



Sección 1 – Instrucciones condicionales. (Realizar en hojas blancas).

Etapa 1: Instrucciones condicionales en pseudocódigo

Escenario:

Se tienen cuatro estudiantes, numerados del 1 al 4. A uno de ellos se le va a obsequiar un regalo. Las condiciones para recibirlo son:

1. Debe cumplir años en el segundo semestre del año.
2. Debe tener, al momento, 17 años o más.
3. De aquellos que cumplen las condiciones anteriores, se escogerá al de menor edad.
4. De no cumplirse las condiciones, no se le da regalo a ninguno

1. Describa una secuencia de pasos que se deben seguir para encontrar al estudiante afortunado.

2. Dado lo anterior, escriba una expresión que permita identificar si se debe escoger al estudiante 1.

SI(

)

Entonces

Escoger estudiante 1

Etapa 2: Expresiones condicionales – Uso de AND, OR y negaciones

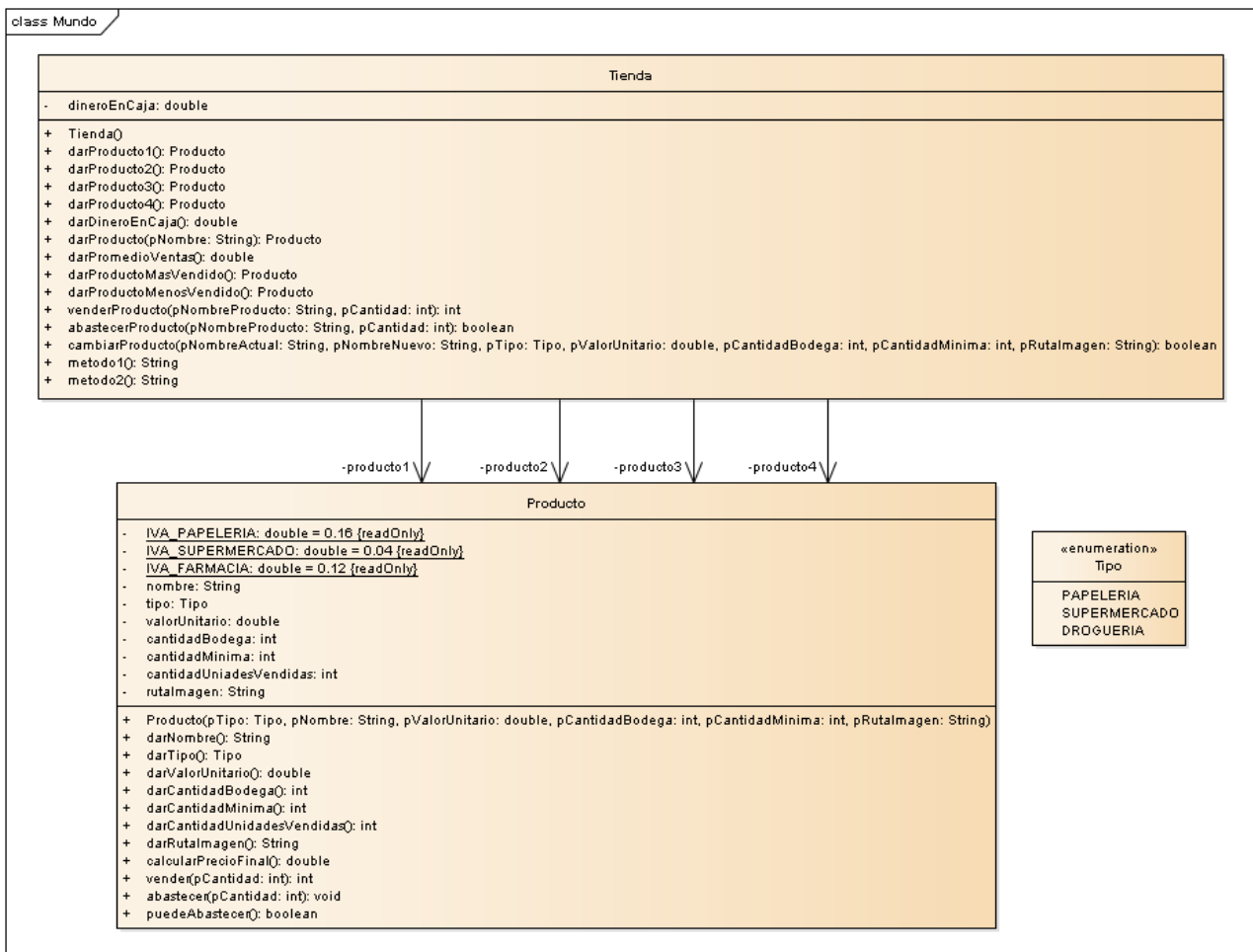
1. Complete la siguiente tabla con el resultado de la expresión

P	Q	R	(P&&Q) &&!R	(P !Q)&&R	R&&!(P&&!Q)
true	true	true			
true	true	false			
true	false	true			
true	false	false			
false	true	true			
false	true	false			
false	false	true			
false	false	False			

Etapa 3: Expresiones condicionales en Java

Escenario:

Considerando el ejercicio de la Tienda, cuyo diagrama UML del modelo del mundo se muestra a continuación, desarrolle en hojas blancas los siguientes puntos.



Métodos clase Producto

1. Aumente el valor unitario del producto utilizando la siguiente política: si el producto cuesta menos de \$1000, aumentar 1%. Si cuesta entre \$1000 y \$5000, aumentar el 2%. Si cuesta más de \$5000 aumentar el 3%. Al final retornar el nuevo valor

```
public double subirValorUnitario()
```

```
{
```

```
}
```

2. Método que indica si se puede hacer un pedido, es decir, si se reduce la **cantidadBodega**, la cantidad indicada, se mantiene la política de tener mayor o igual cantidad que la **cantidadMinima**.

```
public boolean puedeHacerPedido( int cantidad )
```

```
{
```

```
}
```

3. Modificar el precio del producto, utilizando la siguiente política: si el producto es de droguería o papelería debe disminuir un 10%. Si es de supermercado debe aumentar un 5%.

```
public void cambiarValorUnitario()
```

```
{
```

```
}
```

Métodos clase Tienda

4. Indique cuál es el **producto** que tiene el precio final más alto de la tienda.

```
public Producto darProductoMasCaro()
```

```
{
```

```
}
```

5. Utilizando el método anterior, este método **retorna** el **precio final** del producto más caro de la tienda

```
public double darPrecioProductoMasCaro()
```

```
{
```

```
}
```

6. Retorna la **cantidad** que hay en bodega del producto que se indica como **parámetro**, así:

Si llega 1, retorna la cantidad de p1.

Si llega 2, retorna la cantidad de p2.

Si llega 3, retorna la cantidad p3.

Si llega 4, retorna la cantidad p4.

```
public double darPrecioProductoMasCaro ()
{

}
}
```

Sección 2 – Expresiones condicionales. (Realizar en el ejemplo Tienda).

Para la parte práctica vamos a implementar una nueva funcionalidad que nos permita realizar pedido automático de productos. Para ello debemos:

En la clase Tienda:

1. Crear un método **public String realizarPedidoAutomatico()**

En este método se debe realizar un pedido para cada uno de los 4 productos, de acuerdo con la cantidad actual en bodega, de la siguiente forma:

Producto 1:

- Si la cantidad en bodega del producto termina en 0, pedir 4 unidades del producto.
- Si la cantidad en bodega del producto termina en 2, pedir 10 unidades del producto.
- Si la cantidad en bodega del producto termina en 7, pedir 7 unidades del producto.
- En cualquier otro caso diferente de los anteriores, pedir 8 unidades del producto.

Producto 2:

- Si la cantidad en bodega del producto termina en 5, pedir 3 unidades del producto.
- Si la cantidad en bodega del producto termina en 7, pedir 7 unidades del producto.
- Si la cantidad en bodega del producto termina en 9, pedir 1 unidades del producto.
- En cualquier otro caso diferente de los anteriores, pedir 8 unidades del producto.

Producto 3:

- Si la cantidad en bodega del producto termina en 1, pedir 5 unidades del producto.
- Si la cantidad en bodega del producto termina en 2, pedir 6 unidades del producto.

Si la cantidad en bodega del producto termina en 3, pedir 7 unidades del producto.

En cualquier otro caso diferente de los anteriores, pedir 10 unidades del producto.

Producto 4:

Si la cantidad en bodega del producto termina en 9, pedir 11 unidades del producto.

Si la cantidad en bodega del producto termina en 5, pedir 10 unidades del producto.

Si la cantidad en bodega del producto termina en 2, pedir 9 unidades del producto.

En cualquier otro caso diferente de los anteriores, pedir 3 unidades del producto.

Para implementar esto, se debe utilizar la instrucción condicional switch y se debe retornar un mensaje indicando el éxito de la operación.

En la clase InterfazTienda:

1. Modificar el método **void reqFuncOpcion1()** para llamar al método del punto anterior y mostrar la respuesta al usuario.