

**UNIVERSIDADE UNIP TATUAPÉ**

**CURSO DE GRADUAÇÃO CIÊNCIAS DA COMPUTAÇÃO**

**DAVI COUTINHO PIRES**

**BRUNO SOARES CARVALHO**

**GUSTAVO BEHRING ARAÚJO**

**HENRIQUE RUIZ DO NASCIMENTO**

**MATHEUS MOREIRA DIAS**

**ATIVIDADES PRÁTICAS SUPERVISIONADAS**

**LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS**

**SÃO PAULO**

**2021**

## SUMÁRIO

INTRODUÇÃO .....	3
OBJETIVO.....	5
ELEMENTOS DE PROGRAMAÇÃO ORIENTADA A OBJETOS.....	6
DISSERTAÇÃO.....	12
PROJETO (ESTRUTURA) DO PROGRAMA .....	18
RELATÓRIO COM AS LINHAS DE CÓDIGO .....	21
BIBLIOGRAFIA .....	25

## INTRODUÇÃO

Já faz muito tempo desde que os cientistas alertaram para os perigos que a humanidade correria caso o descuido com o meio ambiente continuasse para que de forma egoísta se mantivesse o conforto e consumismo das sociedades. Todo processo industrial é completamente dependente de insumos, que são os recursos necessários para manter a cadeia de produção.

Esses recursos podem ser mão de obra, infraestrutura, recursos financeiros e o que será analisado nesse trabalho, os recursos naturais. Alguns processos geofísicos na Terra são devidamente equilibrados de forma que coincidentemente, haja condições apropriadas de vida, e o que vem ganhando mais atenção é o efeito estufa. Ele não só protege a Terra de raios ultravioleta, como também proporciona um controle de temperatura que é essencial para toda a biodiversidade.

O grande problema evidenciado é quando a indústria de forma geral utilizando de recursos naturais, consegue gerar poluentes que degradam o meio atmosférico. O que será evidenciado nesse trabalho são os poluentes gerados pela indústria automobilística, e não só a produção do produto, mas a utilização no dia a dia desses agentes poluentes que são frutos de uma tecnologia considerada ultrapassada que ainda se mantém em alta graças a falta de infraestrutura e investimentos governamentais, além de logística e a situação econômica de cada país.

Dessa forma, faz-se necessário assimilar e relacionar parte da culpa dos danos atmosféricos causados à utilização de veículos que utilizam a queima de combustíveis fósseis para sua locomoção, propor soluções inovadoras e conscientizar as pessoas acerca da importância de possíveis revoluções nesse meio.

Como proposta de conscientização, conterà nesse documento informações básicas relativas à importância da conservação do meio ambiente mediante os impactos de ações humanas. Também será apresentado um programa feito em linguagem Java com conceitos fundamentais de programação orientada a objetos, que coletará dados do usuário, os processarão e a partir de uma lógica baseada em métodos definidos por entidades governamentais, informará ao usuário o nível de CO<sub>2</sub> emitido por seu veículo pessoal, além de informar o quanto é possível deixar de poluir caso utilize meios alternativos.

## OBJETIVO

Esse trabalho tem por objetivo adequar o leitor a compreensão de conceitos básicos presentes na relação de ações humanas e o impacto negativo no planeta Terra. Além de, estabelecer conexão entre os veículos automóveis movidos a combustão com o aquecimento global recorrente e futuro.

Quantificar a poluição atmosférica e principalmente a emissão de CO<sub>2</sub> gerada por esses veículos, levando como base cálculos determinados por órgãos especialistas em estimar essa categoria de dado.

Projetar e construir um programa em Java utilizando conceitos de programação orientada a objeto, que necessite de entrada de dados por um usuário e informe com base nesses dados informados, comparações que levem a conscientização e uso racional dos veículos, com propósito à sustentabilidade.

## ELEMENTOS DE PROGRAMAÇÃO ORIENTADA A OBJETOS

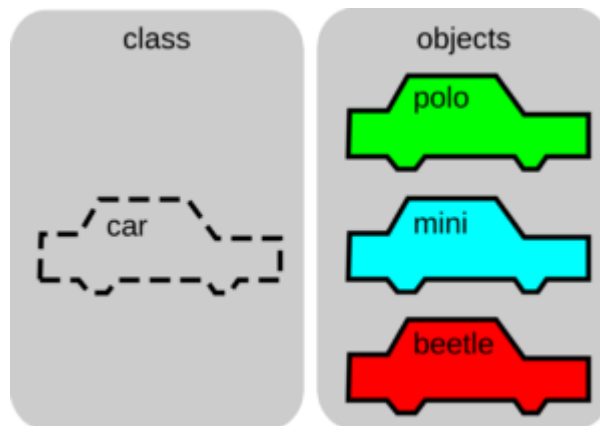
Para construção do programa descrito nesse trabalho, será utilizado a linguagem de programação Java, e a confecção será baseada no paradigma de programação orientado a objetos. Mas para isso, deve-se primeiro evidenciar o que são e para que servem cada um desses elementos.

Diferente do paradigma estruturado de programação, a orientação a objetos é baseada em recursos de modularização de processos, tentando aproximar esses processos a realidade. Essa modularização do programa é feita a partir de classes, que interagem entre si através da criação de objetos de outras classes (um elemento baseado em generalização), fazendo com que seja possível utilizar seus métodos ou atributos.

Sendo mais descritivo, uma classe é como um molde que possui a função de categorizar um determinado grupo de elementos que serão utilizados durante a execução do sistema. E dentro do paradigma discutido, elas auxiliam muito na visualização de um projeto na totalidade, fazendo com que a construção e manutenção do código seja mais dinâmica graças a diminuição da quantidade de linhas utilizadas em cada módulo. Nesse processo de categorização do programa, é possível atribuir-lhes algumas ações que os seus elementos podem executar e características, ou qualidades, que são respectivamente os métodos e atributos de uma classe.

Já os objetos servem para conectar esses módulos conforme a categoria adotada pela classe, os atributos e os métodos. Com eles é possível executar ações em uma classe que em outra também deveria ser possível, além de criar os elementos que fariam parte desse conjunto designado. Um exemplo simples capaz de demonstrar os conceitos descritos acima, é a imagem abaixo:

**Figura 1** – Classes e Objetos



**Fonte:** <https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>

Como é possível observar, a classe é uma abstração de uma categoria de um elemento qualquer. Os objetos por sua vez são como os elementos criados propriamente ditos, podendo conter nomes, cores ou ações e qualidades diferentes, desde que elas estejam especificadas na classe.

Dentro do conceito de classe, é possível estabelecer algumas qualidades ou características para o determinado grupo de elementos. Essas características são chamadas de atributos, se parecem muito com variáveis e graças a linguagem Java ser fortemente tipada, é necessário indicar o tipo primitivo ao declará-lo.

Para guardar caracteres, pode ser usada o tipo primitivo `char`, que armazena em memória um caractere, ou a classe nativa do Java `String`, que permite armazenamento de diversos caracteres em sequência. Para números existem um pouco mais de opções, sendo elas `short` para guardar números de -32.768 a 32767, `int`: -2.147.483.648 a 2.147.483.648 e `long`: -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807. Por final, para tipo ponto flutuante (com casas depois da vírgula) podem ser utilizados `float` para precisão de até 7 casas decimais e `double` para precisão de 15 casas decimais.

Além dos atributos, é possível criar métodos nas classes, que são como abstrações de ações do elemento físico a se tratar, todavia, são parecidos com funções ou procedimentos da programação estruturada. É possível dois tipos de métodos básicos, o void que apenas é chamado para executar alguma outra função, ou então o método com retorno. Neste citado anteriormente, é necessário adicionar a implementação return e definir um valor correspondente ao tipo primitivo declarado no início do método.

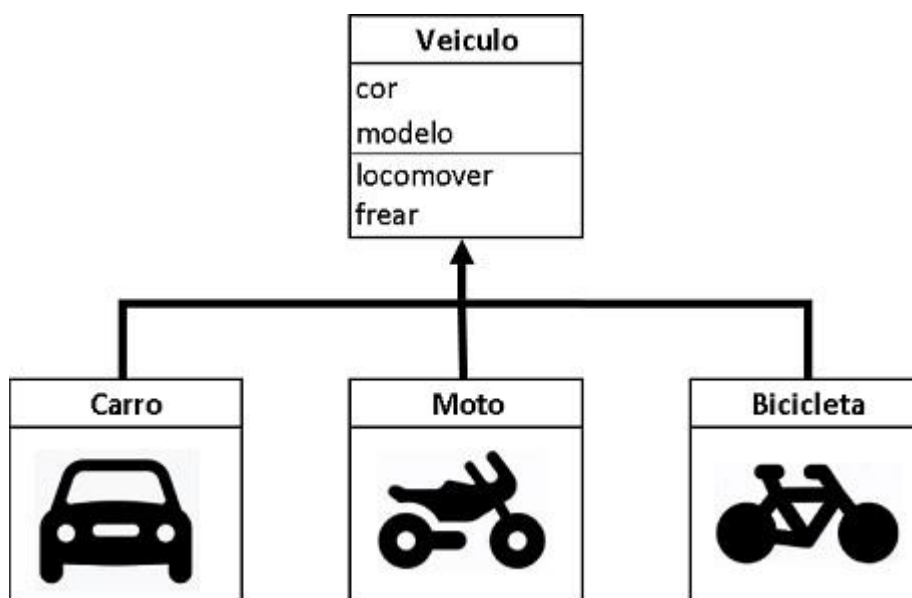
Uma forma de lidar com os atributos e de formatar de forma coerente um programa, é através do encapsulamento. Ao tornar o atributo privado, apenas a classe a qual ele foi criado será capaz de usá-lo e manipulá-lo, no entanto, utilizando os métodos get e set é possível utilizar esses atributos com os métodos utilizando a instância da classe correspondente. O método set recebe um valor do tipo do atributo como parâmetro e faz o atributo receber aquele valor, assim como, o método get serve para pegar e utilizar esse valor atribuído ao atributo. Com isso, é possível definir regras dentro dos métodos para que não sejam alterados ou violados de forma indevida pelo usuário.

Algumas vezes nos programas é necessário alterar um método já declarado em outra classe ou manipulá-lo, para isso, pode ser utilizado o conceito de sobrecarga de métodos. Consiste basicamente em reescrever o método, porém alterando sua assinatura, que são os tipos primitivos dos parâmetros desse método, assim sendo podendo com um mesmo método, executar funções diferentes. Esse é um conceito também de polimorfismo, que será especificado mais para frente ao mencionar herança.

Para evidenciar outros conceitos é necessário primeiro estabelecer o conceito de herança entre classes. Uma herança como de forma intuitiva em seu nome diz, é o processo de passagem de algum valor para algum elemento ou indivíduo. Nesse caso é a passagem de todos os atributos e métodos de uma classe, denominada classe-mãe, para outras classes, denominadas classes filhas, através do comando extends. Esse conceito abre muitas possibilidades e funções para um programa, uma delas é a adição de novas classes que sejam semelhantes ou tenham características únicas, porém também as características da classe mãe.



**Figura 2 – Herança**



**Fonte:** <https://testeavelocidade.com.br/heranca-superclasse-e-subclasse-java/>

Através da herança, é possível utilizar funções para boas práticas de programação que tornam o sistema muito mais padronizado e entendível para manutenção, reutilização de código e a própria implementação de funções novas, entre eles, o método construtor deve ser citado. O método construtor é responsável pela criação de instâncias de uma classe qualquer em outra, ou seja, o objeto. Sempre que é criado um objeto em uma classe sem criar o método construtor na classe que vai ser instanciada exigindo algum parâmetro na assinatura do método, um construtor vazio é criado pelo próprio Java sem implementação.

Entretanto, criar o método construtor pedindo algum parâmetro que será atribuído a algum atributo da classe ou da classe mãe é uma boa prática de programação, e ao atribuir o parâmetro à classe mãe, é necessário utilizar o comando `super`. Caso haja mais de um método construtor com o mesmo nome, mas com assinatura diferente, a própria linguagem de programação identificará qual deles deve utilizar com base no parâmetro enviado pelo programa.

Agora com o conhecimento de herança em mente, é possível descrever o que é sobrescrita de um método. Esse conceito é um tipo de polimorfismo que consiste em reescrever um método existente em uma classe mãe e diferente da sobrecarga, mantém o tipo de parâmetro definido no método, mas cria uma versão daquele método, podendo alterar suas funções e utilizar aquele que o deseja seja na classe filha ou na classe mãe. Este conceito citado é uma clara representação de polimorfismo, que é a utilização de um elemento de formas diferentes.

Finalizando os conceitos principais do paradigma de programação orientado a objetos, é preciso relacionar as funções explicadas com os modificadores de acessos e comportamentos. O modificador public, garante que os elementos de um programa sejam acessados e modificados por qualquer classe em qualquer pacote. Enquanto o protected restringe o acesso desse elemento a apenas acessível através dos pacotes. E o private é responsável por proibir o acesso aos elementos designados de qualquer classe ou pacote. Dessa forma, só é possível acessá-los através dos métodos get e set, que além de padronizar o sistema, adiciona segurança seja da pessoa que está programando, ou de uma situação atípica.

Os modificadores de comportamento como no próprio nome dizem, serve para mudar o comportamento inicial que um determinado elemento do sistema possui. Existem três, são eles, static, final e abstract. Uma classe pode ter apenas um entre final e abstract, sendo que final torna a classe impossível de ser instanciada, ou seja, criar um objeto em memória, enquanto abstract torna a classe possível de ser instanciada, porém, apenas através de suas classes filhas, obrigando que seja implementado a herança.

Para métodos e atributos, também alguns dos três modificadores pode ser utilizado, sendo que o static torna um método e um atributo mais fáceis de serem acessados e utilizados, possibilitando que sejam manipulados e invocados sem que haja a presença de um objeto da classe da qual correspondem. Já no modificador final, torna tanto o método quanto o atributo impossível de manipular através de uma possível sobrescrita ou sobrecarga, e de alterar valores de um atributo, tornando-o uma constante. Por final para métodos portando o abstract,

é obrigatório que eles sejam implementados pelas classes filhas, dessa forma, eles não possuem implementação na superclasse, além disso, só podem existir em classes abstratas.

Para terminar os conceitos principais do paradigma abordado nesse trabalho, um conceito que não tem em todas as linguagens, mas que possui na linguagem utilizada para esse trabalho, é a “interface”. Existem diversas discussões na web que se trata de se uma interface é uma classe ou algo diferente. Internamente ao codificá-la, se declara como interface, porém, ao salvar e buscar o arquivo criado no diretório guardado, a extensão é a mesma de uma classe. Entretanto, ela possui um comportamento diferente de uma classe normal, bem-parecido com o de uma classe abstrata. A diferença entre uma interface e uma classe abstrata é que na interface não se pode declarar atributos e todos os seus métodos devem obrigatoriamente ser implementados em alguma outra classe ao colocar “implements” na declaração dessa classe.

## DISSERTAÇÃO

Hoje em dia, diante do cenário em que, grande parte do desenvolvimento da humanidade se encontra com atenção voltada para a sustentabilidade no meio ambiente, se faz necessário entender as implicações negativas causadas pelos impactos de ações humanas no planeta, estabelecer relações de causa e efeito que demonstrem atitudes corretas para combater os problemas e desafios acerca desse assunto, além de propor soluções inovadoras e eficazes que garantam que seja preservado o ambiente terrestre.

A princípio, é importante entender o porquê cuidar do meio ambiente. No planeta terra, há diversas variáveis que permitem que exista a vida como é conhecida, dentre elas a água, que é importante tanto diretamente para os seres vivos que necessitam equilibrar sua temperatura corporal, seu P.H, sua energia ou qualquer outro fator ligado diretamente ao seu consumo, como também, seres que dependem de processos naturais da água para sobreviver como a chuva, a regulação da temperatura do planeta, de um bioma específico ou uma vegetação apropriada.

Outro meio físico importantíssimo que deve ser preservado, é o meio terrestre. Nele há o solo, que é responsável por manter a vida e ecossistema de toda fauna e flora. A partir do solo, as vegetações e biomas são formados, que por consequência, também viabiliza a existência dos ecossistemas.

Ademais, sem o solo, não haveria sequer a civilização montada até agora, talvez não haveria civilização nenhuma, já que sem os conhecimentos da agricultura, que são dependentes da terra, seria impossível os humanos saírem do status nômade e começar a criar casas, até eventualmente se transformar em uma sociedade. Isso sem contar os processos industriais, que dependem diversas vezes de substâncias e alimentos que são cultivados através de diversas técnicas, para satisfazer o consumo da sociedade atual e futura.

Embora seja necessário conservar toda a natureza, o meio físico que tem recebido muita atenção nesses últimos anos, é o meio atmosférico. A poluição atmosférica pode impactar negativamente o ar que está na atmosfera terrestre, causando danos à saúde dos animais, inclusive os humanos, gerando problemas em todo o sistema respiratório a longo prazo.

Soma-se a isso, é capaz de impactar a água, fazendo com que ocorra o fenômeno chuva ácida e ocorra diminuição na intensidade das tempestades. Além disso, pode atrapalhar a captação de energia solar, que é uma fonte de energia extremamente limpa e necessária, e impacta os rendimentos de colheitas de alimentos importantes como o trigo ou a soja.

Entretanto, a maior preocupação atual, é voltada para o impacto da poluição do ar para os climas. Nesse contexto, o maior receio é que os gases nocivos que estão no ar gerem o famigerado efeito estufa, que tem relação direta com o aquecimento global e que protege as coisas na atmosfera da terra dos raios ultravioleta do Sol.

Basicamente, a terra seria um planeta com uma temperatura mais baixa se não fosse pelo efeito estufa. Ele permite que haja a absorção e acúmulo de energia para que a temperatura terrestre se encontre em uma temperatura adequada para manter a vida, os processos bioquímicos e geofísicos como são conhecidos hoje. Esse efeito é causado por diversos gases que se acumulam na estratosfera, mas o que mais vale frisar é o ozônio, que dá nome a tão falada camada de ozônio.

A camada de ozônio além de ser responsável por manter o equilíbrio térmico do planeta, também mitiga os danos que seriam causados pelas radiações solares. Esse equilíbrio é muito importante, pois se há um desequilíbrio da quantidade de ozônio para mais, já que a própria camada se renova com o tempo, muito mais energia seria acumulado na terra e tornaria um ambiente inviável para vida, assim como Vênus. Da mesma forma que se houver menos ozônio do que o necessário, os raios solares não seriam filtrados e haveria um aquecimento global devido à quantidade de energia impactando diretamente os seres vivos gerando uma extinção em massa global.

O grande problema atual com relação a tudo isso, é o fato de a produção capitalista impulsionada pelo consumismo desvairado da sociedade estar gerando exatamente os gases que são tóxicos e reagem com o ozônio, em quantidades absurdas que não permitem que a camada de ozônio se recicle, e isso pode mudar toda a dinâmica da vida na terra.

Dentre tantos produtos e processos que impactam negativamente para a aceleração demasiada do aquecimento global e destruição da camada de ozônio, os veículos automotores a combustão são os principais vilões no combate desta causa, em especial os automóveis urbanos.

Eles são capazes de gerar processos de produção tóxicos para o meio ambiente, como a produção de etanol e gasolina em refinarias de petróleo. Além da queima desses combustíveis realizada por esses veículos também gerar poluentes que agravam essa problemática, inclusive quando se analisa esses dados a nível global.

O principal composto emitido pelos veículos que será discutido nesse trabalho, é o CO<sub>2</sub> (Dióxido de carbono). Um gás que é gerado a partir da queima de combustíveis fósseis e de todos os hidrocarbonetos, inclusive da queima de florestas a fruto de desmatamento. Esse gás é proveniente dos efeitos negativos citados acima, e estima-se que desde 1750 até 1999 já havia um aumento da concentração de CO<sub>2</sub> na atmosfera de 31%, inclusive, modelos sugerem que ele seja responsável por 60% do aquecimento global gerado pelas ações humanas.

Nesse sentido, esse trabalho tem por finalidade indicar através de equações e métodos já desenvolvidos, a quantidade de CO<sub>2</sub> emitido por um veículo segundo os dados de entrada do usuário, no intuito de conscientizar o usuário sobre suas ações e os impactos que influenciarão um futuro não tão distante.

Para determinar o impacto causado por um determinado veículo, sem considerar o processo de fabricação do automóvel, e demais processos corriqueiros, apenas com a emissão diária durante a queima, é preciso considerar alguns pontos. Como, por exemplo, o tipo de combustível utilizado, o consumo médio em km/L e a distância percorrida.

O tipo de combustível que será utilizado impacta diretamente o resultado, de modo que, o etanol tem níveis de emissão menores por se tratar de um composto renovável, entretanto, libera menos energia em sua reação gerando um torque menor do motor, além de prejudicar a eficiência do veículo, sendo necessário o abastecimento de forma mais recorrente.

Para efeitos de comparação, o diesel é extremamente poluente, sendo capaz de emitir muito CO<sub>2</sub> e ter uma eficiência extremamente baixa, algo em torno de 3 km/L em determinados veículos, tudo isso a custo de um torque maior do motor e preço menor por Litro. Mas será demonstrado abaixo que ele tem seu propósito, e há situações em que a gasolina que é muito eficiente pode acabar se tornando uma grande vilã.

Por sua vez, a gasolina é um vilão e tanto em questão de níveis de emissão de carbono na atmosfera. Ela com o diesel será o foco desse trabalho, e será analisado os níveis de emissão médio por pessoa considerando também a quantidade de pessoas em cada veículo além do combustível analisado que será: gasolina para carros populares ou urbanos, e Diesel para ônibus e outros veículos capazes de carregar diversas pessoas.

Para estimar os valores desejados, é necessário definir o percurso diário em km que será considerado, para efeitos didáticos o exemplo a seguir adotará ida e volta como 40 km. Após isso, basta multiplicar esse valor por 365 para saber a distância percorrida anual, que nesse caso é 14 600 km. Agora entra no cálculo o consumo médio de combustível, que para esse caso será um carro utilizando gasolina, e percorrendo 10 km/L. Com isso, divide-se a distância percorrida anual pelo consumo médio, ficando  $14\ 600/10 = 1460$  Litros, sendo este o consumo anual de gasolina.

A partir daqui basta multiplicar o consumo anual em Litros pelos fatores de multiplicação do combustível utilizado, que para esse caso tem-se  $1460 \cdot 0,82 \cdot 0,75 \cdot 2,21 = 1\ 984,35$  kg CO<sub>2</sub> anual. Os valores correspondentes aos fatores de multiplicação podem ser explicados pela densidade da gasolina que é 0,75 kg/L, assim como o fator de transformação em CO<sub>2</sub> ser igual a 2,21 kgCO<sub>2</sub> /L e por final, considere que a quantidade de etanol presente na gasolina do

Brasil é em torno de 18% a 25%, então o volume máximo de gasolina pura que se tem por litro de gasolina é 82%, o que explica o fator de multiplicação igual a 0,82.

Será utilizado essa mesma lógica para calcular a emissão de CO<sub>2</sub> de veículos a diesel, apenas alterando alguns fatores de multiplicação, como a transformação do combustível em CO<sub>2</sub> que será 3,41 KgCO<sub>2</sub>/L, além da densidade ser maior que a da gasolina, com o valor de 0,853 kg/L.

Soma-se a isso, os dados fornecidos pelo órgão CBCS, disponibilizados via web serão utilizados para processamento dos dados com base também na potência do motor do veículo analisado pelo usuário, e são eles:

**Figura 3 – Tabela Motor Combustível**

Motor	Etanol (gCO <sub>2</sub> /Km)		Gasolina 20% etanol (gCO <sub>2</sub> /Km)	
1.0 (flex)	44,9	61,0	133,7	152,2
1.4 a 1.6 (flex)	49,5	67,0	141,8	164,7
1.8 (flex)	58,4	78,2	170,7	188,7
1.4 a 3.8 (gasolina)			182,5	283,0

**Fonte:**[http://www.cbcs.org.br/sbcs10/website/userFiles/palestras\\_sbcs\\_10/emissao\\_co2\\_transporte.pdf](http://www.cbcs.org.br/sbcs10/website/userFiles/palestras_sbcs_10/emissao_co2_transporte.pdf)

É com base nesses princípios que o programa a ser construído nesse trabalho adotará para informar ao usuário o quanto seu veículo polui o meio ambiente, e mostrará comparações que definam o quanto transportes públicos como ônibus podem ser mais vantajosos no quesito abordado nesse artigo em determinadas situações, já que embora a emissão de poluentes seja maior, a quantidade de pessoas dentro deles, diminua proporcionalmente a média de CO<sub>2</sub> gerada para cada pessoa.



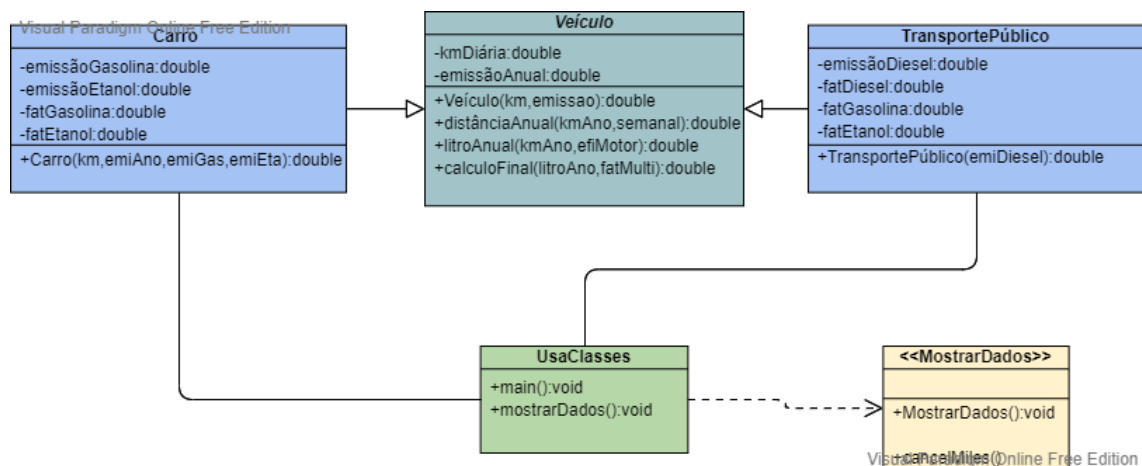
Soma-se a isso, também conscientizar o usuário que mesmo que não seja tão vantajoso utilizar um transporte público como meio de preservar o meio ambiente ao invés de um carro movido ao combustível correspondente, também demonstrar que existem outras opções de utilização mais vantajosas, como, por exemplo, o etanol, que em comparação com o diesel ou a gasolina, possui emissão de CO<sub>2</sub> de 2,5 a 4,5 vezes menor.

## PROJETO (ESTRUTURA) DO PROGRAMA

Para construir o programa em Java capaz de sanar as necessidades propostas, com boas práticas de programação e utilizando alguns conceitos importantes apresentados para garantir que as funcionalidades sejam implementadas e demonstrar os conhecimentos vistos em sala de aula, será necessário criar 4 classes e uma interface. Essa interface terá apenas um método que mostrará os resultados para o usuário no terminal, que será implementado na classe “UsaClasses”, responsável por executar todos os métodos de todas as classes, assim como tratar os dados do usuário e manipular atributos de forma devida.

As outras três classes, serão a base de toda a proposta do trabalho. Nelas terão os atributos, os fatores de multiplicação para conversão da queima de combustível em CO2, os métodos que serão invocados na “UsaClasses”, e a grande maioria dos conceitos de LPOO descritos no tópico “Elementos de programação orientada a objetos” deste trabalho.

**Figura 4 – Diagrama de classes do sistema a ser criado**



**Fonte: Autoral**

Este diagrama acima, demonstra as classes, relação entre elas, seus métodos, atributos e modificadores de acesso e comportamento. Inicialmente é possível verificar que todas as classes têm seus atributos como private (-), sendo necessário gerar os métodos get e set de todos os atributos do programa para utilização deles através da UsaClasses.

Logo em seguida, evidenciasse o uso de herança nesse programa, dado que a classe veículo é uma generalização comum entre a classe Carro e a classe TransportePúblico, já que carro e ônibus são ambos veículos. Dessa forma, utilizando o método construtor com parâmetros para definir os atributos da classe veículos, é possível usar o método construtor das suas classes filhas para setar além do atributo delas próprias, os atributos da classe mãe, através do comando super, que indicia superclasse. Com isso, ao instanciar os objetos Carro e ônibus, todos os atributos serão definidos dado os parâmetros definidos pelo usuário como entrada de dados.

Quanto aos modificadores de comportamento dos elementos das classes, a classe veículo possui o modificador abstract, tornando-a abstrata. Esse recurso serve para deixar a codificação padronizada e organizada, pois, já que a classe Veículo é uma superclasse, então, seus elementos apenas podem ser acessados através de suas classes filhas.

Como a classe é abstrata, o intuitivo é que se utilize também métodos abstratos dentro dela, para que seja acessado através apenas de suas classes filhas. No entanto, como forma de aplicação dos conhecimentos adquiridos em sala de aula, nesse trabalho será criado métodos estáticos na classe veículos, como forma de aplicação dos importantes conceitos de LPOO e para tornar mais fácil a utilização desses métodos.

Para utilizar os modificadores de comportamento de forma assertiva, também foi declarado as classes filhas como finais, com o comando final. Pois elas não terão outras classes que herdarão seus elementos gerando uma herança encadeada, com isso, não serão instanciadas e conterão o comando final.

Por final, a interface MostrarDados, serve para organizar as saídas de dados para o usuário no terminal, após o processamento da entrada de dados dada pelo usuário. Dessa forma, a UsaClasses que a implementará, será obrigada a construir o método mostrarDados():void, que será invocado no final do programa.

A tela do programa está funcionando da seguinte forma, primeiramente para se o usuário escolhe utilizar gasolina como parâmetro:

**Figura 5 – Tela de resultado Gasolina**

```
run:
Digite a distância diária percorrida em Km abaixo(Ida e Volta):
10,5
Digite quantas vezes por semana realiza esse trajeto:
5
Digite a eficiência do motor em Km/Litro abaixo :
9,38
Escolha o tipo de combustível utilizado:
[1] Gasolina
[2] Etanol
1
Digite quantas pessoas vão embarcadas durante seu trajeto:
2
A estimativa de sua emissão média de CO2 anual por pessoa é: 306 Kg/CO2
A estimativa de emissão média de CO2 anual por pessoa em um ônibus com média de 30 pessoas é: 95 Kg/CO2
BUILD SUCCESSFUL (total time: 36 seconds)
```

**Fonte:** Autoral

Dessa vez, para se o usuário deseja utilizar etanol como parâmetro para cálculo:

**Figura 6 – Tela resultado Etanol**

```
run:
Digite a distância diária percorrida em Km abaixo(Ida e Volta):
10,5
Digite quantas vezes por semana realiza esse trajeto:
5
Digite a eficiência do motor em Km/Litro abaixo :
9,38
Escolha o tipo de combustível utilizado:
[1] Gasolina
[2] Etanol
2
Digite quantas pessoas vão embarcadas durante seu trajeto:
2
A estimativa de sua emissão média de CO2 anual por pessoa é: 82 Kg/CO2
A estimativa de emissão média de CO2 anual por pessoa em um ônibus com média de 30 pessoas é: 95 Kg/CO2
BUILD SUCCESSFUL (total time: 13 seconds)
```

**Fonte:** Autoral

## RELATÓRIO COM AS LINHAS DE CÓDIGO

Abaixo estarão imagens com as linhas de código do programa para visualização e compreensão do seu funcionamento interno de acordo com o algoritmo.

Classe-Carro

**Figura 7 – Classe Carro**

```
public final class Carro extends Veiculo {  
    private double emissãoCarro; // em Kg/L  
    private final double fatGasolina = 2.2755; //porcentagem de gasolina pura*densidade*fator de transformação em co2  
    private final double fatEtanol = 0.5185; //densidade*fator de transformação em co2  
  
    public Carro(double kmDiária, double emissãoAnual, double emissãoCarro) {  
        super(kmDiária, emissãoAnual);  
        this.setEmissãoCarro(emissãoCarro);  
    }  
  
    public double getFatGasolina() {  
        return fatGasolina;  
    }  
  
    public double getFatEtanol() {  
        return fatEtanol;  
    }  
  
    public double getEmissãoCarro() {  
        return emissãoCarro;  
    }  
  
    public void setEmissãoCarro(double emissãoCarro) {  
        this.emissãoCarro = emissãoCarro;  
    }  
}
```

**Fonte:** Autoral

## Classe-TransportePúblico

**Figura 8 – Classe TransportePúblico**

```
public final class TransportePúblico {  
    private double emissãoDiesel; //em Kg/L  
    private final double fatDiesel = 2.89; //densidade*fator de transformação em co2  
  
    public TransportePúblico(double emissãoDiesel) {  
        this.setEmissãoDiesel(emissãoDiesel);  
    }  
  
    public double getEmissãoDiesel() {  
        return emissãoDiesel;  
    }  
  
    public void setEmissãoDiesel(double emissãoDiesel) {  
        this.emissãoDiesel = emissãoDiesel;  
    }  
  
    public double getFatDiesel() {  
        return fatDiesel;  
    }  
}
```

**Fonte:** Autoral

## Classe-Veículo

**Figura 9 – Classe Veículo**

```
public abstract class Veículo {  
    private double kmDiária, emissãoAnual;  
  
    public Veículo(double kmDiária, double emissãoAnual){  
        this.setKmDiária(kmDiária);  
        this.setEmissãoAnual(emissãoAnual);  
    }  
  
    public static double distânciaAnual(double kmDiária, double semanal){  
        double resultado = kmDiária * semanal * 4 * 12;  
        return resultado;  
    }  
  
    public static double litroAnual(double kmAnual, double eficienciaMotor){  
        double resultado = kmAnual / eficienciaMotor;  
        return resultado;  
    }  
  
    public static double calculoFinal(double litroAnual, double fatMulti){  
        double resultado = litroAnual * fatMulti;  
        return resultado;  
    }  
  
    public double getKmDiária() {  
        return kmDiária;  
    }  
}
```

**Fonte:** Autoral

**Figura 10 – Classe Veículo .2**

```
public void setKmDiária(double kmDiária) {
    this.kmDiária = kmDiária;
}

public double getEmissãoAnual() {
    return emissãoAnual;
}

public void setEmissãoAnual(double emissãoAnual) {
    this.emissãoAnual = emissãoAnual;
}
}
```

**Fonte:** Autoral

## Classe-UsaClasses

**Figura 11 – Classe UsaClasses**

```
import java.util.ArrayList;
import java.util.Scanner;

public class UsaClasses implements MostrarDados{

    public static void main(String[] args) {
        double fatMulti = 0;
        Scanner leia = new Scanner(System.in);
        ArrayList<Double> guardar = new ArrayList();

        System.out.println("Digite a distância diária percorrida em Km abaixo(Ida e Volta): ");
        guardar.add(leia.nextDouble()); //guardar[0]
        System.out.println("Digite quantas vezes por semana realiza esse trajeto:");
        guardar.add(leia.nextDouble()); //guardar[1]
        System.out.println("Digite a eficiência do motor em Km/Litro abaixo :");
        guardar.add(leia.nextDouble()); //guardar[2]

        Carro gol = new Carro(guardar.get(0),Veiculo.distânciaAnual(guardar.get(0),guardar.get(1)),2.21);
        TransportePúblico ônibus = new TransportePúblico(3.41);
    }
}
```

**Fonte:** Autoral

**Figura 12 – Classe UsaClasses.2**

```
System.out.println("Escolha o tipo de combustivel utilizado:");
System.out.println("[1] Gasolina ");
System.out.println("[2] Etanol ");
int escolha = leia.nextInt();
if (escolha == 1){
    fatMulti = gol.getFatGasolina();
}else
    if (escolha == 2){
        fatMulti = gol.getFatEtanol();
    }

System.out.println("Digite quantas pessoas vão embarcadas durante seu trajeto: ");
guardar.add(leia.nextDouble()); //guardar[3]
```

**Fonte:** Autoral

**Figura 13 – ClasseUsaClasses.3**

```
gol.setEmissaoCarro(Veiculo.calculoFinal(Veiculo.litroAnual(Veiculo.distanciaAnual(guardar.get(0), guardar.get(1)), guardar.get(2)), fatMulti));
System.out.println("A estimativa de sua emissão média de CO2 anual por pessoa é: " + Math.round(gol.getEmissaoCarro()/guardar.get(3)) + " Kg/CO2");

ônibus.setEmissaoDiesel(Veiculo.calculoFinal(Veiculo.litroAnual(Veiculo.distanciaAnual(guardar.get(0), guardar.get(1)), 2.567), ônibus.getFatDiesel()));
System.out.println("A estimativa de emissão média de CO2 anual por pessoa em um ônibus com média de 30 pessoas é: " + Math.round(ônibus.getEmissaoDiesel()/30) + " Kg/CO2")

}

@Override
public void imprimirDados() {

}

}
```

**Fonte:** Autoral



## BIBLIOGRAFIA

<https://educacao.uol.com.br/disciplinas/quimica/quimica-do-automovel-1-combustao-da-gasolina-e-do-alcool.htm>

<https://brainly.com.br/tarefa/873078>

[http://esalqlastrop.com.br/capa.asp?pi=calculadora\\_emissoes](http://esalqlastrop.com.br/capa.asp?pi=calculadora_emissoes)

<https://summitmobilidade.estadao.com.br/ir-e-vir-no-mundo/automoveis-sao-a-principal-fonte-de-emissao-de-gases-poluentes/#:~:text=e%20na%20vegeta%C3%A7%C3%A3o.-,Sa%C3%BAde,a%20sa%C3%BAde%20em%20longo%20prazo.>

<https://mundoeducacao.uol.com.br/quimica/combustiveis.htm>

<https://www.ecodebate.com.br/2012/08/14/os-impactos-do-automovel-no-meio-ambiente/>

<https://www.educamaisbrasil.com.br/enem/geografia/destruicao-da-camada-de-ozoni>

<https://wribrasil.org.br/pt/blog/2019/06/5-impactos-pouco-conhecidos-da-poluicao-do-ar-atmosferica>

[http://www.cbcs.org.br/sbcs10/website/userFiles/palestras\\_sbcs\\_10/emissao\\_co2\\_transporte.pdf](http://www.cbcs.org.br/sbcs10/website/userFiles/palestras_sbcs_10/emissao_co2_transporte.pdf)