

|  |
| --- |
| Parque Eólico -  Trabalho Final |
|  |
| 2 junho  Programação II   Diogo Rodrigues 42743  Diogo Antão 43734 |



*Introdução*

Este trabalho consiste na criação de um programa em linguagem JAVA para a gestão de um empreendimento eólico, composto por vários aerogeradores em diferentes posições (latitude, longitude e altitude), com este programa conseguimos adicionar / remover aerogeradores ao sistema, adicionar ou remover componentes dos mesmo pois diferem entre si como também devolver diferentes métodos de cálculo das potências produzidas pelos mesmos.

*Uma imagem com texto, jornal, documento

Descrição gerada automaticamente*

*Estruturação do trabalho e construção de classes*

A classe Aerogerador, onde é iniciado o construtor para cada aerogerador juntando assim as informações acerca da sua localização (altitude, latitude, longitude, localidade e data de instalação), a sua altura (distância do eixo do rotor ao chão), para efeitos de densidade do ar para cálculo das potências consideramos o ponto mais alto como altitude (distância do nível do mar até à base do aerogerador) + altura.

São definidos também métodos para calcular a área do rotor tendo em conta o seu raio getArea() , como métodos para calcular o coeficiente de potência de acordo com o cenário do aerogerador que diferem em tamanho do raio e tipo de gerador, dependendo desses valores é calculado através do método *potenciaTurbina(double)* a potencia do aerogerador tendo em conta essas características.

Nesta classe é definida também métodos para fazer manutenções ao aerogerador (trocar rotor, trocar gerador ou ambos)

Na classe Atmosfera é criada através de um construtor uma atmosfera com os métodos para retornar o valor da densidade do ar a uma determinada altura escolhida pelo utilizador, foram acrescentados mais 3 métodos para retornar a temperatura do ar (esta fórmula só é válida dentro da troposfera aprox. 12000km, e com isto não será possível criar atmosferas acima dos 12000 metros de altitude).

A classe CoeficientePotencia é uma classe abstrata que consoante o tipo de cenário em que o aerogerador se encaixar irá retornar os valores da tabela das velocidades para cada valor de CP. Em que CoeficientePotencia é a superclasse, cenarioA e cenarioB são subclasses da mesma.

A classe ParqueEolico é onde estão os métodos relativos ao empreendimento eólico como um conjunto de objetos, para a sua definição criamos um ArrayList de Aerogeradores. Nesta classe o seu construtor cria num novo ArrayList, e conta com métodos para devolver o número de aerogeradores no parque, as informações de cada aerogerador ativo, métodos para adicionar e remover aerogeradores, métodos de manutenção de um determinado aerogerador escolhido pelo utilizador e métodos de cálculo de potências, o método *potenciaInst(double,int)* devolve o valor da potencia de um determinado aerogerador num determinado instante.

O método *potenciaInst(double[],int)* recebe um array com vários instantes(velocidades) e devolve um array com os valores das potências de um determinado aerogerador .

O método *somaPotenciaInt(double)*recebe um instante(velocidade) e retorna a soma de todas as potencias para todos os aerogeradores no ArrayList parque.

O método intermedio que ajuda na realização de contas intermedias para o método seguinte *potenciaCord (double[],double[],double[],int)* recebe um conjunto de Estações Meteorológica (velocidade,longitude,latitude) e verifica qual a estação esta mais perto de um dado aerogerador e retorna a potencia associada.

O método *somaPotenciaCord(double[], double[], double[],int)* recebe um conjunto de Estações Meteorológica( velocidade,longitude,latitude) e retorna uma estimativa da potencia global produzida, com a melhor aproximação à velocidade do vento junto de cada aerogerador.

A classe Main é uma classe que contem duas opções de main uma com os testes para as soluções corretas confirmadas de forma analítica e os testes aos erros, mais em baixo em comentário temos uma main interativa que contem um menu que consoante aquilo que o utilizador deseja fazer num ciclo *do while* e que interage com todas as outras classes.

Testes realizados analiticamente

Caso 1 – CenarioA

//Aerogerador (altitude, longitude, latitude, rotor, localidade, data, gerador, altura)

Aerogerador (30, 10000, 20000, 5, “Moita”, "05-06-2022", "G1", 20)

𝑇 = 𝑇0 − 𝐿ℎ

Caso 2 – CenarioB

Aerogerador (50, 10000, 20000, 5, “Pinhal Novo”, "22-07-2022", "G2", 50)

𝑇 = 𝑇0 − 𝐿ℎ