RELATÓRIO POO LÍDER JOÃO CARLOS

**Introdução**: a fim de, não ficar redundante detalhes sobre para que serve o trabalho algo que foi feito com bastante zelo na própria parte do READ.ME, esse relatório tem como intuito apresentar dificuldades, dúvidas, acertos e demais coisas que possam agregar e acrescentar no entendimento e realização do mesmo. Desde já, agradeço ao professor Akira e Hugo por proporcionar o ambiente de estudo adequado e me tirar um pouco da ´´zona de conforto´´, para que assim houvesse uma evolução nas habilidades e aperfeiçoamento da mesma.

**Etapa 1**: Nessa primeira etapa, meu grupo apresentou uma dificuldade imensa. Nosso líder acabou precisando sair por motivos pessoais, deixando nosso grupo um pouco disperso, todavia, procurarmos juntos alcançar o objetivo sendo que, cada integrante ajudou um pouco e um pouco mais, expandindo além de suas issues específicas. Ademais, a etapa 1 teve essas complicações, todavia, conseguimos criar um pequeno programa com alguns códigos que já havíamos, mas aperfeiçoamos, mas sem uma estrutura mvc apropriada. (Deixarei como anexo os códigos que tínhamos.)

**Etapa 2**: Essa etapa teve suas complicações no que diz respeito a IDE. O bluej não foi uma ferramenta funcional para o nosso projeto, de forma que tivemos que mudar um pouco nosso foco que era ´´Criar um programa que fizesse uma tabela depois de colocarmos os dados´´, para criamos um programa mais simples que apenas calculasse o fator de potência e os watts consumidos. Foi meio desesperador, pois eu já tinha boa parte dos códigos e ter que passar para o vscode, aprender mexer em uma nova IDE foi muito cansativo, além de, aprender a mexer e situar o controller e a view, pois eu estava tendo esse problema no bluej. Eu estava sendo limitado a não ter a view, pois não abria (erro do diretório que graças ao Carlos Eduardo conseguimos arrumar), além de não conseguir abrir a interface muitas das vezes. Ademais, conseguir finalizar essa etapa com uma ideia mais simplista que a original, todavia, um protótipo desenvolvido de forma orgânica e que me levou a exposição de algumas ´´fraquezas práticas´´, onde eu preciso melhorar ou não.

**O que precisa ser feito**: A partir desse projeto, podemos ter uma ideia de como será o mercado de trabalho. Há muito desafios ainda para superar acerca desse trabalho, principalmente no que diz impressão de tabela (que era nossa ideia inicial), de modo que futuramente será necessários novos commits acerca do código e da view.

Anexo dos códigos:

ConsumoEnergia:

import java.util.Scanner;

public class ConsumoEnergia {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Informe a potência em Watts: ");

double potencia = scanner.nextDouble();

System.out.print("Informe as horas de uso por dia: ");

double horas = scanner.nextDouble();

System.out.print("Informe o número de dias no mês: ");

double dias = scanner.nextDouble();

double consumo = calcularConsumo(potencia, horas, dias);

System.out.println("Consumo de Energia: " + consumo + " kWh");

}

public static double calcularConsumo(double potencia, double horas, double dias) {

return (potencia \* horas \* dias) / 1000; // Conversão para kWh

    }

}

FatorPotência:

import java.util.Scanner;

public class FatorPotencia {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Informe a potência ativa em Watts: ");

double potenciaAtiva = scanner.nextDouble();

System.out.print("Informe a potência aparente em VA: ");

double potenciaAparente = scanner.nextDouble();

double fatorPotencia = calcularFatorPotencia(potenciaAtiva, potenciaAparente);

System.out.println("Fator de Potência: " + fatorPotencia);

}

public static double calcularFatorPotencia(double potenciaAtiva, double potenciaAparente) {

return potenciaAtiva / potenciaAparente;

    }

}

DemandaMaquinas:

import java.util.Scanner;

public class DemandaMaquinas {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Informe a potência de cada máquina em Watts: ");

double potenciaPorMaquina = scanner.nextDouble();

System.out.print("Informe o número de máquinas: ");

int numeroMaquinas = scanner.nextInt();

double demandaTotal = calcularDemandaTotal(potenciaPorMaquina, numeroMaquinas);

System.out.println("Demanda Total das Máquinas: " + demandaTotal + " kW");

}

public static double calcularDemandaTotal(double potenciaPorMaquina, int numeroMaquinas) {

return (potenciaPorMaquina \* numeroMaquinas) / 1000; // Conversão para kW

    }

}

Maquinas:

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Maquina {

double potencia; // Potência em Watts

double horasPorDia; // Horas de uso por dia

double fatorPotencia; // Fator de Potência

Maquina(double potencia, double horasPorDia, double fatorPotencia) {

this.potencia = potencia;

this.horasPorDia = horasPorDia;

this.fatorPotencia = fatorPotencia;

}

}

Eficiencia setor:

public class EficienciaSetor {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Maquina> maquinas = new ArrayList<>();

System.out.print("Informe o número de máquinas no setor: ");

int numeroMaquinas = scanner.nextInt();

for (int i = 0; i < numeroMaquinas; i++) {

System.out.println("Máquina " + (i + 1) + ":");

System.out.print("Informe a potência em Watts: ");

double potencia = scanner.nextDouble();

System.out.print("Informe as horas de uso por dia: ");

double horasPorDia = scanner.nextDouble();

System.out.print("Informe o fator de potência: ");

double fatorPotencia = scanner.nextDouble();

maquinas.add(new Maquina(potencia, horasPorDia, fatorPotencia));

}

double consumoTotal = calcularConsumoTotal(maquinas);

System.out.println("Consumo Total de Energia do Setor: " + consumoTotal + " kWh");

double fatorPotenciaMedio = calcularFatorPotenciaMedio(maquinas);

System.out.println("Fator de Potência Médio do Setor: " + fatorPotenciaMedio);

}

public static double calcularConsumoTotal(List<Maquina> maquinas) {

double consumoTotal = 0;

for (Maquina maquina : maquinas) {

consumoTotal += (maquina.potencia \* maquina.horasPorDia \* 30) / 1000;

}

return consumoTotal;

}

public static double calcularFatorPotenciaMedio(List<Maquina> maquinas) {

double fatorPotenciaTotal = 0;

for (Maquina maquina : maquinas) {

fatorPotenciaTotal += maquina.fatorPotencia;

}

return fatorPotenciaTotal / maquinas.size();

    }

}