

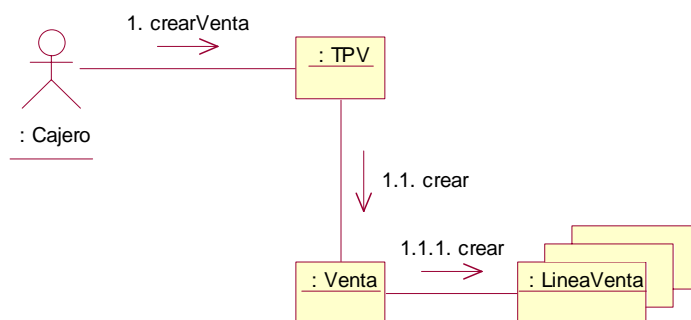
Ejercicios

Diagramas de interacción

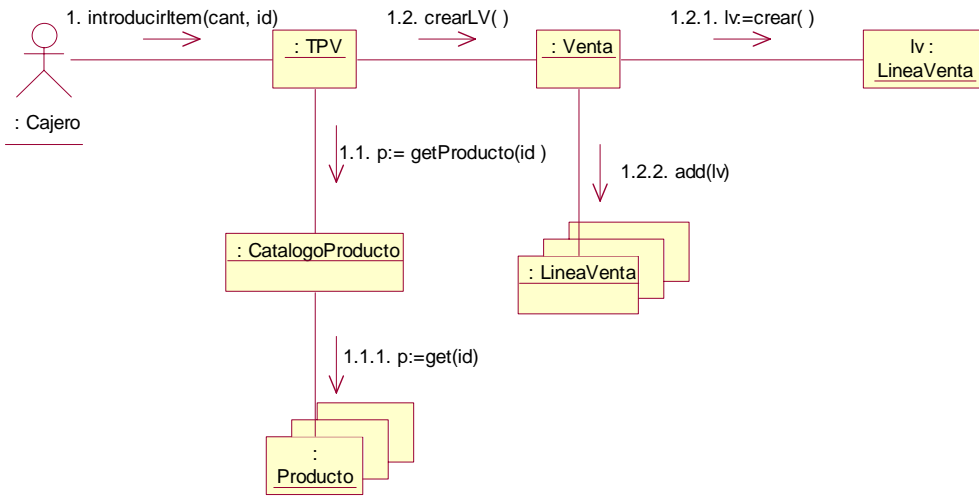
Ejercicio 1.

Dados los siguientes diagramas de interacción, correspondientes a la gestión de un TPV, obtener el modelo del dominio mediante un diagrama de clases con toda la información posible (atributos y métodos en las clases, así como las relaciones entre dichas clases).

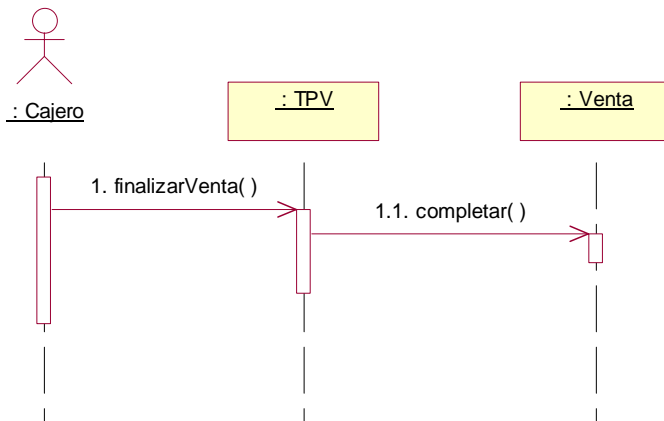
1. CrearVenta



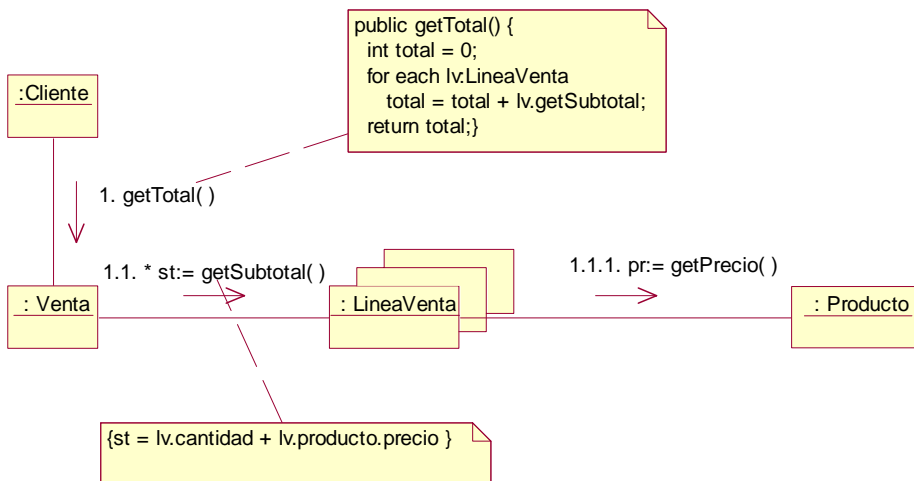
2. IntroducirItem



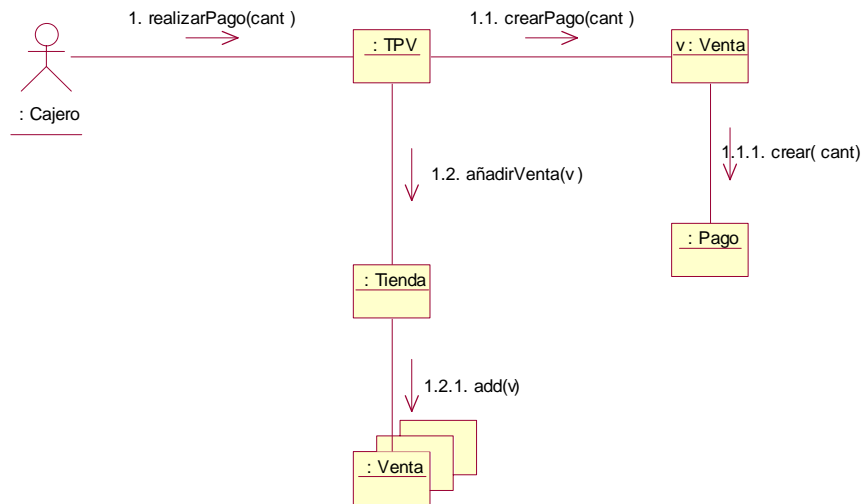
3. FinalizarVenta



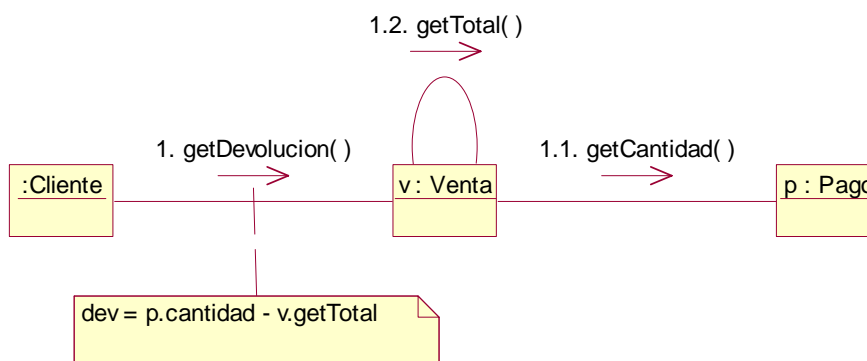
4. CalcularTotal



5. RealizarPago



6. CalcularDevolución



Ejercicio 2.

Dado el código Java de las clases, correspondientes a la gestión de un TPV, obtener el diagrama de clases con toda la información posible (atributos y métodos en las clases, así como las relaciones entre dichas clases). Además, se debe comprobar que coincide con el diagrama de clases anterior. Si no es así, entonces se tiene que corregir el código Java de estas clases de acuerdo con el diagrama de clases anterior.

```

public class Pago {
    private Dinero cantEntregada;

    public Pago (Dinero cantidad) { cantEntregada = cantidad; }
  
```

```
        public Dinero getCantEntregada () { return cantEntregada; }
    }

public class CatalogoProducto {
    private Map productos = new HashMap ();

    public CatalogoProductos () {
        ItemId id1 = new ItemID (100);
        ItemId id2 = new ItemID (200);
        Dinero precio1 = new Dinero (3);
        Dinero precio2 = new Dinero (5);
        Producto p;
        p:= new Producto (id1, precio1, "producto 1");
        productos.put (id1, p);
        p = new Producto (id2, precio2, "producto 2");
        productos.put (id2, p);
    }

    public Producto getProducto (ItemId id) {
        return (Producto) productos.get(id);
    }
}

public class TPV {
    private CatalogoProducto catalogo;
    private Venta venta;

    public TPV(CatalogoProducto cp) { catalogo = cp; }

    public void crearNuevaVenta () {venta = new Venta();}

    public void finalizarVenta () { venta.completar(); }

    public void introducirItem (ItemId id, int cant) {
        Producto p = catalogo.getProducto (id);
        Venta.crearLineaVenta (p, cant);
    }

    public void realizarPago() { venta.crearPago(cant); }
}

public class Producto {
    private itemID id;
    private Dinero precio;
    private String descripcion;

    public Producto (ItemID id, Dinero precio, String desc) {
        this.id = id;
        this.precio = precio;
        this.descripcion = desc;}

    public ItemId getId() { return id; }

    public Dinero getPrecio() { return precio; }

    public String getDescripcion() { return descripcion; }
}

public class Venta {
    private List lineaVentas = new ArrayList();
    private Date fecha = new Date();
    private boolean esCompleta;
    private Pago pago;
```

```

    public Dinero getDevolucion() {
        return pago.getCantEntregada().minus(getTotal() );
    }

    public void completar() { esCompleta = true; }

    public void crearLineaVenta(Producto p, int cant) {
        lineaVentas.add(new LineaVenta(p,cant));
    }

    public Dinero getTotal() {
        Dinero total = new Dinero();
        Iterator i = lineaVentas.iterator();
        while (i.hasNext()) {
            LineaVenta lv = (LineaVenta) i.next();
            total.add(lv.getSubtotal());
        }
        return total;
    }

    public void crearPago (Dinero cantEntregada) {
        pago = new Pago(cantEntregada);
    }
}

public class LineaVenta {
    private int cantidad;
    private Producto producto;

    public LineaVenta(Producto p, int cant) {
        this.producto = p;
        this.cantidad = cant; }

    public Dinero getSubtotal () {
        return producto.getPrecio().times(cantidad);
    }
}

public class Tienda {
    private CatalogoProducto catalogo;
    private TPV tpv;

    public TPV getTPV {return TPV; }
}

```

Ejercicio 3.

Construir los diagramas de interacción para modelar los siguientes comportamientos dinámicos de la operación “crearLaberinto”:

a. Versión 1

```

public class JuegoLaberinto {
    public Laberinto crearLaberinto () {
        Laberinto lab = new Laberinto();
        Habitacion h1 = new Habitacion();
        Habitacion h2 = new Habitacion();
        Puerta puerta = new Puerta(h1, h2);
    }
}

```

```
        lab.añadeHabitacion(h1);
        lab.añadeHabitacion(h2);
        h1.añadePuerta(puerta);
        return lab;
    }
}
```

b. Versión 2

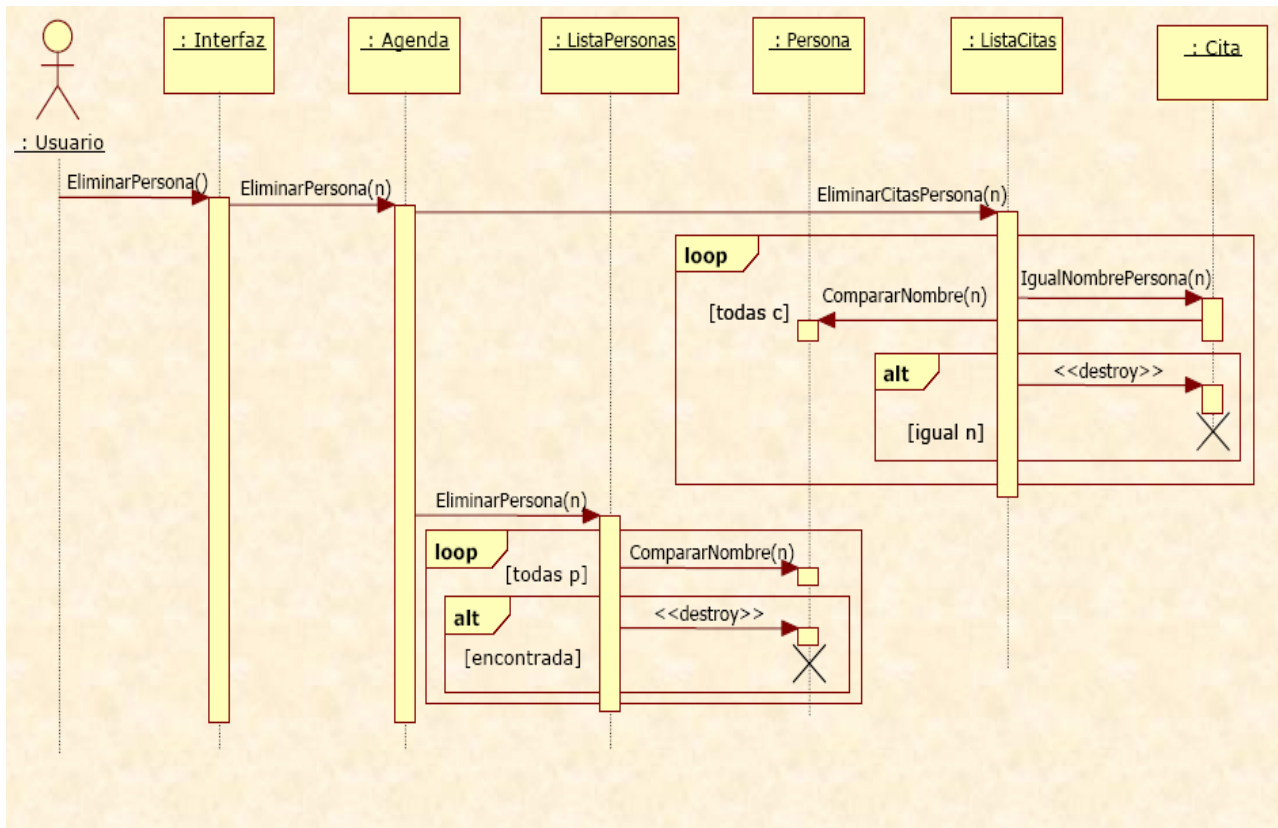
```
public class JuegoLaberinto {
    private Laberinto lab;
    private boolean conVentana;

    public JuegoLaberinto() {
        lab = new Laberinto();
        conVentana = true;
    }

    public void crearLaberinto () {
        Habitacion h;
        for (int i=0; i<10; i++) {
            h = new Habitacion();
            if (conVentana == true)
                h.añadeVentana(new Ventana());
            lab.añadeHabitacion(h);
        }
    }
}
```

Ejercicio 4.

Dado el siguiente diagrama de secuencia, correspondiente al escenario del caso de uso para eliminar una persona de una agenda, representar el diagrama de clases parcial en UML con toda la información posible (atributos y métodos en las clases, y relaciones entre dichas clases).



Ejercicio 5.

Un sistema bancario está formado por cuentas. Una cuenta puede pertenecer a una persona o a una empresa. Una empresa tiene contratados a un conjunto de personas. Una persona puede estar contratada por varias empresas. El contrato de una empresa a una persona tiene un número de contrato y una fecha. Por otra parte, se cobran comisiones bancarias de mantenimiento a las cuentas. Para una cuenta de empresa, la comisión es de 9 euros, mientras que para la comisión de una cuenta personal, se considera el saldo de la cuenta y si está por debajo de un cierto valor (100 euros), la comisión es de 20€ y en caso contrario de 15€

Se pide:

- Representar la parte estática mediante un diagrama de clases detallado.
- Utilizando los métodos y clases del diagrama de clases del apartado a), construir los diagramas de secuencia necesarios para el siguiente escenario: “Existen 1 empresa (e) y 2 personas (p1, p2). Se abre una cuenta para la empresa (c3) y una cuenta para cada persona (c1, c2). La empresa “e” contrata a las personas “p1” y “p2”. La empresa “e” recibe una subvención de 10.000€, de los cuales paga 2.000€ a “p1” y 3.000€ a “p2”. A continuación, el banco cobra la comisión de mantenimiento a la empresa “e” y a las personas “p1” y “p2”.