

Luciano Nicolas Spinazzola DNI 41778502

Firma:

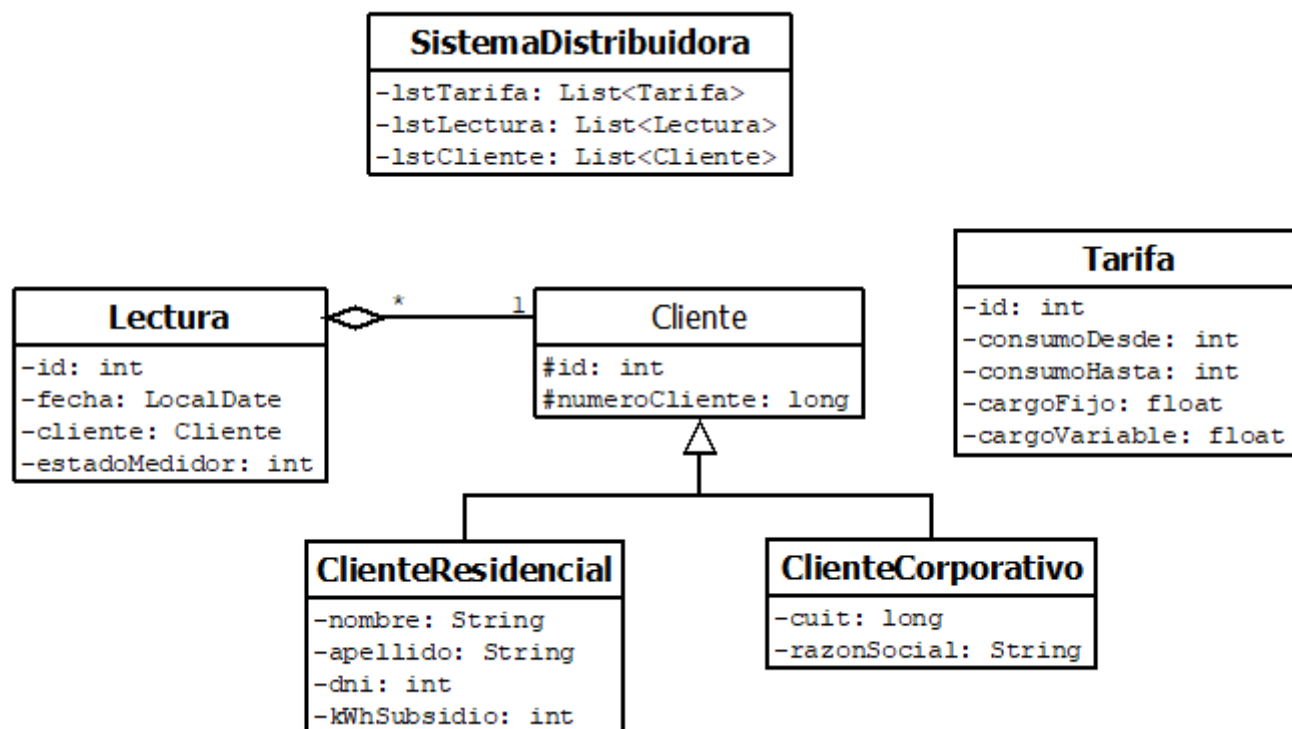
Licenciatura en Sistemas
Orientación a Objetos 1

IDE: Eclipse

Presentación del proyecto:

Crear una carpeta llamada **tuApellidoNombre** y dentro crear el proyecto del examen con **tuNroDni** como nombre.Entrega: **por Aula Virtual**Sistema Distribuidora Eléctrica

Se nos pide poner en marcha un sistema para gestionar el proceso de lectura de medidores, calcular el total a pagar según el tipo de cliente. La empresa tiene instalado un medidor de corriente eléctrica en el domicilio de cada cliente y todos los meses un empleado de la misma (el inspector) pasa tomando la lectura del mismo. El inspector toma nota de la cantidad de kWh que indica (el estado), a qué cliente corresponde y la fecha en la que se realizó la medición.



Cliente (Abstracta) # int id # long numeroCliente	ClienteCorporativo - long cuit - String razonSocial	ClienteResidencial - String nombre - String apellido - int dni - int kWhSubsidio
Tarifa - int id - int consumoDesde - int consumoHasta - float cargoFijo //es monto fijo (\$) - float cargoVariable //es \$ por KWh	Lectura - int id - LocalDate fecha - Cliente cliente - int estadoMedidor	SistemaDistribuidora - List<Tarifa> lstTarifa - List<Lectura> lstLectura - List<Cliente> lstCliente

Casos de Uso

Nota: Ningún CU implementa excepciones.

- 1) + **agregarCliente(long numeroCliente, String nombre, String apellido, int dni, int kWSubsidio) : boolean**
El id se calcula de forma autoincremental, teniendo en cuenta que la lista puede tener altas y bajas de objetos.
- 2) + **agregarCliente(long numeroCliente, long cuit, String razonSocial) : boolean**
El id se calcula de forma autoincremental, teniendo en cuenta que la lista puede tener altas y bajas de objetos.
- 3) + **agregarTarifa(int consumoDesde, int consumoHasta, float cargoFijo, float cargoVariable) : boolean**
El id se calcula de forma autoincremental, teniendo en cuenta que la lista puede tener altas y bajas de objetos.
- 4) + **traerCliente(long numeroCliente) : Cliente**
- 5) + **agregarLectura(LocalDate fecha, Cliente cliente, int estadoMedidor) : boolean**
El id se calcula de forma autoincremental, teniendo en cuenta que la lista puede tener altas y bajas de objetos.
- 6) + **traerLectura(int mes, int anio, Cliente cliente) : Lectura**
Equals de objeto Cliente es true si numeroCliente son iguales
- 7) + **calcularConsumo(Lectura lecturaAnterior) : int**
Retorna el consumo, diferencia entre estadoMedidor de las dos lecturas
- 8) + **traerTarifa(int consumo) : Tarifa**
Retorna objeto Tarifa, que cumpla ((consumoDesde <= consumo) y (consumo <= int consumoHasta))
- 9) + **calcularTotalACobrar(int mes, int anio, Cliente cliente) : float**
 - 1° Recupera el objeto Lectura según parámetros (int mes, int anio, Cliente cliente)
 - 2° Recupera el objeto Lectura del mes anterior (1°). //utilizar minusMonths(1)
 - 3° Calcular el consumo del cliente entre las lecturas
 - 4° Determinar la tarifa según el consumo.
 - 5° $totalVariable = cargoVariable * consumo$
 - 6° Si el cliente es **ClienteResidencial** en el caso de tener subsidio aplica descuento= $cargoVariable * kWhSubsidio$
Si totalVariable >= descuento entonces totalVariable=totalVariable - descuento en caso contrario totalVariable=0.
 - 7° Retorna la suma cargoFijo + totalVariable

TestOO1Tema1.java

Nota: Al comenzar cada test indique el número a resolver ej: System.out.println("Test 1"); y luego la implementación.

Test1) Agregar e imprimir a todos los clientes

ClienteResidencial [numeroCliente=4483979, apellido=Perez, nombre=Juan, dni=11111111, kWSubsidio=500]

ClienteResidencial [numeroCliente=4971920, apellido=Rodriguez, nombre=Marcela, dni=22222222, kWSubsidio=0]

ClienteCorporativo [numeroCliente=7324211, cuit=333333333333, razonSocial=UTT]

ClienteCorporativo [numeroCliente=7329827, cuit=444444444444, razonSocial=Cooperativa El Grito Sagrado]

Test2) Agregar e imprimir las siguientes tarifas:

Tarifa [consumoDesde=0, consumoHasta=50, cargoFijo=300.0, cargoVariable=2.0]

Tarifa [consumoDesde=51, consumoHasta=100, cargoFijo=350.0, cargoVariable=2.1]

Tarifa [consumoDesde=101, consumoHasta=200, cargoFijo=450.0, cargoVariable=2.5]

Tarifa [consumoDesde=201, consumoHasta=400, cargoFijo=500.0, cargoVariable=3.0]

Tarifa [consumoDesde=401, consumoHasta=2000, cargoFijo=1000.0, cargoVariable=4.0]

Test3) Agregar e imprimir las siguientes lecturas de medidor:

LecturaMedidor [fecha=2021-12-25, ClienteResidencial [numeroCliente=4483979, apellido=Perez, nombre=Juan, dni=11111111, kWSubsidio=500], estadoMedidor=180]

LecturaMedidor [fecha=2022-01-25, ClienteResidencial [numeroCliente=4483979, apellido=Perez, nombre=Juan, dni=11111111, kWSubsidio=500], estadoMedidor=255]

LecturaMedidor [fecha=2021-12-25, ClienteResidencial [numeroCliente=4971920, apellido=Rodriguez, nombre=Marcela, dni=22222222, kWSubsidio=0], estadoMedidor=100]

LecturaMedidor [fecha=2022-01-25, ClienteResidencial [numeroCliente=4971920, apellido=Rodriguez, nombre=Marcela, dni=22222222, kWSubsidio=0], estadoMedidor=165]

LecturaMedidor [fecha=2021-12-25, ClienteCorporativo [numeroCliente=7324211, cuit=333333333333, razonSocial=UTT], estadoMedidor=3526]

LecturaMedidor [fecha=2022-01-25, ClienteCorporativo [numeroCliente=7324211, cuit=333333333333, razonSocial=UTT], estadoMedidor=4385]

LecturaMedidor [fecha=2021-12-25, ClienteCorporativo [numeroCliente=7329827, cuit=444444444444, razonSocial=Cooperativa El Grito Sagrado], estadoMedidor=10862]

LecturaMedidor [fecha=2022-01-25, ClienteCorporativo [numeroCliente=7329827, cuit=444444444444, razonSocial=Cooperativa El Grito Sagrado], estadoMedidor=11981]

Test4) Traer e imprimir la lectura del mes=1, anio=2022, cliente numeroCliente=4483979

Test5) Traer e imprimir la lectura anterior del mes=12, anio=2021, cliente numeroCliente=4483979

Test6) Calcular e imprimir el consumo entre los objetos Lectura del punto Test4 y Test5

Test7) Traer e imprimir la tarifa para el consumo calculado en el Test6

Test8) Calcular e imprimir el total a cobrar al cliente con numeroCliente 4483979 y el periodo 1/2022