

UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO  
Facultad de Ciencias



**Introducción a las Ciencias de la Computación**

*Práctica 01: Variables de Tipos Primitivos*

Profesora:

Amparo López Gaona

Ayudante:

Adrián Aguilera Moreno

Ayudante de Laboratorio:

Kevin Jair Torres Valencia

## Objetivo

El alumno se familiarizara con la teoría acerca del uso y aplicación, de variables de tipo primitivo, así como operadores sobre ellas, trabajando con la declaración e inicialización de datos primitivos.

## Introducción

En todo programa en Java es preciso asignar un identificador (o nombre) a cada elemento que se utiliza, sea éste clase, atributo, método, objeto, dato, etc. Un identificador se construye como una sucesión de caracteres alfanuméricos.

Existe un conjunto de palabras que no pueden ser usadas como identificadores porque tiene un significado especial para java, cada una de éstas se denomina palabra reservada. Los tipos de datos definidos en java, y denominados primitivos, son:

- Tipo numérico (enteros o reales).
- Tipo carácter.
- Tipo Booleano.

| Tipo                 | Descripción   |
|----------------------|---|
| <code>byte</code>    | Entero de 8 bits                                      |
| <code>short</code>   | Entero de 16 bits                                     |
| <code>int</code>     | Entero de 32 bits                                     |
| <code>long</code>    | Entero de 64 bits                                     |
| <code>float</code>   | Real en 32 bits<br>con 7 dígitos en la parte decimal  |
| <code>double</code>  | Real en 64 bits<br>con 15 dígitos en la parte decimal |
| <code>char</code>    | Carácter en 16-bits (Unicode)                         |
| <code>boolean</code> | Booleano  |

Para declarar una variable se escribe el tipo del dato seguido de su identificador y al final un punto y coma. Por ejemplo, la declaración `int edad;` define la variable `edad` de tipo entero.

Para trabajar con los datos primitivos se tienen los operadores de la siguiente tabla

| Operador       | Descripción            |
|----------------|------------------------|
| <code>+</code> | Suma                   |
| <code>-</code> | Resta                  |
| <code>*</code> | Multiplicación         |
| <code>/</code> | División               |
| <code>%</code> | Residuo de la división |

## Prólogo

En un reino no muy lejano, donde los VHS aún reinan y los dragones coleccionan películas en lugar de oro, nace un negocio peculiar: **Trash the Dragon**, el último bastión de entretenimiento físico en la era digital. Este Blockbuster medieval, custodiado por un dragón cinéfilo, necesita urgentemente una solución moderna para sobrevivir al caos de los registros manuales, las reservas perdidas y los clientes impacientes.

Durante el semestre **2026-1**, los valientes alumnos de **Introducción a Ciencias de la Computación** se embarcarán en una misión épica: desarrollar un programa que administre este negocio fantástico. Desde la gestión de películas y clientes hasta el control de préstamos y devoluciones, cada línea de código será una espada afilada contra el desorden.

¿Lograrán domar al dragón y convertir Trash The Dragon en el imperio digital que merece ser? ¡La aventura apenas comienza!

## Desarrollo

El blockbuster **Trash The Dragon** necesita registrar información básica de una película para construir su catálogo. Para esta primera práctica deberás:

1. Declarar variables de tipos primitivos en una clase llamada `Practica1`.
2. Guardar en ellas los datos de una película:
  - Año de estreno
  - Duración en minutos
  - Precio de renta
  - Calificación por estrellas
  - Disponibilidad
  - Clasificación de edad
3. Calcular el precio final de la renta aplicando:
  - IVA = 16% del precio base.
  - Descuento del 10% si la película tiene más de 10 años.
4. Imprimir en consola todos los valores y el resultado del precio final.

## Restricciones:

1. No se puede usar `if`, `switch`, `for`, `while`, `for-each`, entre otros
2. Solo pueden hacer uso de operaciones aritméticas y de asignación.

## Formato de Entrega

1. Las prácticas se entregarán en parejas.
2. NO incluir los archivos .class dentro de la carpeta.
3. Los archivos de código fuente deben estar documentados.
4. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
5. La práctica se debe subir al Github Classroom correspondiente.
6. La entrega en classroom debe contener el link HTTPS de su repositorio y es lo único que se debe entregar.
7. **Nota: Para alcanzar una calificación superior a 5, la práctica debe compilar correctamente y sin advertencias, en caso contrario tendrán 0 en automático**