



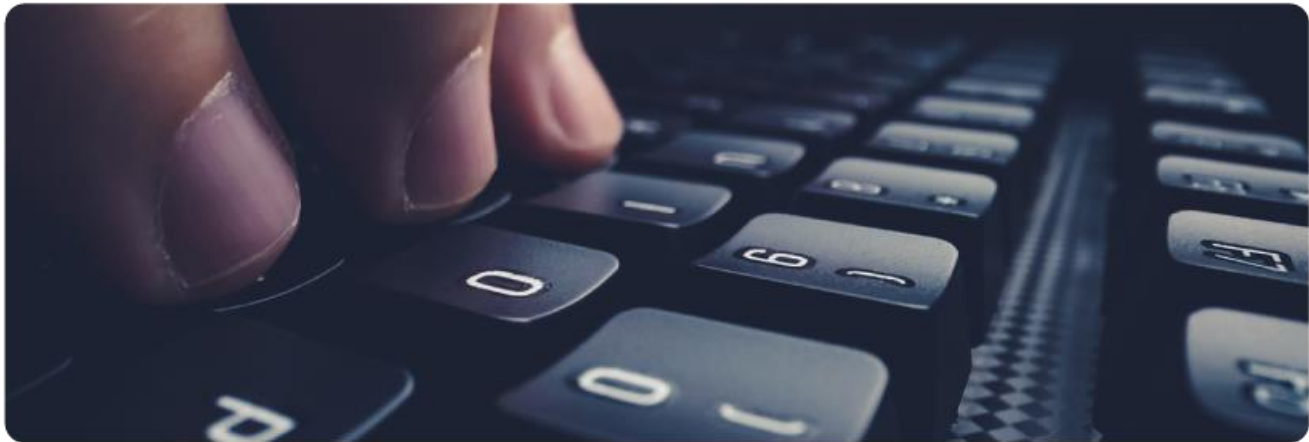
## Ciclo 2

# Semana 1 y 2

## *Introducción a la Programación Orientada a Objetos (POO)*

### Reto 1

## C2 - Semana 1 y 2 - Reto 1



### Considere el siguiente contexto

La alcaldía de Bogotá ha recibido en los últimos tiempos muchas denuncias por abusos en el cobro de los servicios públicos, como principio de solución a este tema, el distrito ha decidido realizar el desarrollo de una aplicación de escritorio por consola que permita de al usuario detectar si le están haciendo un cobro no apropiado a su consumo de cada servicio público.

Usted ha sido contratado por el distrito para desarrollar la aplicación que permite a los usuarios a saber cuánto deben pagar por Servicio Público: **AGUA, LUZ o GAS NATURAL**, bajo las siguientes condiciones:

- Si el estrato socioeconómico donde se encuentra el predio es 1 o 2: hay un descuento del 50% sobre el total de la factura.
- Si el estrato socioeconómico donde se encuentra el predio es 3 o 4: hay un descuento del 10% sobre el total de la factura.
- Si el estrato socioeconómico donde se encuentra el predio es 5 o 6: hay un sobrecosto del 25% sobre el total de la factura.
- Para el GAS NATURAL, el valor del metro cúbico es \$5234.
- Para el AGUA, el valor del metro cúbico es \$9256.
- Para el servicio de ENERGÍA (LUZ), el valor de cada KW (kilowatt) es de \$6200.



## Semana I

Introducción a la Programación Orientada a Objetos (POO)

El valor por factura de cada uno de los servicios se calcula teniendo en cuenta la siguiente fórmula: **CONSUMO x VALOR UNIDAD**

El usuario debe poder escoger qué servicio público quiere pagar interactuando con un MENÚ (**Ayuda: Haga uso de Switch/Case**) y considerando todas las condiciones presentadas previamente (**Ayuda: Haga uso de if/else if/else**).

Posdata: debido a los daños en infraestructura ocasionados por hinchas de equipos de fútbol, todas las facturas tienen un impuesto del 1% sobre el valor de la factura.

Desarrolle la solución utilizando el lenguaje de **programación JAVA y el IDE Netbeans**. Diseñe un diagrama de clases que siga el estilo arquitectural MVC, en donde queden representados: paquetes, clases, atributos, métodos y relaciones.