JAVA PROJECT ENERGY

JUAN ALEJANDRO APACHE MORENO

JUAN DIEGO ARIAS SILVA

JEISON JULIAN GUERRERO LIBERATO

CORPORACION UNIVERSITARIA MINUTO DE DIOS

PROGRAMACION ORIENTADA A OBJETOS

BOGOTÁ D.C

Este proyecto fue desarrollado con el objetivo de simular un sistema de facturación de energía eléctrica, orientado a la gestión de clientes y el consumo de energía. Los elementos principales identificados en la fase de análisis fueron:

- **Usuarios**: Clientes que consumen energía y deben recibir facturación.
- **Requisitos funcionales**:
    - Registro de clientes.
    - Registro de consumos eléctricos.
    - Cálculo automático del valor a pagar.
    - Generación de factura en formato PDF.
- **Requisitos no funcionales**:
    - Arquitectura basada en el patrón MVC.
    - Uso de Java como lenguaje principal.
    - Uso de bibliotecas externas para generación de PDFs (iText).

2. Diseño

El diseño sigue una arquitectura **MVC (Modelo-Vista-Controlador)**:

- **Modelo (Model)**:
    - Cliente.java: Clase que encapsula la información del cliente (nombre, ID, etc.).
    - Consumo.java: Clase que representa el consumo de energía por parte de un cliente.
    - Registrador.java: Gestiona las colecciones de clientes y consumos.
- **Controlador (Controller)**:
    - ControladorCliente.java: Administra las operaciones sobre los clientes.
    - ControladorConsumo.java: Administra los consumos y calcula valores de factura.
- **Vista (View)**:
    - Vista.java: Maneja la interacción con el usuario a través de consola.
- **Aplicación Principal (App)**:
    - App.java: Inicia el sistema, enlaza los controladores con la vista y ejecuta el menú principal.
- **Dependencias**:
    - iTextPDF: Para la generación de facturas en formato PDF.

3. Codificación

El desarrollo del código fue realizado utilizando **Java SE**, empleando buenas prácticas como:

- Encapsulamiento de datos en el modelo.
- Separación de responsabilidades entre clases.
- Modularidad en los controladores para facilitar mantenibilidad.
- Uso de colecciones para almacenar datos.
- Generación de documentos PDF que simulan facturas reales.

Ejemplos de implementación:

- En ControladorConsumo.java, se realiza el cálculo del valor a pagar en base a un consumo registrado.
- En Registrador.java, se almacenan listas de clientes y consumos con métodos para añadir y consultar datos.
- En Vista.java, se presenta un menú interactivo para que el usuario pueda registrar clientes, consumos y generar facturas.