

DIAGRAMAS DE SECUENCIA

EJERCICIOS DE REFUERZO

1.- Dibujar el diagrama de secuencia para el siguiente caso de uso:

- Identificador: RealizaPedido.
- Actor que lo inicia: Cliente.
- Precondiciones: Un cliente registrado en el sistema ha accedido correctamente al mismo.
- Secuencia de eventos de flujo:
 1. El cliente introduce su nombre y dirección.
 2. El cliente introduce el código del producto que desea incluir en el pedido.
 3. El sistema aporta la descripción y el precio del producto.
 4. El sistema almacena temporalmente el pedido.
 5. El cliente introduce la información de la tarjeta de pago.
 6. El cliente pulsa el control "Ejecuta".
 7. El sistema requiere confirmación del banco.
 8. El pago es confirmado, se acepta el pedido, se almacena en la base de datos, y se le asigna un ID que se retorna al cliente.
- Postcondiciones: Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente.

2.- Dibujar el diagrama de secuencia para el siguiente caso de uso:

- Identificador: RealizaPedido.
- Actor que lo inicia: Cliente.
- Precondiciones: Un cliente registrado en el sistema ha accedido correctamente al mismo.
- Secuencia de eventos de flujo:
 1. El cliente introduce su nombre y dirección.
 2. Si el cliente introduce el ZIP, el sistema introduce la ciudad y región.
 3. El cliente introduce los códigos de los productos que desea incluir en el pedido.
 4. Por cada código de producto introducido se hace lo siguiente:
 - a. El sistema aporta la descripción y el precio del producto.
 - b. El sistema suma el precio del producto en el total del pedido.
 5. El sistema almacena temporalmente la lista de productos incluidos en el pedido.
 6. El cliente introduce la información de la tarjeta de pago.
 7. El cliente pulsa el control "Ejecuta".
 8. El sistema requiere confirmación del banco. Si la información es incorrecta, el sistema requiere su corrección al cliente.
 9. Cuando el pago es confirmado, se acepta el pedido, se almacena y se le asigna un ID que se retorna al cliente.
- Postcondiciones: Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente.

3.- Dibujar el diagrama de secuencia para el método *doPaint()* en los distintos casos descritos a continuación:

a.-

```
protected void doPaint(){  
    Compas unCompas = new Compas();  
    Circulo unCirculo = unCompas.dibujarCirculo(5);  
    Pincel unPincel = Cartuchera.getPinceles().getItem(2);  
    unPincel.setColor(new Color("#336677"));  
    unPincel.pintar(unCirculo);  
    int superficie = unCirculo.getSuperficie();  
}
```

b.-

```
protected void doPaint(Painter painter, Config config) {  
    painter.drawRect(x, y, width, height);  
    // Cause painting of shapes to be relative to this shape  
    painter.translate(x, y);  
    for (Shape s : shapes) {  
        Rectangle clip = s.getClip();  
        painter.setClip(clip);  
        s.paint(painter);  
    }  
    // Restore graphics origin  
    painter.translate(-x, -y);  
}
```

c.-

```
protected void doPaint(Painter painter, Config config) {  
    painter.drawRect(x, y, width, height);  
    // Cause painting of shapes to be relative to this shape  
    boolean translate = config.needsTranslation();  
    if (translate) {  
        painter.setTransformsEnabled(true);  
        painter.translate(x, y);  
    }  
    for (Shape s : shapes) {  
        s.paint(painter);  
    }  
}
```

d.-

```
protected void doPaint(Painter painter, Config config) {  
    painter.drawRect(x, y, width, height);  
    // Cause painting of shapes to be relative to this shape  
    boolean translate = config.needsTranslation();  
    if (translate) {  
        painter.setTransformsEnabled(true);  
        painter.translate(x, y);  
    } else {  
        painter.setTransformsEnabled(false);  
        painter.translate(0, 0);  
    }  
    for (Shape s : shapes) {  
        s.paint(painter);  
    }  
}
```

4.- Dibujar el diagrama de secuencia para el método ejecutarConsulta() descrito a continuación:

```
private void ejecutarConsulta(){  
    Connection connection = new Connection("localhost/mydb");  
    Command command = new Command();  
    command.setConnection(connection);  
    command.setText("SELECT * FROM USERS");  
    connection.open();  
    DataReader reader = command.execute();  
    connection.close();  
    while(int i < reader.getResultSize())  
    {  
        Printer.write(reader.getRow(i));  
        i++;  
    }  
    connection.close();  
}
```

5.- Realizar el diagrama de secuencia del caso de uso ACTUALIZAR CATÁLOGO descrito a continuación:

- La aplicación mostrará una pantalla en la que se puede ver la lista del catálogo actual. Desde esa pantalla se podrá seleccionar la operación a realizar: añadir, borrar o modificar.
- Si se pulsa añadir se mostrará la ventana de inserción de datos, para teclear la descripción, el precio, el código, y también el proveedor. El proveedor se puede elegir de una lista o se puede añadir un nuevo proveedor. En este caso se mostrará una nueva ventana para teclear sus datos y añadirlo a la base de datos. Finalmente el producto se añadirá a la base de datos.
- Si se ha elegido borrar el producto, se selecciona del catálogo, se consulta si tiene pedidos y si no está en ningún pedido se borra del catálogo.
- Si se ha elegido modificar el producto, se mostrará la ventana con los datos del producto para modificarlos, se teclean los datos y se modifica.

Considerar las siguientes líneas de vida: Jefe de Marketing, Ventana, Producto y Proveedor.