



## TAREA PARA PROG10: Mantenimiento de la persistencia de los objetos. Curso 2023-24

### ENUNCIADO

- Se trata de realizar un proyecto Java en NetBeans llamado:  
**Tarea\_Prog10\_23\_24\_Apellido1Apellido2Nombre**
- Al proyecto que yo adjunto se le cambiará el nombre con el formato anterior, añadiéndole vuestros apellidos y nombre.
- Debéis **crear un usuario local a XEPDB1 llamado PROG10**, en el que crearéis las tablas .
- Crearéis en Oracle 18c XE los objetos y tablas que se adjuntan al fichero:  
**TAREA\_PROG10\_CREARTABLAS.SQL** (Crea todas las tablas y tipos)
- Crea un **TIPO\_MUEBLE**:  

```
CREATE OR REPLACE TYPE TIPO_MUEBLE AS OBJECT(  
    IDMUEBLE          NUMBER(4),  
    DESCRIPMUEBLE     VARCHAR2(50),  
    PRECIOUNITARIO    NUMBER(8,2),  
    UNIDADESEXISTENTES NUMBER(4)  
)
```
- Crea un tipo **TIPO\_VENTA**:  

```
CREATE OR REPLACE TYPE TIPO_VENTA AS OBJECT(  
    IDVENTA NUMBER(6),  
    NOMBRECLI VARCHAR2(30),  
    UNIDADES VEND NUMBER(4),  
    MUEBLE REF TIPO_MUEBLE -- Mueble que se ha vendido  
)
```
- MUEBLE: es una referencia a un objeto de TIPO\_MUEBLE.
- Crea la tabla **MUEBLES\_OBJ** que contiene registros que son objetos de tipo TIPO\_MUEBLE.  

```
CREATE TABLE MUEBLES_OBJ OF TIPO_MUEBLE(  
    IDMUEBLE PRIMARY KEY);
```
- Crea la tabla **VENTAS\_OBJ** que contiene registros que son objetos de tipo TIPO\_VENTA:  

```
CREATE TABLE VENTAS_OBJ OF TIPO_VENTAS(  
    IDVENTA PRIMARY KEY);
```
- Si necesitáis eliminarlos podéis ejecutar:  
**TAREA\_PROG10\_BORRARTABLAS.SQL** (Elimina todas las tablas y tipo)
- El proyecto tendrá 4 paquetes:
  - **controladores**: en el que realizarás una clase llamada **ConexionBD**, parecida a la que os he dejado en la plataforma en el proyecto: UT10\_Ejemplo.
  - **imagenes** (con algunas imágenes que se utilizan para los botones)
  - **modelos**: crearéis las clases para los modelos: Muebe y Venta.
  - **app**: en el que crearéis los JFrames que utilicéis. En nuestro caso,  
**AppGestionMueblesProg10Nombre**, donde Nombre será vuestro nombre.

## Formulario: AppGestionMueblesProg10Nombre

En este formulario se gestionará los datos de la tabla de objetos : **MUEBLES\_OBJ** y **VENTAS\_OBJ**.

**Hay que controlar que los datos introducidos sean correctos**, por ejemplo: los campos numéricos (txtIdVenta, txtUnidadesVend, txtIdMuebleVenta, txtIdMueble, txtPrecioUnitario, txtUnidadesExistentes) serán correctos si no deja introducir caracteres alfabéticos. Se pueden utilizar patrones.

La tabla **MUEBLES\_OBJ** será una tabla de objetos TIPO\_MUEBLE en el que este tipo tiene los siguientes campos:

- Identificador del Mueble (IDMUEBLE:NUMBER(4))
- Descripción del Mueble (DESCRIPMUEBLE: VARCHAR2(50))
- Precio Unitario (PRECIOUNITARIO: NUMBER(8,2))
- Unidades Existentes (UNIDADESEXISTENTES: NUMBER(4))

La tabla **VENTAS\_OBJ** será una tabla de objetos de TIPO\_VENTA que tiene los siguientes campos:

- Identificador de la venta (IDVENTA: NUMBER(6))
- Nombre del cliente (NOMBRE\_CLI: VARCHAR2(30))
- Unidades vendidas (UNIDADES VEND: NUMBER(4))
- Mueble: MUEBLE será una referencia a un TIPO\_MUEBLE, será el mueble al que pertenece la venta.

El diseño del formulario es:

The screenshot shows a Java Swing application window titled "TAREA PROG10-GESTIÓN DE MUEBLES". The window contains two main panels: "VENTAS MUEBLES" and "MUEBLES".

The "VENTAS MUEBLES" panel has the following components:

- Input fields for "Id Venta", "Nombre Cliente", "Unidades Vendidas", and "Id Mueble".
- A "SALIR" button (pink) with a door icon.
- Five action buttons: "Insertar" (green), "Buscar" (green), "Modificar" (green), "Eliminar" (green), and "Limpiar" (pink).

The "MUEBLES" panel has the following components:

- Input fields for "Id Mueble", "Descripción", "Precio Unitario", and "Unidades Existentes".
- A "LISTAR MUEBLE" button (green) with a document icon.
- Five action buttons: "Insertar" (green), "Buscar" (green), "Modificar" (green), "Eliminar" (green), and "Limpiar" (pink).

### VENTA MUEBLES:

Componentes del JFrame para insertar los datos de la ventas de los muebles:

- Un `JTextField` para cada uno de los atributos la venta de un mueble.
- Botones: Insertar, Buscar, Modificar, Eliminar, Limpiar

**No hay que utilizar `FocusLost`.**

## Funcionamiento de los Botones:

- **Botón Insertar:**

Si alguno de los campos está en blanco, saldrá un mensaje, por ejemplo: ALGUNO DE LOS CAMPOS ESTÁ VACÍO (mediante un JOptionPane) y volverá el foco al campo en cuestión.

Si el id de la venta ya existe en la tabla VENTAS\_OBJ, visualizaremos un mensaje (mediante un JOptionPane), no dejará insertar y volverá el foco a txtIdVenta..

Si el id de mueble(txtIdMuebleVentas) no existe en la tabla MUEBLES\_OBJ, visualizaremos el mensaje: MUEBLE NO EXISTE (mediante un JOptionPane), no dejará insertar la venta y volverá el foco a txtIdMuebleVentas.

Si las unidades introducidas son mayores que las unidades existentes en la tabla MUEBLES\_OBJ visualizaremos un mensaje 'LAS UNIDADES COMPRADAS SON MAYORES QUE LAS EXISTENTES' y no dejará insertar el registro en la tabla VENTAS\_OBJ.

Si todo está correcto insertaremos los datos en la tabla VENTAS\_OBJ y además restaremos las unidades de la venta a las unidades existente en el mueble.

Una vez insertado limpiaremos los jTextField y pondremos el foco en el id venta.

- **Botón Buscar:**

Si el id de la venta (txtIdVenta) está en blanco, saldrá un mensaje ID VENTA NO PUEDE ESTAR VACÍO (mediante un JOptionPane) volverá el foco a txtIdVenta.

Si el id de la venta (txtIdVenta ) no existe, visualizaremos un mensaje ID VENTA NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdVenta.

Si el id de la venta existe, visualizaremos en los jTextField los demás campos.

- **Botón Modificar:**

Antes de pulsar este botón deberíamos pulsar el **Botón Buscar:**

Si alguno de los campos está en blanco, saldrá un mensaje, por ejemplo: NOMBRE DE CLIENTE NO PUEDE ESTAR VACÍO (mediante un JOptionPane) y volverá el foco al campo en cuestión.

Si el id de la venta (txtIdVenta ) no existe en la tabla VENTAS\_OBJ, visualizaremos un mensaje ID VENTA NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdVenta.

Si el id de mueble(txtIdMuebleVentas) no existe en la tabla MUEBLES\_OBJ, visualizaremos el mensaje: MUEBLE NO EXISTE (mediante un JOptionPane), no dejará insertar la venta y volverá el foco a txtIdMuebleVentas.

Si las unidades introducidas son mayores que las unidades existentes en la tabla MUEBLES\_OBJ visualizaremos un mensaje 'LAS UNIDADES COMPRADAS SON MAYORES QUE LAS EXISTENTES' y no dejará actualizar el registro en la tabla VENTAS\_OBJ.

Si existe en la tabla VENTAS\_OBJ, actualizaremos los datos en la tabla VENTAS\_OBJ. limpiaremos los jTextField y pondremos el foco en el id de la venta (txtIdVenta).

- **Botón Eliminar:**

Si el id de la venta está en blanco, saldrá un mensaje ID VENTA NO PUEDE ESTAR VACÍO (mediante un JOptionPane) y volverá el foco a txtIdVenta.

Si el id de la venta (txtIdVenta ) no existe en la tabla VENTAS\_OBJ, visualizaremos un mensaje ID VENTA NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdVenta.

Preguntaremos (mediante un JOptionPane) si deseamos eliminarlo realmente. Si se responde que Si, se eliminará la venta.

Limpiaremos los jTextField y pondremos el foco en el id de la venta (txtIdVenta).

- **Botón Limpiar:**

Pone a blanco los jTextField de la venta de muebles.

- **Botón Listar Muebles:**

Se trata de visualizar por consola un listado de todas las ventas lo siguiente: id venta, nombre del cliente, descripción del artículo, unidades vendidas, , precio unitario y el total de la venta que será el resultado de multiplicar el precio unitario por las unidades vendidas.

Una vez generado el listado visualizaremos un mensaje. (mediante un JOptionPane)

TOTAL VENDIDO: es la suma de todos los totales de las ventas de todos los artículos.

ID. VENTA	NOMBRE CLIENTE	DESCRIP. MUEBLE	PRECIO	UNIDADES VEND.	TOTAL VENTA
=====	=====	=====	=====	=====	=====
XXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX	999999	99999999,99	99999999,99
XXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX	999999	99999999,99	99999999,99

.....  
TOTAL VENDIDO: XXXXXXXXXX

## **MUEBLES:**

Componentes del JFrame para introducir los datos de los muebles:

- Un JTextField para cada uno de los atributos de un mueble.
- Botones: Insertar, Buscar, Modificar, Eliminar, Limpiar

## **Funcionamiento de los Botones:**

- **Botón Insertar:**

Si alguno de los campos está en blanco, saldrá un mensaje, por ejemplo: DESCRIPCIÓN NO PUEDE ESTAR VACÍO (mediante un JOptionPane) y volverá el foco al campo en cuestión.

Si el IDMUEBLE (txtIdMueble) ya existe en la tabla MUEBLE\_OBJ, visualizaremos un mensaje MUEBLE YA EXISTE (mediante un JOptionPane), no dejará insertar y volverá el foco a txtIdMueble.

Si todo está correcto insertaremos los datos en la tabla MUEBLE\_OBJ.

Una vez insertado limpiaremos los JTextField y pondremos el foco en el txtIdMueble.

- **Botón Buscar:**

Si el IDMUEBLE (txtIdMueble) está en blanco, saldrá un mensaje ID MUEBLE ESTÁ VACÍO (mediante un JOptionPane) volverá el foco a txtIdMueble.

Si el IDMUEBLE(txtIdMueble) no existe en la tabla MUEBLE\_OBJ, visualizaremos un mensaje MUEBLE NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdMueble.

Si el IDMUEBLE (txtIdMueble ) existe en la tabla MUEBLE\_OBJ, visualizaremos en los JTextField los demás campos.

- **Botón Modificar:**

Antes de pulsar este botón deberíamos pulsar el **Botón Buscar**.

Si alguno de los campos está en blanco, saldrá un mensaje, por ejemplo: NOMBRE DE CLIENTE NO PUEDE ESTAR VACÍO (mediante un JOptionPane) y volverá el foco al campo en cuestión.

Si el IDMUEBLE (txtIdMueble) no existe en la tabla MUEBLE\_OBJ, visualizaremos un mensaje MUEBLE NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdMueble.

Si existe en la tabla MUEBLE\_OBJ, actualizaremos los datos en la tabla MUEBLE\_OBJ. limpiaremos los JTextField y pondremos el foco en el IDMUEBLE

- **Botón Eliminar:**

Si el IDMUEBLE está en blanco, saldrá un mensaje ID MUEBLE ESTÁ VACÍO (mediante un JOptionPane) y volverá el foco a txtIdMueble.

Si el IDMUEBLE (txtIdMueble) no existe en la tabla MUEBLE\_OBJ, visualizaremos un mensaje MUEBLE NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdMueble.

Si hay alguna venta de ese MUEBLE, no dejará eliminarlo.  
Preguntaremos (mediante un JOptionPane) si deseamos eliminarlo realmente. Si se responde que Si, se eliminará el mueble..  
Limpiaremos los jTextField y pondremos el foco en el IDMUEBLE.

- **Botón Limpiar:**

Pone a blanco los jTextField de los muebles.

### **Criterios de puntuación. Total 10 puntos.**

- Parte del fichero VENTAS MUEBLES: 6 puntos
- Parte del fichero MUEBLES: 4 puntos.

### **Recursos necesarios para realizar la Tarea.**

- Ordenador personal.
- JDK y JRE de Java SE.
- Oracle Database 18c XE
- Entorno de desarrollo Apache NetBeans
- Ejercicios y apuntes subidos a la unidad por el profesor en **Recursos Adicionales Aportados por el Profesor.**

### **Qué hay que entregar**

- Proyecto NetBeans comprimido funcionando y **sin errores.**  
**Tarea\_PROG10\_20\_21\_Apellido1Apellido2Nombre**
- Se creará un **informe PDF**, en el se adjuntará la ejecución del proyecto con vuestra conexión de fondo. Además, se incluirá la descripción de todos los métodos y las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea:  
**Tarea\_PROG10\_20\_21\_Apellido1Apellido2Nombre.pdf**
- Se **los dos archivos se comprimirán en uno solo:**  
**Tarea\_PROG10\_20\_21\_Apellido1Apellido2Nombre.rar**
- En caso de tener que realizar un segundo envío el nombre será:  
**Tarea\_PROG10\_20\_21\_Apellido1Apellido2Nombre\_ENVIO2.rar**