

## Exercicio 1

Crea as táboas PRODUCTOS, CLIENTES E VENTAS nas bases de datos MySQL e SQLite. En MySQL crea unha base de datos de nome unidade2. En SQLite garda as táboas nun ficheiro que se chame unidade2.db.

As táboas son as seguintes:

### PRODUCTOS:

ID numérico, clave primaria  
DESCRIPCION varchar(50), non nulo  
STOCKACTUAL numérico  
STOCKMINIMO numérico  
PVP numérico

### CLIENTES:

ID numérico, clave primaria  
NOME varchar(50), non nulo  
ENDEREZO varchar(50)  
POBOACION varchar(50)  
TELEFONO varchar(20)  
NIF varchar(10), no nulo, único

### VENTAS:

IDVENTA numérico, clave primaria  
DATAVENTA data, non nulo  
IDCLIENTE numérico, clave allea a CLIENTES  
IDPRODUCTO numérico, clave allea a PRODUCTOS  
CANTIDADE numérico  
Un cliente pode ter varias ventas

Unha vez creadas as táboas nas bases de datos fai un programa Java que encha as táboas PRODUCTOS e CLIENTES (os datos a insertar defínense no propio programa). O programa Java recibe un argumento ao executalo dende a liña de comandos cuxo valor é 1 ou 2. Se o valor é 1 debes encher as táboas da base de datos de MySQL e se é 2 debes enchelas na base de datos de SQLite.

## Exercicio 2

Partindo das táboas anteriores, realiza un programa Java para insertar ventas na táboa VENTAS. O programa recibe varios parámetros dende a liña de comandos:

1. O primeiro parámetro indica a base de datos onde se insertará a venta (1 ou 2) e o seu significado é como no exercicio anterior.
2. O segundo parámetro indica o identificador da venta.
3. O terceiro parámetro indica o identificador do cliente.
4. O cuarto parámetro indica o identificador do produto.
5. O quinto parámetro indica a cantidade.

Realiza as seguintes comprobacións antes de insertar a venta na táboa:

- O identificador de venta non debe existir na táboa VENTAS.
- O identificador de cliente debe existir na táboa CLIENTES.
- O identificador de produto debe existir na táboa PRODUCTOS.
- A cantidade debe ser maior que 0.
- A data da venta é a data actual.

Unha vez insertada a fila na táboa visualizar unha mensaxe indicándoo. Se non se poido realizar a inserción visualizar o motivo (non existe o cliente, non existe o produto, cantidade menor ou igual a 0, ...).

Executa o programa e inserta varias vendas nas distintas bases de datos.

### Exercicio 3

Preténdese realizar un listado das vendas dun cliente. O programa recibe dous argumentos dende a liña de comandos, o primeiro indica a base de datos da que se consultarán as vendas e o segundo o identificador do cliente cuxas vendas vanse consultar. O programa debe visualizar a seguinte información:

```
Ventas do cliente: nome do cliente
Venta: idventa, Data venta: data
        Produto: descripcón do produto
        Cantidade: cantidade   PVP: pvp
        Importe: cantidade*pvp
Venta: idventa, Data venta: data
        Produto: descripcón do produto
        Cantidade: cantidade   PVP: pvp
        Importe: cantidade*pvp
...
Número total de vendas: ____
Importe total: ____
```

### Exercicio 4

Crea un procedemento almacenado na base de datos ao que se lle pasen como parámetros de entrada o identificador do produto e a cantidade dun produto nunha venta, e actualice o STOCKACTUAL na táboa PRODUCTOS.

Modifica o código do exercicio 2 para que cando se realice unha venta se actualice o stock.

Crea unha función que devolva o STOCKACTUAL dun produto cuxo identificador pasarase como parámetro a dito procedemento.

Modifica o código do exercicio 2 para que se realice a comprobación de que existan as unidades dispoñibles en stock para que a venta sexa realizada con éxito.

Todas as accións que compoñen unha venta funcionarán como unha unidade de execución e deben ser tratadas como unha transacción.

### Exercicio 5

Preténdese realizar a consulta dos datos dun produto. Pedirase ao usuario o identificador do produto e amosaranse os seus datos. Se o stock actual está por debaixo do stock mínimo emitirase unha mensaxe indicándoo.