

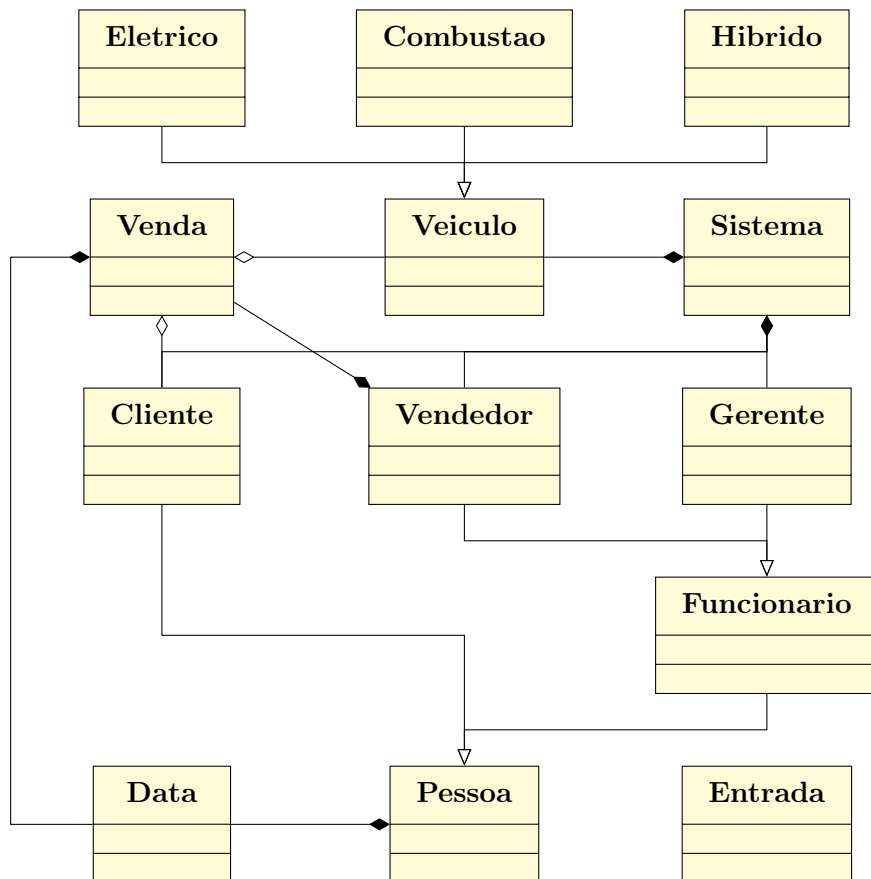
Atividade 2: Implementação

POO - BSI - Ifes Serra

8 de outubro de 2025

1 Diagrama de Classes UML

Nesta atividade, você deve começar a implementar o sistema feito na Atividade 1. As classes do sistema e as relações entre elas são mostradas a seguir:



Note que há uma nova classe chamada **Entrada**, que vai cuidar de toda a leitura de dados do sistema e interação com o usuário do sistema. **Nenhuma leitura de dados pode ser feita fora da classe Entrada**. A seguir, uma descrição detalhada dos atributos e métodos que devem ser implementados em cada classe por enquanto:

Entrada
- input : Scanner
- lerLinha(msg : String) : String - lerInteiro(msg : String) : int - lerDouble(msg : String) : double + menu(s : Sistema) : int + cadCliente(s : Sistema) + cadVendedor(s : Sistema) + cadGerente(s : Sistema) + cadVeiculo(s : Sistema) + cadVenda(s : Sistema) - relatorioMensal(s : Sistema) - relatorioAnual(s : Sistema) - relatorioVendedor(s : Sistema)

Alguns métodos já foram implementados no arquivo *Entrada.java* disponibilizado. Há um método com o *menu* do sistema:

Escolha uma opção:

- 1) Cadastrar Cliente
- 2) Cadastrar Vendedor
- 3) Cadastrar Gerente
- 4) Cadastrar Veículo
- 5) Cadastrar Venda
- 6) Relatório de Vendas Mensal
- 7) Relatório de Vendas Anual
- 8) Relatório de Vendas do Vendedor:
- 0) Sair

Os métodos de cadastro leem as informações apropriadas sobre cada novo objeto, **nesta ordem**:

1. Para clientes:

```
Digite o nome do cliente: Hilario Seibel Junior
Digite o cpf do cliente: 123
Digite o dia do nascimento do cliente: 4
Digite o mês do nascimento do cliente: 12
Digite o ano do nascimento do cliente: 1981
Digite o email do cliente: hsjunior@gmail.com
```

2. Para vendedores:

```
Digite o nome do vendedor: Maria da Silva
Digite o cpf do vendedor: 345
Digite o dia do nascimento do vendedor: 3
Digite o mês do nascimento do vendedor: 3
Digite o ano do nascimento do vendedor: 2003
Digite o salário mensal fixo do vendedor: 3000
Digite o percentual de comissão deste vendedor: 0.2
```

3. Para gerentes:

Digite o nome do gerente: Pedro da Silva
Digite o cpf do gerente: 456
Digite o dia do nascimento do gerente: 4
Digite o mês do nascimento do gerente: 4
Digite o ano do nascimento do gerente: 2004
Digite o salário mensal fixo do gerente: 4000
Digite a senha do gerente: 456

4. Para veículos:

Digite a Marca do veículo: BYD
Digite a Marca do veículo: Song Pro
Digite o ano de fabricação do veículo: 2025
Digite o mês de fabricação do veículo: 7
Digite o ano do modelo do veículo: 2026
Digite o valor do veículo: 199000
Escolha o tipo do veículo:
1) Elétrico
2) Combustão
3) Híbrido:
3
Digite a autonomia do motor (em km): 1000
Digite a capacidade do motor (em L): 50
Digite a autonomia da bateria (em km): 110
Digite a capacidade da bateria (em kWh): 18

5. Para vendas:

Vendedores cadastrados:
Maria da Silva - CPF: 345

Digite o CPF do vendedor: 345

Veiculos cadastrados:
1) BYD Song Pro 2025/2026 - Autonomia: 1100.0km (Híbrido)

Escolha um veículo pelo número: 1

Clientes cadastrados:
Hilario Seibel Junior - CPF: 123 - hsjunior@gmail.com
Wagner Kirmse - CPF: 234 - wagnerkc@gmail.com

Digite o CPF do cliente: 123

Digite o desconto (em R\$): 5000
Digite o dia da venda: 5
Digite o mês da venda: 9
Digite o ano da venda: 2025
Digite o chassi do veículo: 12345

O relatório mensal lê o mês e o ano, e imprime os dados das vendas neste mês do ano, no seguinte formato:

```
*** RELATÓRIO DE VENDAS MENSAL DE 9/2025 ***
Vendedor: Maria da Silva (Salário neste mês: RS3398.0)
Veiculo: BYD Song Pro 2025/2026 - Autonomia: 1110.0km (Híbrido)
Cliente: Hilario Seibel Junior - CPF: 123 - hsjunior@gmail.com
Valor da venda: R$194000.0
Data: 5/9/2025
*****
Total: R$194000.0
```

Note que no final do relatório há o valor total arrecadado pela concessionária neste mês do ano. Além disso, cada venda deve exibir (ao lado do nome do vendedor responsável pela venda) o salário do vendedor neste mês do ano.

O relatório anual lê o ano, e imprime os dados das vendas neste ano, no seguinte formato:

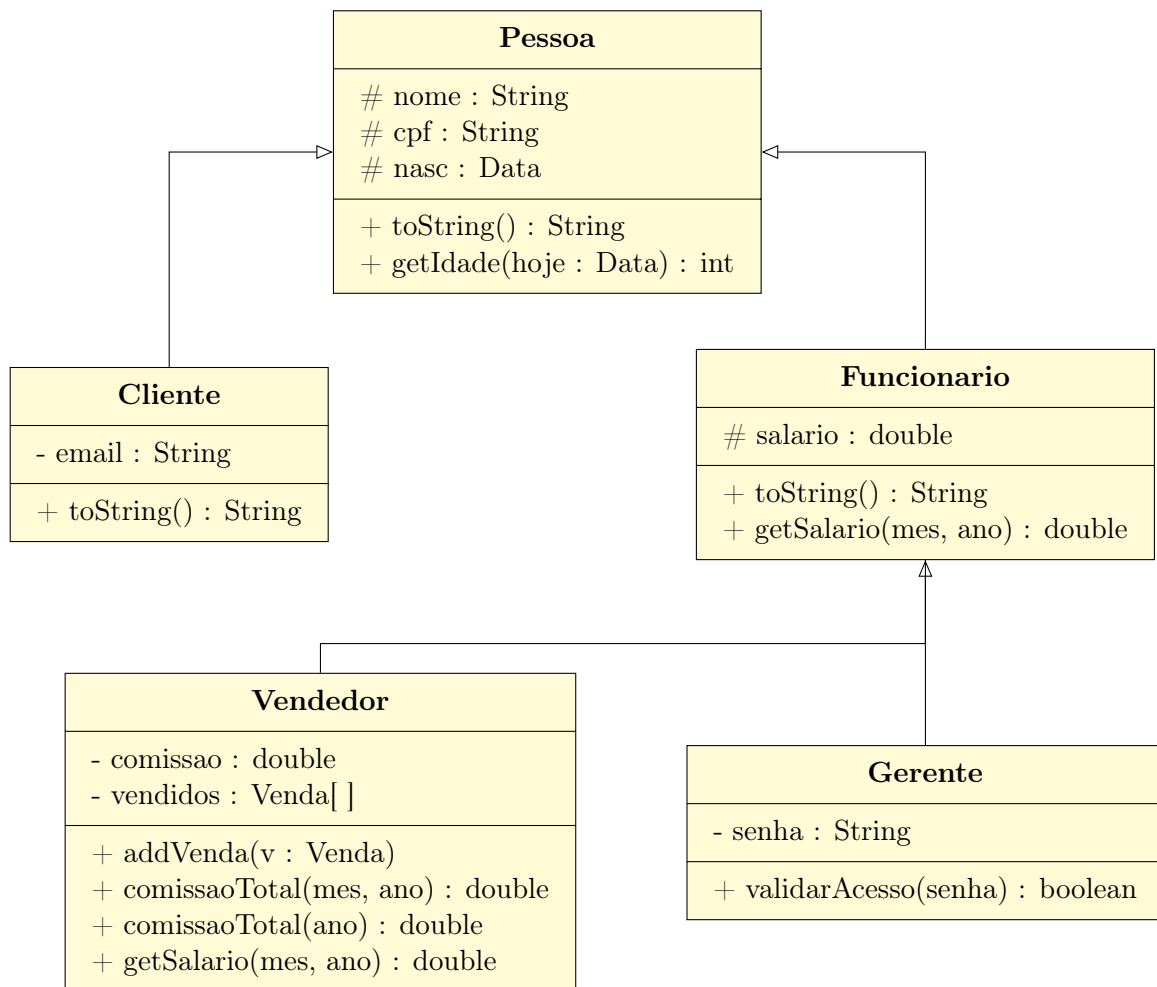
```
Vendedor: Maria da Silva
Veiculo: BYD Song Pro 2025/2026 - Autonomia: 1110.0km (Híbrido)
Cliente: Hilario Seibel Junior - CPF: 123 - hsjunior@gmail.com
Valor da venda: R$194000.0
Data: 5/9/2025
*****
Vendedor: Maria da Silva
Veiculo: Honda HRV 2015/2016 - Autonomia: 400.0km (Combustão)
Cliente: Wagner Kirmse - CPF: 234 - wagnerkc@gmail.com
Valor da venda: R$67500.0
Data: 10/10/2025
*****
Total: R$261500.0
```

Note que no final do relatório há o valor total arrecadado pela concessionária neste ano.

Analogamente, o relatório por vendedor lê o cpf de um vendedor e exibe todas as duas vendas:

```
*** RELATÓRIO DE VENDAS DO VENDEADOR ***
Vendas do vendedor Maria da Silva :
Veiculo: BYD Song Pro 2025/2026 - Autonomia: 1110.0km (Híbrido)
Cliente: Hilario Seibel Junior - CPF: 123 - hsjunior@gmail.com
Valor da venda: R$194000.0
Data: 5/9/2025
*****
Veiculo: Honda HRV 2015/2016 - Autonomia: 400.0km (Combustão)
Cliente: Wagner Kirmse - CPF: 234 - wagnerkc@gmail.com
Valor da venda: R$67500.0
Data: 10/10/2025
*****
Total: R$261500.0
```

O diagrama abaixo mostra como os clientes e funcionários serão modelados:



O método **toString** em uma classe qualquer indica como um objeto desta classe será representado quando for impresso. Na classe **Pessoa**, por exemplo, caso o nome seja “João da Silva” e seu CPF seja “123.456.789-00”, o método deve retornar a String “João da Silva - CPF: 123.456.789-00”. Fazendo isso, supondo que o nome deste objeto é p1, isso significa que a impressão de p1:

```
System.out.println(p1);
```

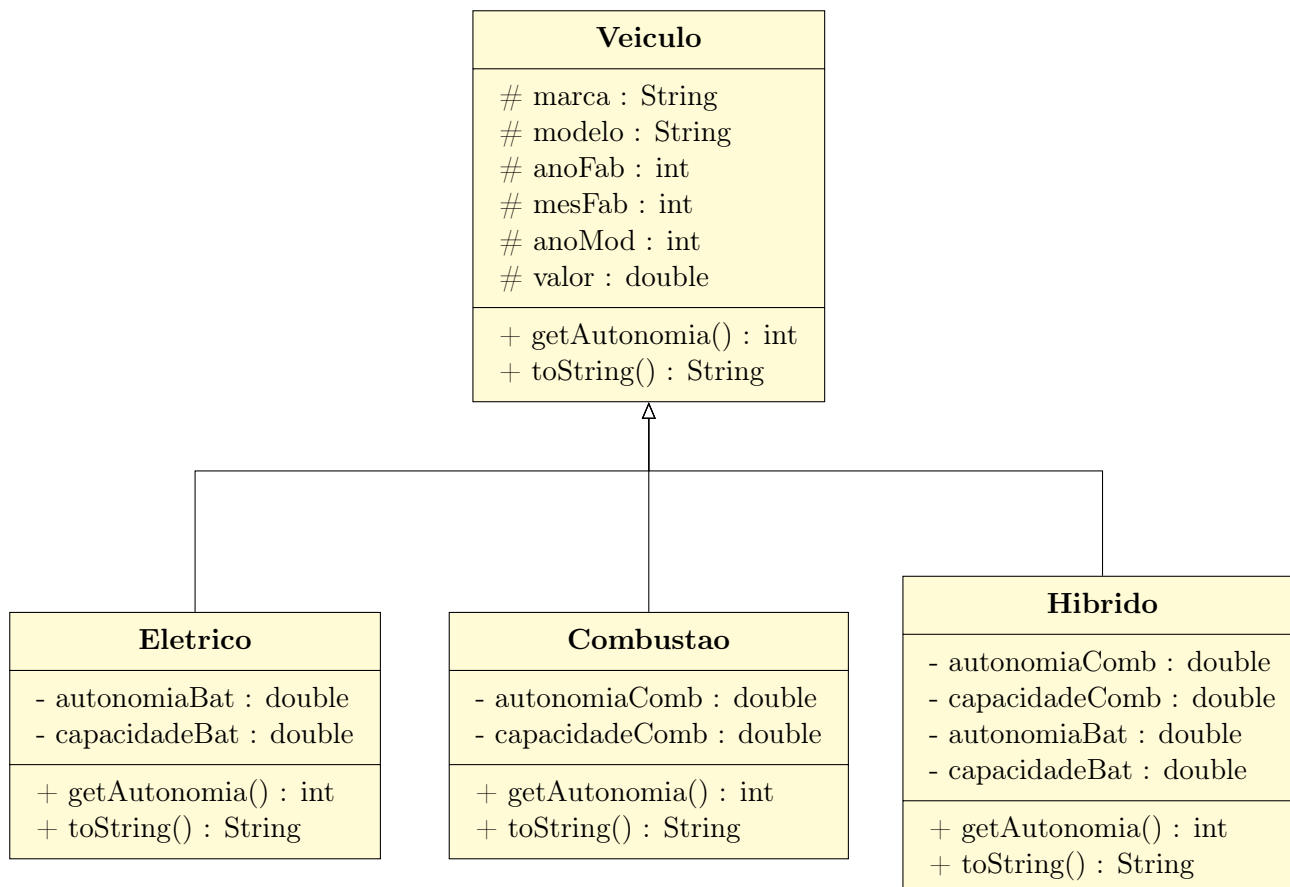
irá exibir na tela:

João da Silva - CPF: 123.456.789-00

Na classe **CLiente**, o método **toString** deverá ser *reescrito* para exibir também seu email:

Wagner Kirmse - CPF: 234 - wagnerkc@gmail.com

Já os veículos serão representadas a partir do diagrama a seguir:



O método **getAutonomia** retorna a soma total de todas as autonomias de cada tipo de veículo (por enquanto, na classe **Veiculo**, ele vai retornar apenas zero).

O método **toString** da classe **Veiculo** vai exibir:

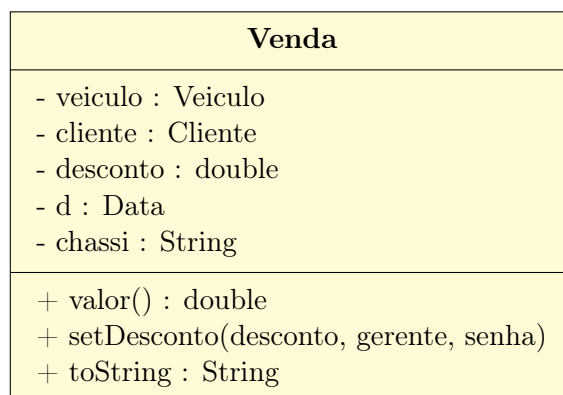
Honda HRV 2015/2016 - Autonomia: 400.0km

Em cada subclasse, ele vai ser reescrito para adicionar ao fim da String o tipo de veículo:

Honda HRV 2015/2016 - Autonomia: 400.0km (Combustão)

BYD Song Pro 2025/2026 - Autonomia: 1110.0km (Híbrido)

Toda **Venda** possui um veículo, um cliente, um valor de desconto em reais, uma data e o chassi do veículo vendido:



O método **valor()** retorna o valor arrecadado pela concessionária nesta venda (valor total do veículo menos o desconto dado pelo gerente).

O método **setDesconto()** altera o desconto desde que o gerente valide seu acesso e permita o desconto.

O método **toString** da classe **Venda** vai exibir:

Veiculo: BYD Song Pro 2025/2026 - Autonomia: 1110.0km (Híbrido)
Cliente: Hilario Seibel Junior - CPF: 123 - hsjunior@gmail.com
Valor da venda: R\$194000.0
Data: 5/9/2025

Por fim, a classe Sistema armazena todos os clientes, gerentes, vendedores e veículos cadastrados no Sistema Acadêmico:

Sistema
- veiculos : Veiculo[] - vendedores : Vendedor[] - gerentes : Gerente[] - clientes : cliente[]
+ adicionar(veiculo) + adicionar(vendedor) + adicionar(gerente) + adicionar(cliente) + listarClientes() + listarVendedores() + listarGerentes() + listarVeiculos() + localizarCliente(cpf) : Cliente + localizarGerente(cpf) : Gerente + localizarVendedor(cpf) : Vendedor + atribuirVendaVendedor(venda, vendedor) + relatorio(mes, ano) + relatorio(ano) + relatorio(vendedor) + listarTurmas()

O método adicionar, dependendo do tipo do parâmetro, adiciona um vendedor, cliente, gerente ou veículo a seus respectivos conjuntos.

O método **atribuirVendaVendedor** efetivamente adiciona uma venda ao conjunto de vendas de um vendedor.

Se necessário, crie métodos **getters** para acessar o valor de algum atributo que esteja encapsulado nas classes. Para cada conjunto de objetos no diagrama, o aluno deve decidir se é mais vantajoso implementá-lo usando array ou ArrayList. Apenas estas duas estruturas são permitidas nesta atividade.

É importante ler os dados de cadastros e opções do menu na ordem em que foram descritos neste documento, pois a correção será automatizada. Na página da disciplina, há um arquivo “input.txt” com um exemplo de entrada do arquivo. Basta rodar seu programa com esse arquivo na pasta onde o programa está sendo executado, e o arquivo será lido automaticamente (ao invés do programa ler a entrada do teclado). Faça isso apenas quando todas as opções do seu menu já estiverem funcionando.

2 Observações

1. A atividade vale 40 pontos e deve ser entregue até 21/10.
2. O código deve ser feito em Java, considerando as boas práticas de POO (como reuso de código, por exemplo, sempre que possível).

3. A atividade pode ser feita em grupos de até dois integrantes.
4. Atividades entregues após o prazo serão automaticamente rejeitados.
5. Atividades com erro de execução, com formato de saída incorreto, ou que não compilarem terão nota 0.
6. Atividades considerados plágio terão nota 0 para quem copiou e para quem forneceu o exercício, e serão enviados ao Conselho de Ética.
7. A Atividade deve ser enviado na sala da disciplina do AVA.
8. Em caso de dúvidas na especificação da atividade ou na própria atividade, contate-me em sala de aula.