Programación II – ALGORTIRMOS Y ESTRUCTURAS DE DATOS II

**Docente:** Esp. Lic. PEREZ, Nicolás Ignacio

[nicoperez@uade.edu.ar](mailto:nicoperez@uade.edu.ar)

Repositorio de la materia: <https://github.com/NicolasPerezUNLaSMN/PROG_II_UADE_JAVA>

# Facultad de Ingeniería y Ciencias Exactas

Universidad Argentina de la Empresa



# Guía de Programación en Java con TDA

## 1. Instalación de Eclipse

1. Visita la web oficial: https://www.eclipse.org/downloads/

2. Descarga Eclipse IDE for Java Developers.

3. Ejecuta el instalador y sigue los pasos.

4. Elige la carpeta de instalación y haz clic en Install.

5. Una vez finalizada la instalación, haz clic en Launch.

6. Configura el workspace para almacenar tus proyectos.

## 2. ¿Qué es el JDK (Java Development Kit)?

El JDK es el kit de desarrollo de Java e incluye:

- JRE (Java Runtime Environment): Para ejecutar programas Java.

- Compilador (javac): Convierte el código fuente en bytecode.

- Herramientas de desarrollo como jar, javadoc, jdb (depurador).

## Verificar versión del JDK

En la terminal (CMD o Terminal en Mac/Linux), escribe:

java -version

javac -version

Si ves algo como java version "17.0.2", significa que Java está instalado correctamente.

## 3. Instalación de Java en Visual Studio Code (VS Code)

1. Descarga desde la web oficial: https://code.visualstudio.com/

2. Instala y marca la opción 'Agregar al PATH'.

## Configurar el JDK en VS Code

1. Descarga el JDK desde Oracle o Adoptium.

2. Agrega la ruta del JDK al PATH en las Variables de Entorno.

3. Verifica con java -version en la terminal.

## Instalar extensiones en VS Code

1. Abre Extensiones en VS Code.

2. Instala:

- Extension Pack for Java

- Debugger for Java

- Maven for Java

3. Reinicia VS Code.

## 4. Programación con TDA en Java

Un TDA define qué hace una estructura de datos u objeto, pero no cómo lo hace internamente.

Ejemplo de TDAs en Java: Listas, Pilas, Colas, Conjuntos.

## Diferencia entre Métodos Estáticos y Abstractos

Método Estático: No requiere instancia y no se puede sobrescribir.

Método Abstracto: Se debe implementar en una subclase y requiere instancia.

## 6. Organización en Paquetes

Los paquetes organizan el código y facilitan la reutilización.

Estructura del proyecto:

/src

│── datos → Manipulación de archivos o bases de datos.

│── modelo → Clases que representan entidades.

│── funciones → Métodos auxiliares.

│── interfaces → Define contratos (IVehiculo, IPersona).

│── servicios → Lógica de negocio.

│── test → Pruebas unitarias.

## 7. Conclusión

✅ Eclipse y VS Code son entornos válidos para programar en Java.

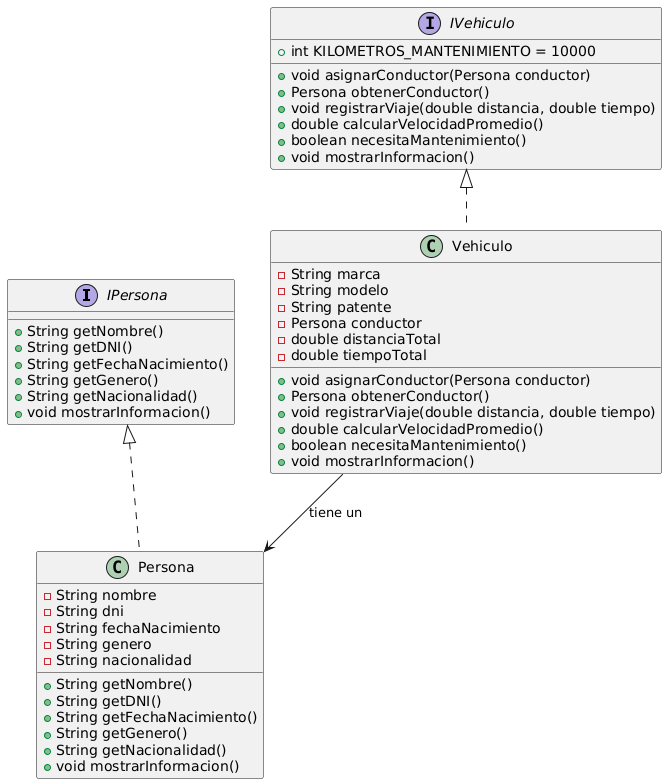
✅ Eclipse es mejor para proyectos grandes, VS Code es más ligero.

✅ La programación con TDA en Java se basa en interfaces y clases.

✅ Los paquetes organizan el código y facilitan su mantenimiento.

## 8. Objetivos de la clase

* Crear el primer proyecto Java y separar métodos en interfaces y clases.
* Trabajar con los métodos clásicos de arreglos de enteros
* Aprender a instanciar clases e invocar métodos
* Comprender las pre, post y axiomas bajo el paradigma de TDA
* Crear la primer Clase con atributos, ver como eso se relaciona con la interfaz.
* Analizar las palabras reservadas nuevas del lenguaje, abstract, public, void, etc
* Completar la interface IArreglosEnteros del ejemplo 1
* Comentar, usar, completar y consultar sobre el proyecto Clase2TDA
* Debatir y empezar a entender cómo se lee un Diagrama de Clases en JAVA



# 9. Tarea para practicar

Luego de comprender, comentar y “jugar” con el código fuente de la clase 2, Clase2TDA, disponible en el repositorio:

<https://github.com/NicolasPerezUNLaSMN/PROG_II_UADE_JAVA/tree/master/Clase%20-%20II>

Crear un nuevo TestVehiculos, en el cual se creen 3 Vehiculos y se le asigne un conductor a cada uno. Para eso crear un Array de Vehiculos (Vehículo[] ), que todos los datos de los vehículos y personas se ingresen por teclado (Scanner), y agregar la validación en el caso que en el género de la Persona no sea ni masculino, ni femenino ni otro, se vuelva a pedir otro género.

# 10. Próxima clase

a. Le prestaremos más atención al concepto de PRE; POST y AXIOMAS

b. Explicaremos el concepto de @Override

c. Relacionaremos más de dos Clases entre sí.

d. Comenzaremos a trabajar con estructuras de datos más complejas y estáticas.