



**Attribution-NonCommercial-
NoDerivatives 4.0 International
(CC BY-NC-ND 4.0)**



Este trabalho está licenciado com uma Licença [Creative Commons -
Atribuição-NãoComercial-SemDerivações 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Programação Orientada a Objetos

2

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

PROF. LUCIO AGOSTINHO ROCHA

AULA 22: PROJETO DE SOFTWARE

1º.SEMESTRE 2023

Unified Modeling Language

Unified Modeling Language

4

- UML (Unified Modeling Language):
 - É uma linguagem de modelagem padronizada e formal para descrever sistemas orientados a objetos (Priestley 2000, Larman 2002).
 - UML fornece modelos e notações formais para a documentação e apresentação dos relacionamentos entre as partes do sistema.
 - Diagramas UML são inter-relacionados e identificam etapas do desenvolvimento do projeto com detalhes que são relevantes para a etapa atual do projeto.
 - Projetos bem elaborados mantêm diagramas padronizados das principais partes do sistema para documentação, consulta, modificação, entendimento e reuso do projeto.

Diagramas UML

Diagramas UML

6

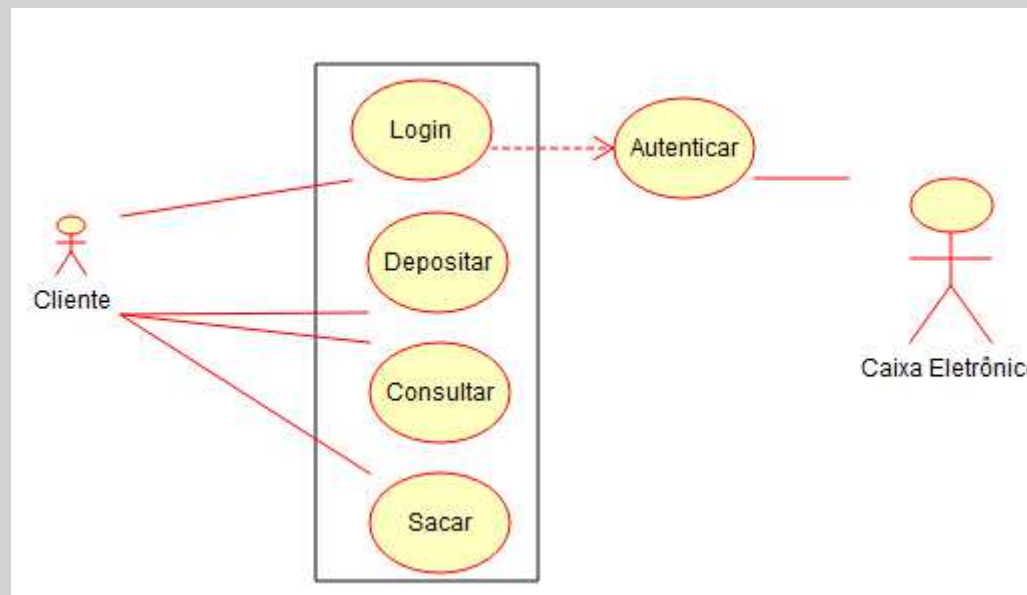
- Diagramas UML (Unified Modeling Language):
 - São representações gráficas formais para documentar, representar e apresentar o relacionamento entre as partes do sistema sem o uso de linguagens de programação.
 - Diagramas UML são inter-relacionados e identificam etapas do desenvolvimento do projeto com detalhes que são relevantes para a etapa atual do desenvolvimento.
 - Projeto bem elaborados mantêm diagramas padronizados das principais partes do sistema para consulta, modificação e entendimento do projeto.

Diagramas UML

7

- Diagrama UML de Casos de Uso

- Mostra uma visão-geral do sistema, com foco nas funcionalidades.
- Mostra os atores (usuários/entidades do sistema)
- Mostra os casos de uso (funcionalidades)
- Mostra os relacionamentos entre os casos de uso e os atores.

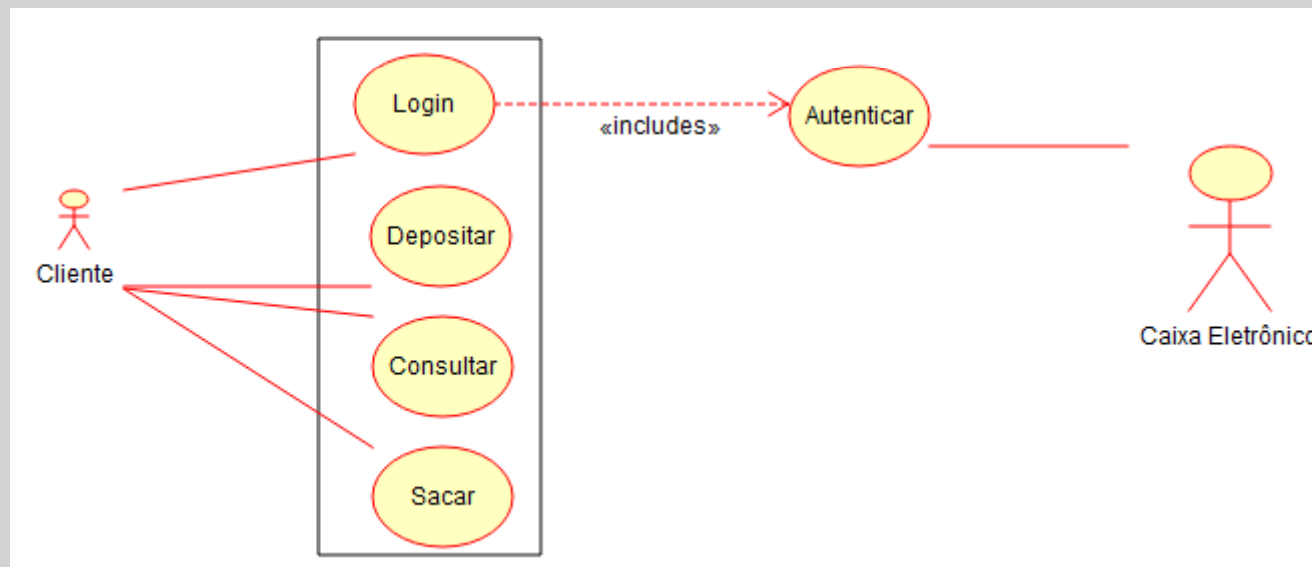


Diagramas UML

8

- Diagrama UML de Casos de Uso

- Include: É uma associação que aproveita a modelagem do caso de uso anterior. Todas as vezes que Login for realizado, a ação Autenticar também será realizada.



Diagramas UML

9

- Diagrama UML de Casos de Uso

- Extends: Toda extensão está associada a uma condição. A ação Emitir Recibo só é realizada SE o Cliente solicitar, após a ação Depositar.

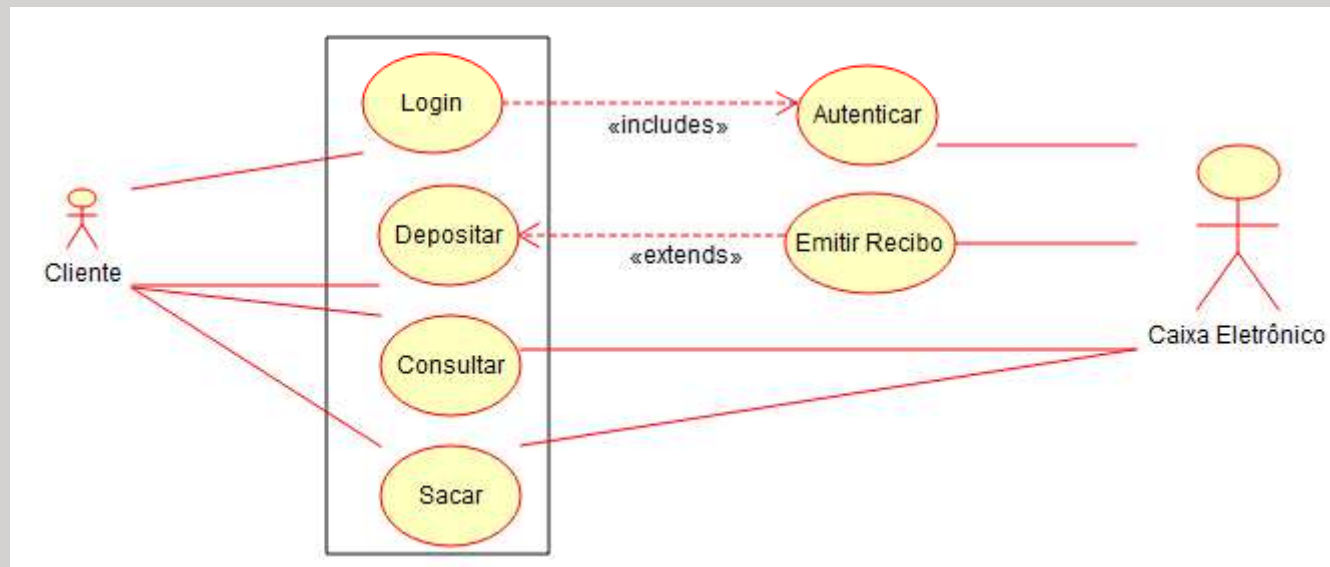


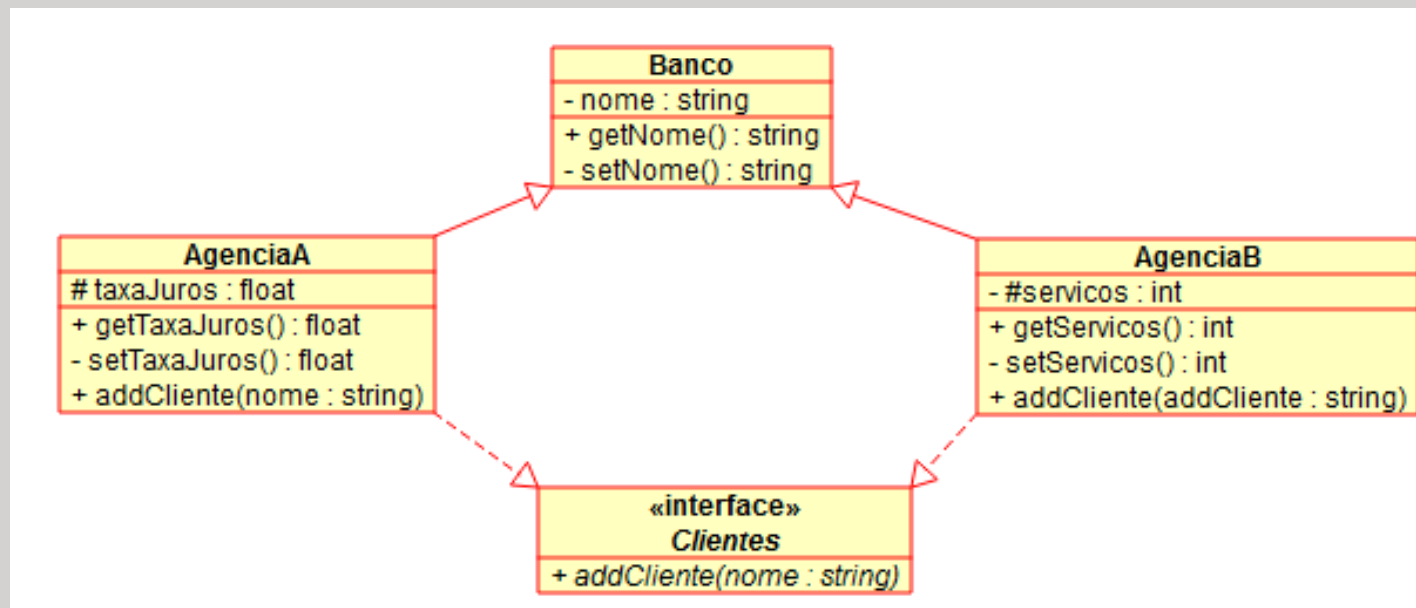
Diagrama de Classes

Diagramas UML

11

- Diagrama UML de Classes

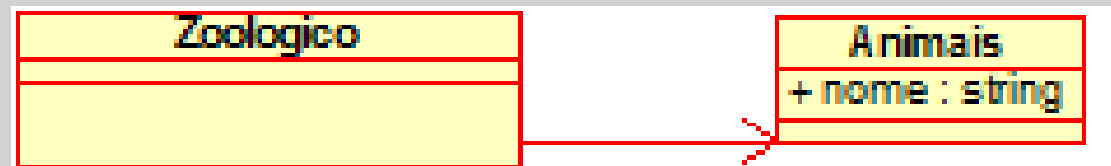
- Mostra as classes e os relacionamentos entre as classes.
- Cada classe possui variáveis privadas (-), públicas (+) e protected(#)
- Cada classe possui métodos privados (-), públicos (+) e protected(#)



Notações em UML

12

- Associação
 - Relação entre duas Classes



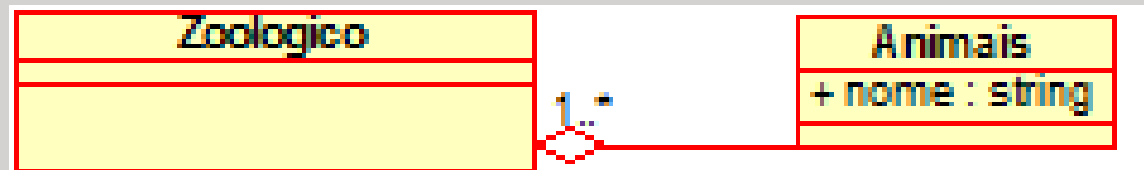
```
public class Zoologico {
    ...
    Animais animal = new Animais( );
    ...
}
```

Notações em UML

13

- **Agregação**

- É uma Associação com mais informações.
- A Classe filha existe independente da Classe pai.
Ex.: Animais existem independente do Zoológico.



```
public class Zoologico {
    ...
    Animais animal = new Animais( );
    ...
}
```

Notações em UML

14

- Composição

- É uma Associação com mais informações.
- Classe filha não existe sem a Classe pai.
Ex.: Portão não existe sem o Zoológico.



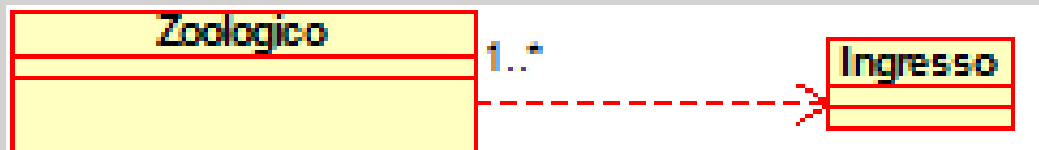
```
public class Zoologico {  
    ...  
    Portão portaoPrincipal = new Portão( );  
    ...  
}
```

Notações em UML

15

- Dependência

- É uma relação mais fraca que a Associação.
- Indica que uma Classe apenas interage com a outra através de métodos. Classes com pouca relação semântica entre si.



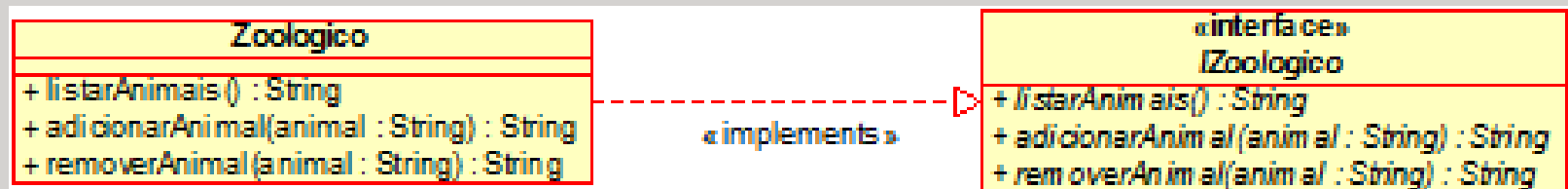
```
public class Zoologico {  
    ...  
    Ingresso.consultaNumero( );  
    public consulta (Ingresso p){  
        ...  
    }  
    ...  
}
```

Notações em UML

16

- Implementação

- Indica a implementação de uma interface.

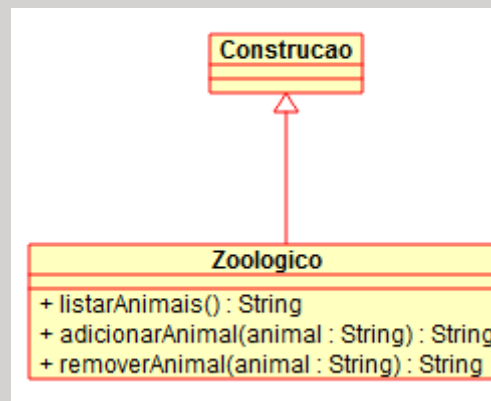


```
public class Zoologico implements IZoologico {  
    ...  
    public String listarAnimais( ){...}  
    public String adicionarAnimal(String animal){...}  
    public String removerAnimal(String animal){...}  
    ...  
}
```


Notações em UML

17

- Generalização
 - Utiliza herança para indicar que uma classe (subclasse) deriva de outra (superclasse).



```
public class Zoológico extends Construção {  
    ...  
}
```

Exercícios

18

<Ver conteúdo na plataforma de ensino>



Referências

19

- Referências bibliográficas da disciplina.
- JACOBSEN, I; BOOCH, G.; RUMBAUGH, J. *The Unified Software Development Process*. Addison-Wesley, 1999
- LARMAN, C. *Applying UML and Patterns*. Prentice Hall, 2002
- RUMIANCEV, P. UML Class Diagram Arrows Guide. Disponível em: <https://medium.com/the-innovation/uml-class-diagram-arrows-guide-37e4b1bb11e>. Acessado em Maio de 2021.
- <https://stackoverflow.com/questions/1874049/explanation-of-the-uml-arrows/23256583> (Gráficos de referência para projetos)
- UML Class Diagrams Reference. Disponível em <https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2015/modeling/uml-class-diagrams-reference?view=vs-2015>. Acessado em Maio de 2021.