas indicando si son seguras o no.
- si la contraseña ingresada posee más caracteres que la longitud por defecto, entonces debería informar al usuario mediante un estado de error (entero).

**Ejercicio 9:** desarrolla un programa que permita al usuario administrar un video club. Este debe posibilitar las altas de nuevas películas. Además, debe registrar la cantidad de ejemplares de cada película y gestionar los préstamos de las mismas. Define las clases que consideres necesarias.