

# Lenguaje de programación I



# Caso

- ✓ El administrador de la empresa Ciberfarma, necesita implementar el proceso de Boleta de venta

The diagram illustrates a 'Boleta' (Sales Slip) form with the following components and callouts:

- Busca y devuelve datos**: Callout pointing to the search icon in the 'Datos del Cliente' section.
- Datos autogenerados**: Callout pointing to the 'Núm' (Number) field.
- Mostrar una tabla con los códigos/Nombre/Cantidad/Precio/Importe**: Callout pointing to the area where a table of products would be displayed.
- Al pulsar agrega a la tabla**: Callout pointing to the shopping cart icon in the 'Datos del Producto' section.

The form itself includes the following fields and buttons:

- Datos del Cliente**: 'Cliente:' field, search icon.
- Datos del Producto**: 'Producto:' field, search icon, 'Cantidad:' field, shopping cart icon.
- Núm**: Autogenerated number field.
- Fecha**: Date field.
- Nuevo**: Button to create a new slip.
- Grabar**: Button to save the slip.
- Total**: Field for the total amount.

# Contenido

- Transacciones
- Aplicaciones

## Logros de la Unidad

- Crear aplicaciones de manejo de bases de datos



# Qué son Transacciones

- ✓ Una transacción es un evento que genera o modifica los datos que se encuentran eventualmente almacenados en un sistema de información.
- ✓ La base de un programa transaccional está en que gestiona los datos de forma que estos deben ser siempre consistente. Si durante el transcurso de una transacción ocurriese algún error, el sistema debe poder deshacer las operaciones realizadas hasta ese instante
- ✓ Ejemplos de transacciones:
  - En un sistema de reserva de pasajes de una línea aérea, tras encontrar un asiento vacío, los datos sobre la reserva de dicho asiento deben ser bloqueados hasta que la reserva se realice, de no ser así, otro operador podría tener la impresión de que dicho asiento está libre cuando en realidad está siendo reservado en ese mismo instante. Sin las debidas precauciones, en una transacción podría ocurrir una reserva doble.

# Qué son Transacciones

- Si se realiza un pago con una tarjeta electrónica, la cantidad de dinero de la cuenta sobre la que realiza el cargo debe disminuir en la misma cantidad que la cuenta que recibe el pago, de no ser así, ninguna de las dos cuentas se modificará.
- ✓ En este tipo de sistemas resulta crítico que exista un rendimiento elevado con tiempos de respuesta cortos

# Transacciones en JDBC

- ✓ Cuando se crea una conexión, cada sentencia SQL es tratada como una transacción y se compromete de forma automática después de que se ejecuta (modo auto-commit o confirmación automática).
- ✓ Sin embargo, hay momentos en los que se desea que una o más sentencias se ejecuten como una unidad, por lo que o bien todas las sentencias se ejecutan, o ninguna de las declaraciones se ejecuta.
- ✓ La manera de permitir que dos o más declaraciones agruparse en una transacción es desactivar el modo auto-commit.

```
con.setAutoCommit (false);
```

# Ejemplo



✓ Método para el registro de la tabla Boleta y tabla Detalle de Boleta

```
public int registrarBoleta(Boleta boleta, ArrayList<DetalleBoleta> detalleBoleta){
//Paso 1 : declarar las variables
Connection conn = null;
PreparedStatement pstmt1 = null;
PreparedStatement pstmt2 = null;
int resultado=-1;
try {
//Paso 2: obtener conexion a la base de datos
conn = new MySqlConnection().getConnection();
//anular el autocommit de los objetos pstmt1 y pstmt2
conn.setAutoCommit(false);
//Paso 3: realizar la sentencia sql en sql1
String sql1 =" insert into tb_Boleta "+
            "(num_bol, cod_lec, fec_emi, cod_emp,total)"+
            " values (?,?,?,?,?)";
//Paso 4: indicar la sentencia sql1 al objeto pstmt1
pstmt1 = conn.prepareStatement (sql1);
//Paso 5: establecer los parametros ( si es que los hubiera depende de la sentencia sql )
pstmt1.setInt(1,boleta.getNumBoleta());
pstmt1.setInt(2,boleta.getCodLector());
pstmt1.setString(3,boleta.getFechaEmision());
pstmt1.setInt(4,boleta.getCodEmpleado());
pstmt1.setDouble(5,boleta.getTotal());
//Paso 6: ejecutar sentencia sql
resultado=pstmt1.executeUpdate();
}
```



# Ejemplo



## ✓ Método para el registro de la tabla Boleta y tabla Detalle de Boleta

```
//del paso 3 al paso 6 es para registrar en la tabla boleta
//del paso 7 al paso 11 es para registrar en la tabla detalle_boleta
//Paso 7: realizar la sentencia sql en sql2
String sql2=" insert into tb_detalle_boleta "+
            " (num_bol,cod_concepto,pre_concepto,can,importe ) " +
            " values (?, ?, ?, ?, ?) ";

//Paso 8 :recorrer los items del ArrayList detalleBoleta
for (DetalleBoleta d : detalleBoleta) {
    //Paso 9: indicar la sentencia sql2 al objeto pstmt2
    pstmt2 = conn.prepareStatement (sql2);
    //Paso 10: establecer los parametros
    pstmt2.setInt(1,d.getNumBoleta());
    pstmt2.setInt(2,d.getCodConcepto());
    pstmt2.setDouble(3,d.getPreConcepto());
    pstmt2.setInt(4,d.getCantidad());
    pstmt2.setDouble(5,d.getImporte());
    //Paso 11: ejecutar sentencia sql2
    resultado=pstmt2.executeUpdate();
}
//Paso 12: Si todo es correcto se confirma la transacción
conn.commit();
} catch (Exception ex) {
    try {
        conn.rollback();
        System.out.print("Ocurrio un evento inesperado método insertBoleta: "+ex.toString());
        resultado=-1;
    } catch (SQLException e) {
```



# Ejemplo



✓ Método para el registro de la tabla Boleta y tabla Detalle de Boleta

```
    }  
} finally {  
    try { //Paso 13 : cerrar los objetos  
        if (pstmt1 != null) pstmt1.close();  
        if (pstmt2 != null) pstmt2.close();  
        if (conn != null) conn.close();  
    } catch (Exception ex2) {  
        System.out.println(ex2.getMessage());  
    }  
}  
//Paso 14 : Se retorna el resultado => 1 registro exitoso, -1 =ocurrio un error  
return resultado;  
}
```

# Ejemplo de Autogenerado en la Boleta

## ✓ Ejemplo de Autogenerado en la Boleta

```
@Override
public String generaNumBoleta() {
    ArrayList<Boleta> lista = listado();
    int codigo = 1;
    DecimalFormat df = new DecimalFormat("0000");
    if (lista != null){
        Boleta x = lista.get(lista.size()-1);
        codigo = Integer.parseInt(x.getNumvta().substring(1))+1;
    }
    return "B" + df.format(codigo);
}
```

@Override

# Referencia

✓ <http://www.jc-mouse.net/java/transacciones-uso-de-commit-y-rollback>

✓ <http://blog.rolandopalermo.com/2012/10/transactions-jdbc.html>

# GRACIAS



## **SEDE MIRAFLORES**

Calle Díez Canseco Cdra 2 / Pasaje Tello  
Miraflores – Lima  
Teléfono: 633-5555

## **SEDE INDEPENDENCIA**

Av. Carlos Izaguirre 233  
Independencia – Lima  
Teléfono: 633-5555

## **SEDE BREÑA**

Av. Brasil 714 – 792  
(CC La Rambla – Piso 3)  
Breña – Lima  
Teléfono: 633-5555

## **SEDE TRUJILLO**

Calle Borgoño 361  
Trujillo  
Teléfono: (044) 60-2000

## **SEDE SAN JUAN DE LURIGANCHO**

Av. Próceres de la Independencia 3023-3043  
San Juan de Lurigancho – Lima  
Teléfono: 633-5555

## **SEDE SAN MIGUEL**

Av. Federico Gallese 847  
San Miguel – Lima  
Teléfono: 632-4900

## **SEDE BELLAVISTA**

Av. Mariscal Oscar R. Benvides 3866 – 4070  
(CC Mall Aventura Plaza)  
Bellavista – Callao  
Teléfono: 633-5555

## **SEDE AREQUIPA**

Av. Porongoche 500  
(CC Mall Aventura Plaza)  
Paucarpata - Arequipa  
Teléfono: (054) 60-3535