



Trabajo Práctico

Asignatura: Programación 3

Año: 2023

Lugar: Facultad de Ingeniería

Integrantes:

Basualdo, Juan Ignacio

Juares, Alan

Trinitario, Bruno

Olave, Juan Ignacio



Informe

Introducción:

En este trabajo se pretende desarrollar un sistema de facturación empresarial para trabajar con contrataciones de servicio de monitoreo y seguridad, con clientes.

Soluciones elegidas:

La clase empresa es la encargada de agregar sus respectivos abonados al sistema o quitarlos en su defecto, crea/remueve sus facturas. También crea sus contrataciones y trabaja con los domicilios ya sea designando a su respectivo abonado o quitándole al mismo.

Se crean domicilios, abonados, servicios, promos y contrataciones. A la contratación se le asigna un domicilio e implementa la lista de servicios con su respectiva promoción, en caso de tenerlas.

A cada abonado se le asignan domicilios y sus correspondientes contrataciones.

Encapsulamos el abonado según su respectivo tipo de pago, a partir de la instancia obtenida en la clase factory y se lo envía a factura, la cual cuenta con las siguientes funcionalidades: imprimir un factura, clonar una factura.

Las excepciones con las que tratamos son:

- `ContratacionInvalidaException`: esta excepción se realiza cuando no se puede crear la contratación debido a la asignación de un domicilio ya vinculado.



- `DomicilioExistenteException`: es lanzada cuando el domicilio ya se encuentra agregado a la lista de domicilios del abonado.
- `DomicilioInexistenteException`: es creada en la situación en la que se solicita eliminar un domicilio de la lista de domicilios del abonado pero este no existe.
- `AbonadoInexistenteException`: es lanzada cuando se quiere eliminar un determinado abonado de la lista que contiene abonados y este no existe en ella.
- `FacturaInexistenteException`: es creada cuando se quiere eliminar una factura de la lista de facturas y no está en la lista.
- `FactoryInvalidoException`: esta excepción se lanzará cuando no existe el tipo String de pago para realizar la instancia del objeto.

Las primeras 5 excepciones planteadas las desarrollamos debido a que existe la posibilidad de que el domicilio que ingrese sea incorrecto o no exista. Y al encontrarse en una lista, la cual puede contar con varios domicilios no es meramente responsabilidad de quien invoque el programa tener conocimiento de cada uno de sus domicilios (lo mismo aplica para abonado y factura).

La última excepción se realiza ya que puede mandarse un tipo String de pago inválido y al existir la posibilidad de que la clase Factory se extienda debido a la implementación de nuevos métodos de pagos aumenta la probabilidad de error.

Las dificultades que se nos presentaron son:

- Le dábamos responsabilidades al domicilio de conocer su abonado, con la sumatoria del costo total del servicio con sus agregados y promos. Solventamos este problema



delegando la responsabilidad a la clase Contratación de informar estos costos previamente mencionados.

- Mala implementación del doble dispatch generado por el problema anteriormente mencionado. Aplicamos double dispatch en la clase 'domicilio'.
- No tuvimos en cuenta que el software desarrollado debe poseer cierto grado de escalabilidad
- Poder adecuar los patrones de diseño al código sin modificar el funcionamiento de este.

