



# INFORME PROYECTO ENERGIA ELECTRICA

Joel Santiago Garcia Arevalo – 910154

Valentina Giraldo - 974754

Programación Orientada a Objetos

UNIMINUTO - 2025

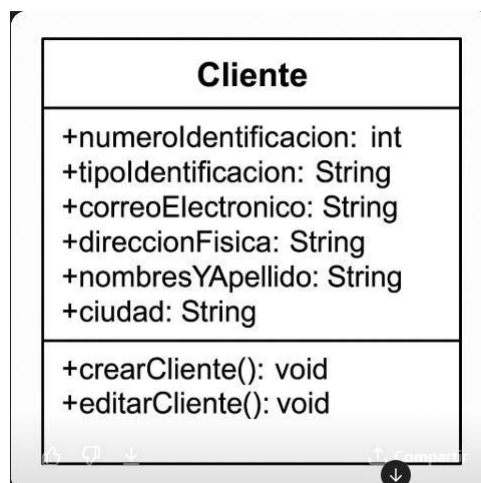
Para nuestro sobre energía eléctrica, hicimos una búsqueda de información desde la descripción del proyecto. Se paramos sustantivos, adjetivos mas sustantivos y verbos, lo que inicialmente nos arrojó esto:

SUSTANTIVOS	CALIFICATIVOS	VERBOS
cliente.java	numero unico de identificacion	Crear cliente
	tipo de identificacion	editar cliente
	correo electronico	
	direccion fisica	
	nombre	
	ciudad	
registrador.java	numero unico de identificacion	crear registrador
	direccion fisica	editar registrador
	ciudad	
consumo.java	hora	cargar consumos
	fecha	cambiar consumo
	franja	

Con esto sabemos que tendremos tres clases principales, cada una con sus respectivos atributos y métodos. Para modelarlo nos guiamos de una inteligencia artificial. A través de las siguientes preguntas:

“crear una clase cliente.java, con el numero único de identificación, tipo de identificación, correo electrónico, dirección física, nombres y apellido, y ciudad. Con los métodos, crear cliente, y editar cliente y generar un diagrama uml.png Puedes tener en cuenta que prefiero a la hora de codificar, los métodos nombrarlos de la manera, mMetodoUno. si llegasen a existir objetos, nombrarlos de la manera objObjeto.”

A lo que nos respondió con el código y con este diagrama:



Despues de eso, era momento de crear el siguiente modelo, es cierto, la IA me pregunto si seguir con el controlador, pero vamos por partes, le pregunte lo siguiente:

“antes de pasar a los controladores, vamos con el siguiente modelo, crear una clase registrador.java, con el numero unico de identificacion, direccion fisica, y ciudad. Con los metodos, crear registrador, y editar registrador y generar un diagrama uml.png  
Recuerda tener en cuenta las recomendaciones que ya te mencione”

Y creo este UML:

Registrador	- numIdentificacion: String - direccionFisica: String - ciudad: String	+ mCrearRegistrador(): void + mEditarRegistrador(): void
-------------	--	---

Para crear la clase consumo pregunte lo siguiente:

“si, vamos con el siguiente modelo, crear una clase consumo.java, con hora, fecha, franja. Con los metodos, cargar consumo, y cambiar consumo y generar un diagrama uml.png  
Para esta clase debes tener en cuenta que el consumo tiene una relacion con los clientes y los registradores, un cliente puede tener varios registradores, pero un registrador no puede tener muchos clientes. asi mismo, un registrador puede tener varios consumos, pero un consumo no muchos registradores. El consumo se medira todos los dias de una mes, cada hora de ese dia. El metodo de crear el consumo creara un consumo aleatorio en kw. Debes saber que tambien es necesario medir el costo de esos kw, este precio dependera de la franja horaria. Las franjas horarias son:

1. La franja uno entre las 00 y las 06 am y el consumo está dentro del rango de 100

= kW/H <= 300 y el precio es de 200COP por kW/H

2. La segunda franja entre la 7am y las 5pm y el consumo oscila en el rango de 300

kW/H <= 600 y el precio es de 300COP por kW/H

3. La tercera franja en el horario de 6pm y las 11pm y el oscila entre 600 > kW/H <1000 y el precio es de 500COP por kW/H

Antes de darme el codigo, crea un diagrama de la relacion de mis tres clases.”

A lo que creo este diagrama de la relacion de mis clases:



Luego si creo el respectivo código

## ## Creators

Joel Santiago Garcia Arevalo - 910154 - Universidad Minuto de Dios | [joel.garcia@uniminuto.edu.co](mailto:joel.garcia@uniminuto.edu.co)

Laura Valentina Giraldo Vargas - 974754 - Universidad Minuto de Dios | [laura.giraldo-va@uniminuto.edu.co](mailto:laura.giraldo-va@uniminuto.edu.co)

## ## Descripción

Este es un proyecto desarrollado en java. En el analizamos, creamos y gestionamos datos sobre los clientes en el sector de la energía eléctrica de Bogotá. Los clientes tienen asociados registradores, cada cliente puede tener más de un registrador. Estos registradores a su vez tienen varios registros de consumo de energía. El costo de este consumo de energía va a depender de una franja horaria, según la hora el costo cambia.

Este proyecto cuenta con:

- Creación y actualización de clientes.
- Creación y actualización de registradores asociados a los clientes.
- Cargar consumos de un usuario por hora o por mes.
- Cargar todos los consumos de todos los usuarios.
- Generar factura tipo .pdf con los datos del cliente de un mes en específico.

## ## Creación pdf

La creación del pdf se hizo mediante la librería itext pdf, para esta fue necesario crear el archivo pom.xml, en el que hay dependencias y configuraciones. También fue necesario añadir a nuestra carpeta lib, el archivo .jar para que pueda leer la librería.

Esta librería no solo funciona para crear archivos pdf, sino para muchas cosas más, este es el link del repositorio que contiene el código base de itextpdf:

<https://github.com/itext/itextpdf/releases/tag/5.5.13.3>

Esta librería la encontramos a través de un foro de la página web StackOverFlow, una plataforma de discusión para desarrolladores, donde se pueden resolver dudas, nos respaldamos en esta para encontrar la librería, a través de ella llegamos también a descargar el archivo .jar con toda la librería para poderla implementar:

<https://repo.itextsupport.com/ui/native/releases/com/itextpdf/itextpdf/5.5.13.3/>

## ## Estructura

EnergiaElectricaPoo/

.vscode/

settings.json

bin/

controllers/

ConsumoControlador.class

SistemaEnergiaCliente.class

documents/

InformeEnergiaElectrica.pdf

models/

Cliente.class

Consumo.class

GenerarFactura.class

Registrador.class

views/

VistaConsola.class

App.class

lib/

itextpdf-5.5.13.3.jar

src/

controllers/

ConsumoControlador.java

SistemaEnergiaCliente.java

documents/

InformeEnergiaElectrica.pdf

models/

Cliente.java

Consumo.java

GenerarFactura.java

Registrador.java

views/

VistaConsola.java

App.java

pom.xml

README.md

En el **\*\*controllers\*\*** estan los controladores que gestionan la logica del proyecto.

En el **\*\*models\*\*** estan los modelos que son las clases que contienen los datos necesarios del proyecto.

En el **\*\*controllers\*\*** esta la clase que me permite ejecutar la aplicación

## **## Instalacion**

Clona el proyecto a traves de git:

<https://github.com/joelgarciaarevalo/EnergiaElectricaPoo>