



TAREA PARA PROG11: Gestión de Base de Datos Relacionales. Curso 2023-24

ENUNCIADO

- Se trata de realizar un proyecto Java en NetBeans llamado:
Tarea_Prog11_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 PROG11**, en el que crearéis las tablas
- Crearéis en Oracle 18c XE los objetos y tablas que se adjuntan al fichero:

TAREA_PROG11_CREARTABLAS.SQL

- Crea una tabla **MUEBLE**:

```
CREATE TABLE MUEBLES(  
    IDMUEBLE          NUMBER(4) CONSTRAINT PK_MUEBLES PRIMARY KEY,  
    DESCRIPMUEBLE     VARCHAR2(50),  
    PRECIOUNITARIO    NUMBER(8,2),  
    UNIDADESEXISTENTES NUMBER(4)  
);
```
- Crea una tabla llamada **VENTAS**:

```
CREATE TABLE VENTAS(  
    IDVENTA           NUMBER(6) CONSTRAINT PK_VENTAS PRIMARY KEY,  
    NOMBRECLI         VARCHAR2(30),  
    UNIDADES VEND     NUMBER(4),  
    IDMUEBLE          NUMBER(4) REFERENCES MUEBLES(IDMUEBLE)  
);
```
- El proyecto tendrá 4 paquetes:
 - **controladores**: en el que realizarás una clase llamada **ConexionBD**.
 - **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, **AppGestionMueblesProg11_Nombre**, donde Nombre será vuestro nombre.

Formulario: AppGestionMueblesProg11Nombre

En este formulario se gestionará los datos de las tabla : **MUEBLES** y **VENTAS**.

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.

La tabla **MUEBLES** será una tabla que tiene los siguientes campos:

- Identificador del Mueble (IDMUEBLE)
- Descripción del Mueble (DESCRIPMUEBLE)
- Precio Unitario (PRECIOUNITARIO)
- Unidades Existentes (UNIDADESEXISTENTES)

La tabla **VENTAS** será una tabla que tiene los siguientes campos:

- Identificador de la venta(IDVENTA)
- Nombre del cliente (NOMBRE_CLI)
- Unidades vendidas (UNIDADESVENT)
- Id Mueble: será un idMueble que es una clave ajena a la tabla MUEBLES, es decir, que debe existir en la tabla MUEBLES.

El diseño del formulario es:

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, 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, 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 visualizaremos un mensaje 'LAS UNIDADES COMPRADAS SON MAYORES QUE LAS EXISTENTES' y no dejará insertar el registro en la tabla VENTAS.

Si todo está correcto insertaremos los datos en la tabla VENTAS.

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 en la tabla VENTAS 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 Actualizar:**

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, 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, 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 visualizaremos un mensaje 'LAS UNIDADES COMPRADAS SON MAYORES QUE LAS EXISTENTES' y no dejará actualizar el registro en la tabla VENTAS.

Si existe en la tabla VENTAS, actualizaremos los datos en la tabla VENTAS. 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, 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
XXXXXX	XXXXXXXXXXXXXX	XXXXXXXXXXXXXX	999999	99999999,99	99999999,99
XXXXXX	XXXXXXXXXXXXXX	XXXXXXXXXXXXXX	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: NOMBRE DE CLIENTE 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, 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.

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, visualizaremos un mensaje MUEBLE NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdMueble.

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

- **Botón Actualizar:**

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, visualizaremos un mensaje MUEBLE NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdMueble.

Si existe en la tabla MUEBLE, actualizaremos los datos en la tabla MUEBLE. 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, visualizaremos un mensaje MUEBLE NO EXISTE (mediante un JOptionPane) y volverá el foco a txtIdMueble.

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 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_PROG11_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_PROG11_20_21_Apellido1Apellido2Nombre.pdf
- Se **los dos archivos se comprimirán en uno solo:**
Tarea_PROG11_20_21_Apellido1Apellido2Nombre.rar
- En caso de tener que realizar un segundo envío el nombre será:
Tarea_PROG11_20_21_Apellido1Apellido2Nombre_ENVIO2.rar