



**Universidad de los Andes**  
Ingeniería de Sistemas y Computación  
ISIS 1205 - Algorítmica y Programación 2  
Taller teórico nivel 12



. En la base de datos del almacén se definió la tabla producto que tiene la siguiente estructura:

```
Tabla: producto  
Campo: codigo varchar(13)  
Campo: nombre varchar(40)  
Campo: precio int  
Campo: unidades int  
PRIMARY KEY: codigo
```

1.1. ¿Cuál es la instrucción SQL que permite asignar el valor 100 al campo "unidades" de todos los registros que tengan en ese campo un valor inferior a 100?

1.2. ¿Cuál es la instrucción SQL que retorna los campos "codigo" y "nombre" de todos los productos cuyo precio sea mayor a \$100.000?

2. Complete el siguiente método de la clase `Almacen` que retorna un vector con objetos de la clase `Producto`, que contiene todos los productos de la base de datos cuyo nombre comienza por la cadena que se recibe como parámetro.

```
public ArrayList listarProductosNombre( String cadena ) throws SQLException
{
    String sql = "SELECT codigo, nombre, precio, unidades FROM producto";
    Statement st = conexion.createStatement( );
    ResultSet resultado = st.executeQuery( sql );

    //Procesa los registros

    resultado.close( );
    st.close( );

    //Retorna el vector
}
```

3. La clase `ServidorAlmacen` ofrece un servicio a `PuntoDeVenta` para contar el número de unidades existentes de un producto. El protocolo de la comunicación entre ellos es el siguiente:

	Cliente	Servidor
1	DAR_UNIDADES_PRODUCTO	
2		PRODUCTO?
3	<i>código de producto</i>	
4a		<i>número de unidades</i>
4b1		ERROR <i>mensaje de error</i>

En el paso 4 de la conversación hay dos opciones: (a) el servidor puede obtener el número de productos, en cuyo caso retorna el número o (b) el servidor obtiene algún error haciendo la consulta o recibe una línea no esperada en la comunicación. En este caso retorna ERROR y enseña el mensaje de error adecuado.

Complete los siguientes métodos de las clases `ServidorAlmacen` y `PuntoDeVenta` para que interactúen de la forma descrita anteriormente. Utilice (no desarrolle) el método `public int darUnidadesProducto (String codigo) throws Exception` que retorna el número de unidades que tiene un producto o genera una Excepción si ocurre algún problema.

### ServidorAlmacen

```

public void recibirConexion( ) throw IOException
{
    int puerto = ...
    socket = new ServerSocket( puerto );
    Socket socketCliente = socket.accept( );
    out = new PrintWriter( socketCliente.getOutputStream( ), true );
    in = new BufferedReader( new InputStreamReader( socketCliente.getInputStream( )
) );

    //Lee, procesa la petición y retorna la respuesta

    // Desconectar al cliente
    out.close( );
    in.close( );
    socketCliente.close( );
}
  
```

## PuntoDeVenta

```
public int consultarUnidadesProducto( ) throw IOException
{
    int puerto = ...
    String host = ...

    // Crea el canal al servidor con el puerto y el host dado

    // Obtiene el lector y el escritor del socket

    // Desarrolla la conversación con el servidor y procesa la respuesta

    // Desconectar al cliente
    out.close( );
    in.close( );
    socketCliente.close( );

    // Retorna la respuesta
}
```

4. A la clase `Producto` se le ha adicionado el método `public void incrementarPrecio(double porcentaje)` que incrementa el precio de un producto en el porcentaje dado como parámetro, el cual puede ser mayor o igual a cero (0).

Producto (from mundo)
<ul style="list-style-type: none"><li>String codigo</li><li>String nombre</li><li>int precio</li><li>int unidades</li></ul>
<ul style="list-style-type: none"><li>Producto(String codigo, String nombre, int precio, int unidades)</li><li>String darCodigo()</li><li>String darNombre()</li><li>int darPrecio()</li><li>int darUnidades()</li><li>String toString()</li><li>void incrementarPrecio(double porcentaje)</li></ul>

A continuación desarrolle la clase de prueba `ProductoTest`, para hacer las pruebas sólo de este método.

```
public class ProductoTest extends TestCase
{
}
}
```