

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package adquirirentrada.TipoAdquisicion;

import adquirirentrada.Adaptador.AdaptadorTarjeta;
import adquirirentrada.DTO.DTODatosCompra;
import adquirirentrada.DTO.DTOPago;
import adquirirentrada.Entrada.CuentaBancaria;
import adquirirentrada.Entrada.Entrada;
import adquirirentrada.Entrada.Usuario;
import adquirirentrada.Excepciones.NoSePuedeAsignarLaEntradaException;
import adquirirentrada.Excepciones.NoSePuedeValidarPagoException;
import adquirirentrada.Repositorios.RepoCuentaBancaria;
import adquirirentrada.Repositorios.RepoEntrada;
import adquirirentrada.Repositorios.RepoTarifa;
import java.util.List;

/**
 *
 * @author nicolas
 */
public abstract class TipoAdquisicion {

    private String nombreTipoAdquisicion;
    private RepoTarifa repoTarifa;
    private RepoCuentaBancaria repoCuentaBancaria;
    protected RepoEntrada repoEntrada;

    public TipoAdquisicion(){
        this.repoTarifa= new RepoTarifa();
        this.repoCuentaBancaria= new RepoCuentaBancaria();
        this.repoEntrada=new RepoEntrada();
    }

    protected abstract int getMonto(DTODatosCompra datosCompra);

    protected DTOPago generarDatosPago(int monto, DTODatosCompra datosCompra, CuentaBancaria cuentaBancariaDestino){

        String cuentaBancaria=cuentaBancariaDestino.getCodigoCuenta();
        String nombreBanco=cuentaBancariaDestino.getNombreBanco();
        String nroTarjetaComprador=datosCompra.getDatosTarjeta().getNumeroTarjeta();
        String claveTarjetaComprador=datosCompra.getDatosTarjeta().getClaveTarjeta();
        return new DTOPago(cuentaBancaria,nombreBanco,nroTarjetaComprador,claveTarjetaComprador,monto);
    }

    protected int getMontoEventBrite(int montoTotal){
        int porcentajeIVA=this.repoTarifa.getTarifaByName("IVA").getPorcentaje();
        int porcentajeEB= this.repoTarifa.getTarifaByName("EventBrite").getPorcentaje();

        int montoEventBrite= montoTotal * porcentajeEB;
        montoEventBrite=montoEventBrite + montoEventBrite * porcentajeIVA;
        return montoEventBrite;
    }
}
```

```
public void adquirirEntrada(CuentaBancaria cuentaBancariaDestino, Usuario comprador, AdaptadorTarjeta adaptadorTarjeta, List<Entrada> entradas, DTODatosCompra datosCompra){

    try{
        entradas.forEach(entrada->entrada.comprar());
        this.repoEntrada.updateEntradas(entradas);
    }
    catch(Exception e){
        throw new NoSePuedeAsignarLaEntradaException("No se pudieron reservar todas las entradas requeridas por favor intentelo mas tarde");
    }

    int precioXEntrada=this.getMonto(datosCompra);
    int montoTotal= precioXEntrada * entradas.size();

    int montoEventBrite=this.getMontoEventBrite(montoTotal);
    int montoOrganizador= montoTotal - montoEventBrite;

    CuentaBancaria cuentaBancariaEventBrite= this.repoCuentaBancaria.getCuentaBancariaEventBrite();

    DTOPago pagoEventBrite= this.generarDatosPago(montoEventBrite,datosCompra,cuentaBancariaEventBrite);
    DTOPago pagoOrganizador= this.generarDatosPago(montoOrganizador,datosCompra,cuentaBancariaDestino);

    try{
        adaptadorTarjeta.validarPago(pagoEventBrite);
        adaptadorTarjeta.validarPago(pagoOrganizador);
        adaptadorTarjeta.realizarPago(pagoEventBrite);
        adaptadorTarjeta.realizarPago(pagoOrganizador);

        entradas.forEach(entrada->entrada.setDuegno(comprador));
        this.repoEntrada.updateEntradas(entradas);
    }catch(Exception e){
        entradas.forEach(entrada->entrada.ponerEnDisponible());
        this.repoEntrada.updateEntradas(entradas);
        throw new NoSePuedeValidarPagoException("Hubo un error en la validacion del pago");
    }

}
```