

Qué vamos a aprender en este nivel:

Modelar **características** de un objeto

- Utilizando la técnica de definir **constantes** para representar los valores posibles de un atributo
- Utilizar **expresiones**
 - Medio para identificar una situación posible en el estado de un objeto y para indicar la manera de modificar dicho estado
- Utilizar **instrucciones condicionales** simples y compuestas como parte del cuerpo de un método
 - Para considerar distintos casos posibles en la solución de un problema
- Clasificar los **métodos** de una clase:
 - Constructores
 - Modificadores
 - Analizadores
- Ciclos

Caso de Estudio: La Tienda

La Tienda

- Vende **cuatro** productos.
- Cada producto tiene:
 - **Nombre**
 - **Tipo**: papelería, supermercado o droguería
 - **Cantidad** actual del producto en la tienda (número de unidades disponibles para la venta que hay en bodega)
 - Número de productos por debajo del cual se debe hacer un nuevo **pedido** al proveedor
 - **Precio** base de venta por unidad
- Para calcular el **precio final** de cada producto, se deben sumar (al precio de base) los impuestos que define la ley (IVA):
 - 16% para los productos de papelería
 - 4% para los productos de supermercado
 - 12% para los productos de droguería

La Tienda

- El programa de manejo de esta tienda debe permitir:
 - Vender al cliente un cierto número de unidades del producto
 - Hacer un pedido de un producto para el cual ya se llegó al tope mínimo definido
 - Mostrar algunas estadísticas de la tienda:
 - El producto mas vendido
 - El producto menos vendido
 - La cantidad total de dinero obtenido por las ventas de la tienda
 - El promedio de ventas de la tienda (valor total de las ventas dividido por el número total de unidades vendidas)

Requerimientos Funcionales

- R1: Vender un producto.
- R2: Hacer pedido de un producto.
- R3: Calcular estadísticas de la empresa.

La Tienda

Latinoamericana: Tienda

Productos

Producto	Cantidad	IVA	Precio	Pedido
lápiz	18	16.0%	\$550.0	NO
aspirina	25	12.0%	\$109.5	NO
borrador	30	16.0%	\$207.3	NO
pan	15	4.0%	\$150.0	SI

Operaciones

Cálculos

Ingresos	\$ 0
Producto más vendido	lápiz
Producto menos vendido	lápiz
Promedio	\$ 0

R1 points to the "Vender Producto" button.

R2 points to the "Pedir Producto" button.

R3 points to the "Cálculos" section.

Requerimientos Funcionales

Nombre	R1 – Vender un producto
Resumen	Vender a un cliente un cierto número de unidades de un producto
Entradas	
<ol style="list-style-type: none">1. el nombre del producto2. la cantidad de unidades	
Resultado	
<p>Si había suficiente cantidad del producto disponible, se vendió al cliente lo que pidió. Si no, se le dio todo lo que se tenía en la bodega de la tienda. En la caja de la tienda se guardó el dinero resultado de la venta. El cliente fue informado de la cantidad que se le vendió.</p>	

Requerimientos Funcionales

Nombre	R2 – Hacer pedido de un producto
Resumen	Hacer un pedido de un producto para el cual ya se llegó al tope mínimo definido
Entradas	
Resultado	
Si es posible realizar el pedido, se incrementa la cantidad del producto disponible con el número de unidades que se pidieron.	

Requerimientos Funcionales

Nombre	R3 – Calcular estadísticas de ventas.
Resumen	Mostrar las siguientes estadísticas: (a) el producto más vendido (b) el producto menos vendido (c) la cantidad total de dinero obtenido por las ventas de la tienda (d) el promedio de ventas de la tienda.
Entradas	
Ninguna	
Resultado	
Se ha presentado por pantalla la información estadística antes descrita.	

Nuevos elementos de modelaje

Tipos simples de datos

- Entero (int) 562
12
- Real (double) 564.78
-98.3
- String (cadena de caracteres) “esta es una cadena”
“”
“ “

Ejemplo tipo double en la Tienda

Producto
valorUnitario

```
public class Producto  
{  
    private double valorUnitario;  
}
```

Ejemplos de posibles instrucciones de cualquier método de la clase Producto:

```
valorUnitario = 23.12;  
int valorPesos = ( int ) valorUnitario;  
valorUnitario = valorUnitario / 1.07;  
int valorPesos = 17 / 3;
```

Operadores aritméticos

- Suma +
- Resta -
- Multiplicación *
- División /
- Módulo %

Calcula el residuo de la división entre dos valores

Ejemplos del operador %

$4 \% 4$	0
$14 \% 3$	2
$17 \% 3$	2
$3 \% 17$	3
$4.5 \% 2.2$	0.1

Tipos simples de datos

- int
- double
- String
- boolean
 - Permite representar valores lógicos (verdadero o falso)
 - Solo tiene dos valores: true y false

Ejemplo tipo boolean en la Tienda

Producto
valorUnitario subsidiado

```
public class Producto
{
    private double valorUnitario;
    private boolean subsidiado;
}
```

Ejemplos de posibles instrucciones de cualquier método de la clase Producto:

```
subsidiado = true;
subsidiado = false;
boolean variableLogica = subsidiado;
```


Tipos simples de datos

- int
- double
- String
- boolean

- char
 - Sirve para representar un caracter (letra)
 - Los caracteres se escriben entre comillas simples „

Ejemplo tipo char en la Tienda

Producto
valorUnitario subsidiado calidad

```
public class Producto
{
    private double valorUnitario;
    private boolean subsidiado;
    private char calidad;
}
```

Ejemplos de posibles instrucciones de cualquier método de la clase Producto:

```
calidad = „A“;
calidad = „B“;
char variableCaracter = calidad;
```

Uso de CONSTANTES

- En los programas se pueden usar **constantes** para:
 - Definir el dominio de un atributo (posibles valores que puede tomar un atributo)
 - Representar valores inmutables (que no van a cambiar durante la ejecución del programa)

Constantes para definir el dominio de un atributo

- Se usan cuando un atributo solo puede tomar un cierto número de valores.
- Ejemplo: el tipo del producto, solo puede ser:
 - supermercado
 - papelería
 - droguería

Constantes para definir el dominio de un atributo

- Se usan cuando un atributo solo puede tomar un cierto número de valores.
- Ejemplo: el tipo del producto, solo puede ser:
 - Supermercado (1)
 - Papelería (2)
 - Droguería (3)

Ejemplo del uso de constantes en la Tienda

Producto
valorUnitario subsidiado calidad tipo

```
public class Producto
{
    // Constantes
    public final static int PAPELERIA = 1;
    public final static int SUPERMERCADO = 2;
    public final static int DROGUERIA = 3;

    // Atributos
    private double valorUnitario;
    private boolean subsidiado;
    private char calidad;
    private int tipo;
}
```

Ejemplos de instrucciones con uso de constantes

- Las siguientes instrucciones pueden ir como parte de cualquier método de la clase **Producto**:

tipo = PAPELERIA;

tipo = SUPERMERCADO;

tipo = DROGUERIA;

Ejemplos de instrucciones con uso de constantes

- El siguiente método podría pertenecer a la clase **Tienda**:

```
public void ejemplo( )  
{  
    ...  
    int tipoVenta = Producto.PAPELERIA;  
    int tipoCompra = Producto.SUPERMERCADO;  
    ...  
}
```


Constantes para representar valores inmutables

- Se usan para representar valores que no van a cambiar durante la ejecución del programa.
- Pueden ser de cualquier tipo (String, double, ...)

Ejemplo del uso de constantes en la Tienda

```
public class Producto
{
    // Constantes
    public final static int PAPELERIA = 1;
    public final static int SUPERMERCADO = 2;
    public final static int DROGUERIA = 3;

    private final static double IVA_PAPEL = 0.16;
    private final static double IVA_FARMACIA = 0.12;
    private final static double IVA_MERCADO = 0.04;

}
```

Ejemplos de instrucciones con uso de constantes

- Las siguientes instrucciones pueden ir como parte de cualquier método de la clase **Producto**:

```
double precio = valorUnitario * (1 + 0.04);
```

```
double precio = valorUnitario * (1 + IVA_MERCADO);
```