UNINASSAU

ENGENHARIA DE SOFTWARES

Curso: Análise e Desenvolvimento de Sistemas
Prof. Me Enoch Menezes de Oliveira Junior

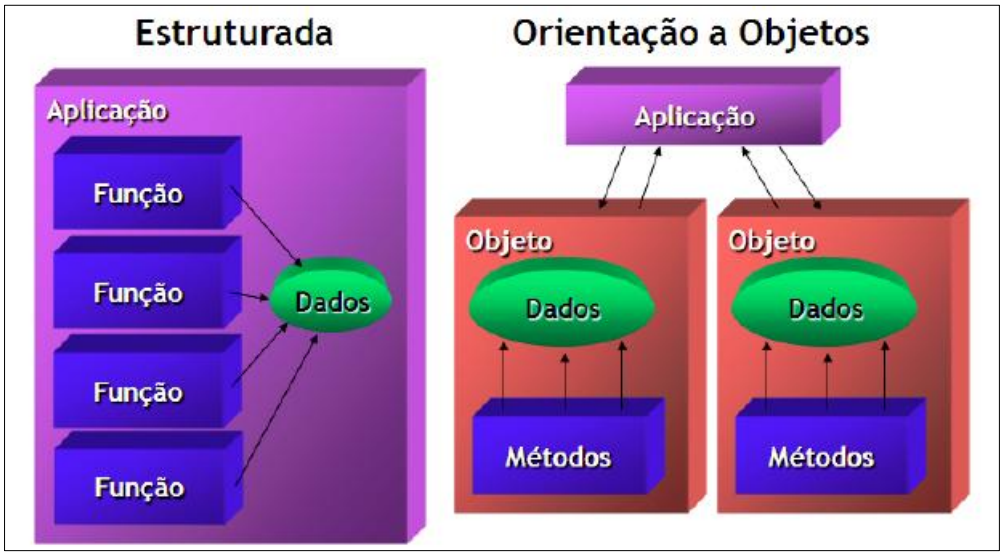
1

UNINASSAU

UML

2

UNINASSAU

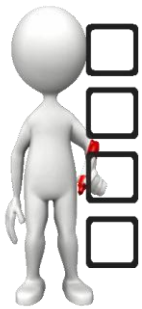


3

UNINASSAU

Orientação à objetos

Programação Estruturada



- **Base:**
  - ✓ **Sequência**: Uma tarefa é executada após a outra, linearmente.
  - ✓ **Decisão**: A partir de um teste lógico, determinado trecho de código é executado, ou não.
  - ✓ **Iteração**: A partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes.
- **Vantagens:**
  - ✓ É fácil de entender. Ainda muito usada em cursos introdutórios de programação.
  - ✓ Execução mais rápida.
- **Desvantagens:**
  - ✓ Baixa reutilização de código
  - ✓ Códigos confusos: Dados misturados com comportamento.

4

UNINASSAU

Orientação à objetos

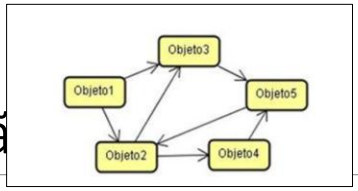
Programação Orientada à objetos

- **Base:**
  - ✓Classes e Objetos.
  - ✓Métodos e Atributos.
- **Vantagens:**
  - ✓Melhor organização do código.
  - ✓Bom reaproveitamento de código.
- **Desvantagens:**
  - ✓Desempenho mais baixo que o paradigma estruturado.
  - ✓Mais difícil compreensão.

5

UNINASSAU

Retrospectiva Orientaçã



Programação Orientada a Objetos

Paradigma de Programação.

Dominante nos dias atuais.

Substituiu as técnicas de programação procedimental (estruturada).

“Fornece um mapeamento direto entre o mundo real e as unidades de organização utilizadas no projeto”.

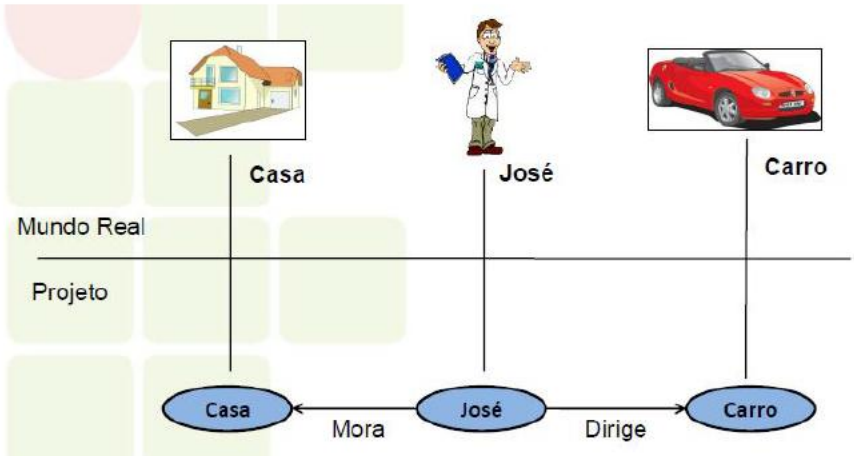
Diversas unidades de software, chamadas de objetos, que interagem entre si.

Separa claramente a noção de o que é feito de como é feito.

6

UNINASSAU

- Representação:



7

UNINASSAU

Definição de Objetos

- Um objeto é algo do mundo real :
  - Concreto ou Abstrato
- As percepção dos seres humanos é dada através dos objetos
- Um objeto é uma entidade que exibe algum comportamento bem definido.

OBJETO = DADOS + OPERAÇÕES

8

UNINASSAU

Objetos

- Características
  - Dados representam características
    - São chamados **atributos**
    - São as variáveis do objeto
- Comportamento
  - Operações definem comportamento
    - São os **métodos** de um objeto
    - São as funções que são executadas por um objeto

9

UNINASSAU

Objetos - Propriedades

- Estado
  - Representado pelos valores dos atributos de um objeto
- Comportamento
  - Definido pelo conjunto de métodos do objeto
  - Estado representa o resultado cumulativo de seu comportamento
- Identidade
  - Um objeto é único, mesmo que o seu estado seja idêntico ao de outro;
  - Seu valor de referência
- Os valores dos **DADOS** são modificados a partir das **OPERAÇÕES** sobre estes dados

10

UNINASSAU

Objetos - Propriedades

- Estado

Acesa

Apagada

- Comportamento

Acender

Apagar

- Identidade

11

UNINASSAU

Método e Atributos

- Atributos
  - Raça: Poodle
  - Nome: Rex
  - Peso: 5 quilos
- Método
  - Latir
  - Comer
  - Dormir

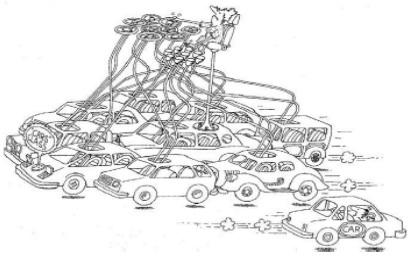
- Atributos
  - Potência: 500cc
  - Modelo: Honda
  - Ano: 1998
- Método
  - Acelerar
  - Frear
  - Abastecer

12

UNINASSAU

Classes

- São especificações para objetos;
- Representam um conjunto de objetos que compartilham características e comportamentos comuns.

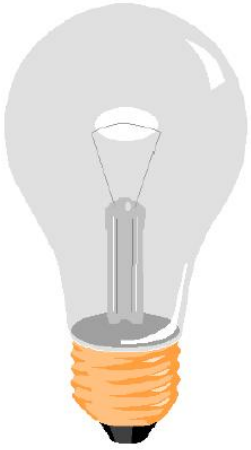


Todo carro tem em comum:
Característica
Cor
Pneu
Direção
Comportamento
Dirigir
Frear

13

UNINASSAU

Classes



Nome da classe

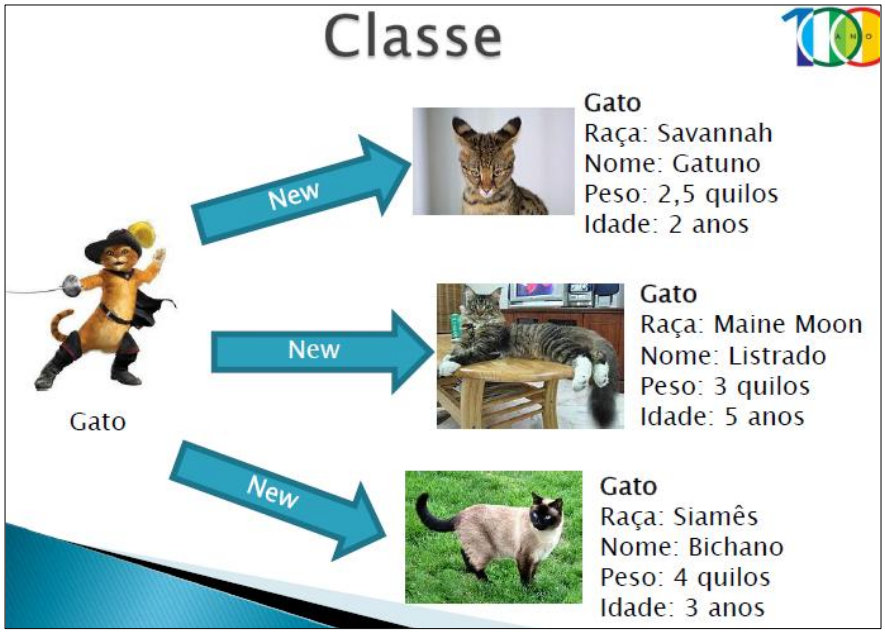
Atributos

métodos

Lampada
- ligada : boolean - potencia : double
+ ligar() : void + desligar() : void + estaLigada() : boolean

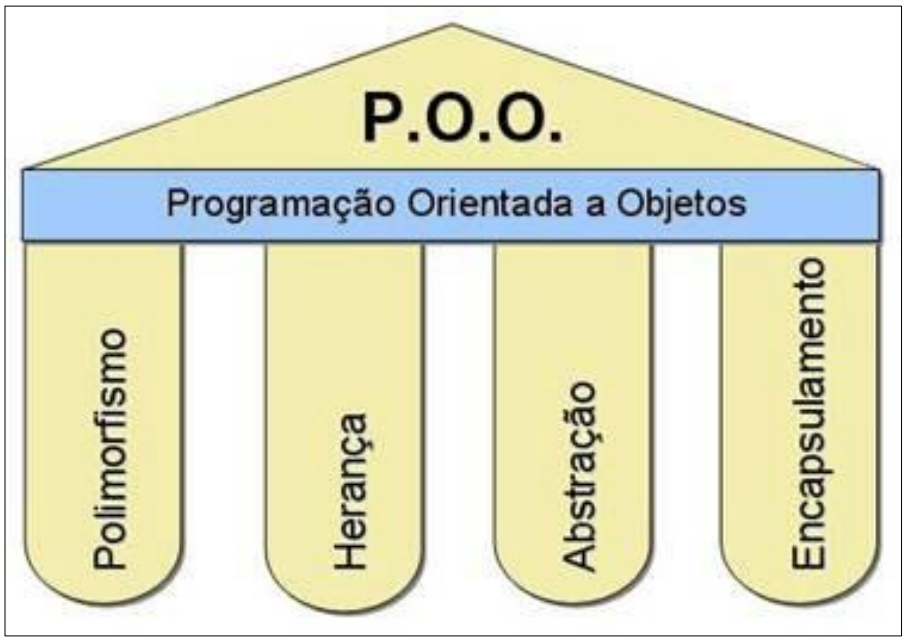
14

UNINASSAU



15

UNINASSAU



16

UNINASSAU

Abstração

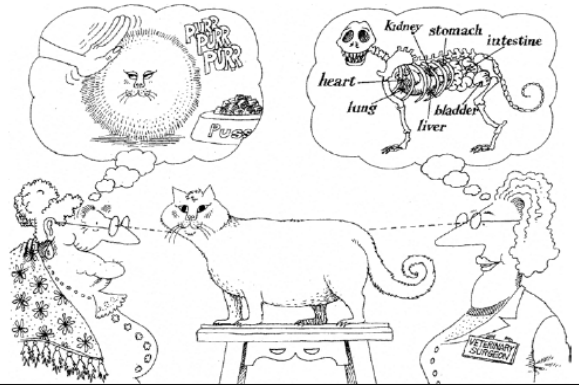
- Abstração é uma das formas fundamentais que nós lidamos com a complexidade.;
- Quando queremos diminuir a complexidade de alguma coisa, ignoramos detalhes sobre as partes para concentrar a atenção no nível mais alto de um problema;
- Não se analisa o "todo", em POO é importante analisar as partes para entender o todo.

17

UNINASSAU

Abstração

- Foca a característica essencial de alguns objetos relativo a perspectiva do visualizador

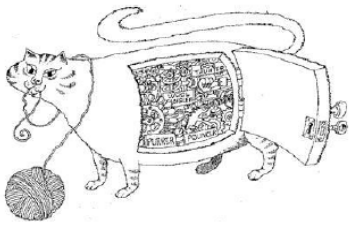


18

UNINASSAU

Encapsulamento

- Encapsulamento é o processo de esconder todos os detalhes de um objeto que não contribuem para as suas características essenciais;
- O encapsulamento é o modo de dar ao objeto seu comportamento "caixa-preta", que é o segredo da reutilização e confiabilidade.



Se o estado de um objeto foi modificado sem uma chamada de método desse objeto, então o encapsulamento foi quebrado

19

UNINASSAU

Visibilidade

- **Private**
  - O nível de acesso se restringe apenas a classe;
  - Não é passado por herança;
- **Public**
  - O nível de acesso é irrestrito;
  - Por padrão, é a visibilidade definida para métodos e atributos em uma classe
- **Protected**
  - É visível em toda a classe;
  - É passado por herança (mesmo em pacotes diferentes);

20

UNINASSAU

Herança

- Herança é o mecanismo para expressar a similaridade entre Classes, simplificando a definição de classes iguais que já foram definidas.
- O que um leão, um tigre, um gato, um lobo e um dálmatas têm em comum?
- Como eles são relacionados?

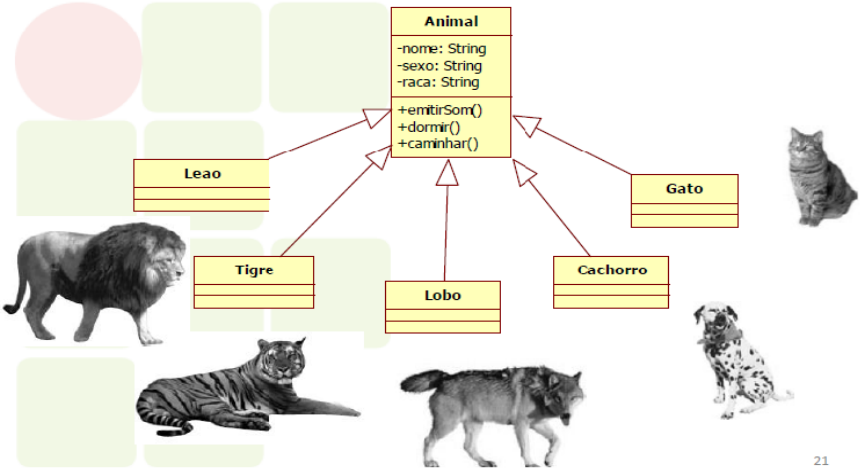




21

UNINASSAU

Herança



22

UNINASSAU

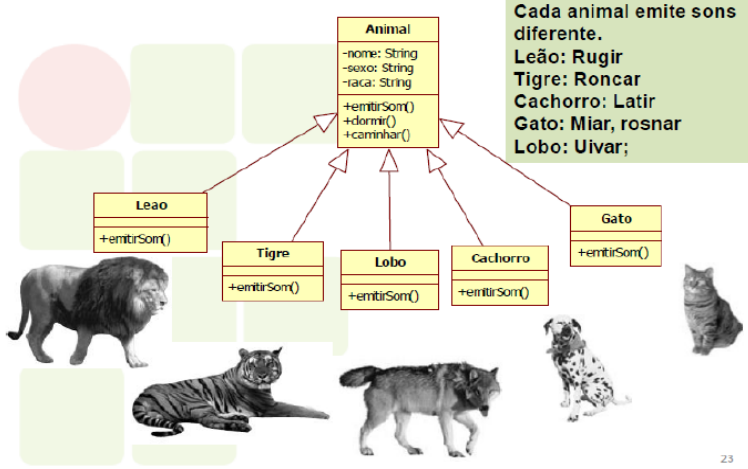
Polimorfismo

- Polimorfimos
  - **Poli** -> varias; **Morfos** -> formas;
- Significa que um objeto pode assumir diferentes formas;
- O conceito de polimorfismo está associado a Herança;
- É caracterizado como o fato de uma operação poder ser implementada de diferentes maneiras pelas classes na hierarquia.

23

UNINASSAU

Polimorfismo



24

UNINASSAU

UML

UNINASSAU

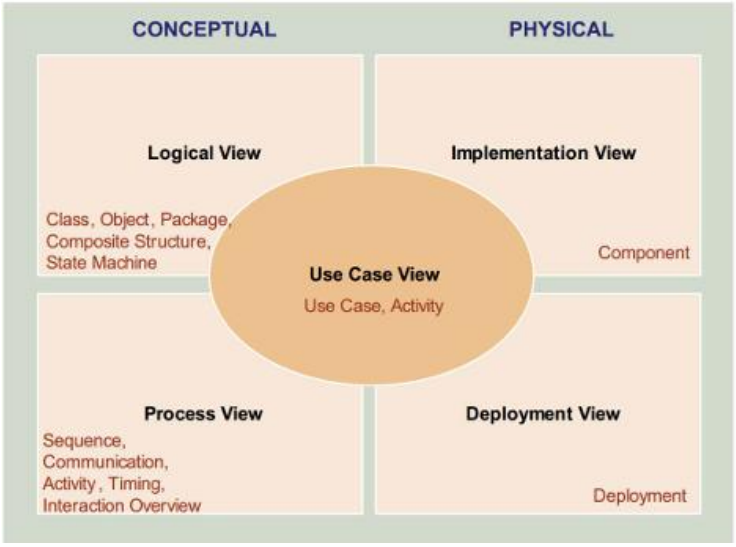
UML

Fases de desenvolvimento:

- **Análise de requisitos.**
- **Análise.**
- **Design (projeto).**
- **Programação.**
- **Testes.**

UNINASSAU

UML



27

UNINASSAU

UML

Cenários (Use Case View ou Scenarios view)

- Apresenta uma visão próxima do usuário, descrevendo cenários de uso da aplicação
- Normalmente é a primeira visão construída
- Diagramas: Casos de Uso

Lógica (Logical View)

- Foco nas funcionalidades, normalmente dividida em subsistemas apresentando a estrutura (classes) envolvidas
- Diagramas: Classe, Pacotes, Componentes

Processos (Process View)

- Foco em aspectos comportamentais
- Diagramas: Atividade, Sequencia, Comunicação

Implementação (Implementation View)

- Útil para gestão de configuração
- Apresenta pacotes e componentes que serão implantados
- Diagramas: Componentes; Pacotes

Implantação (Deployment View)

- Consiste do mapeamento do software no hardware onde será implantado
- Diagramas: Implantação

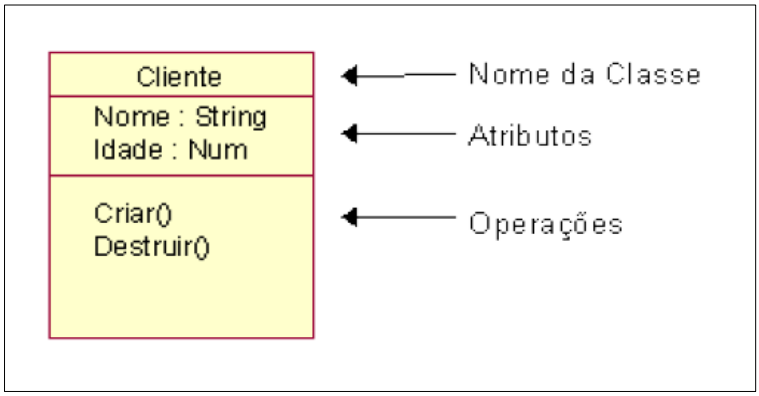
28

UNINASSAU

UML

Modelo de elementos:

Classes:



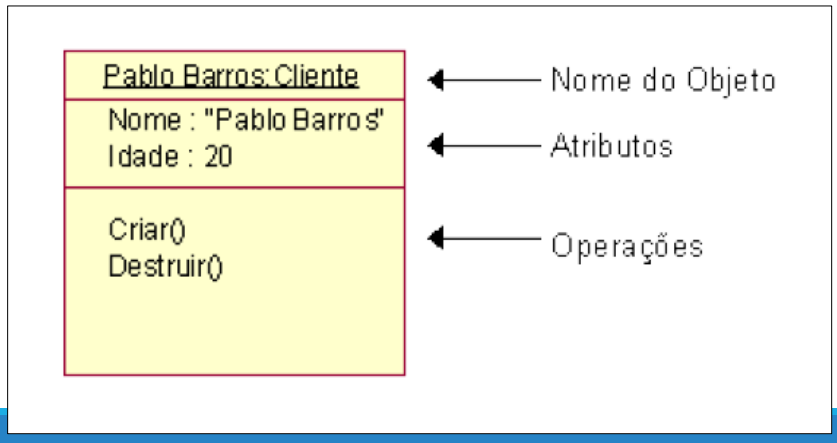
29

UNINASSAU

UML

Modelo de elementos:

Objetos:



30

UNINASSAU

UML

Modelo de elementos:

Estados:

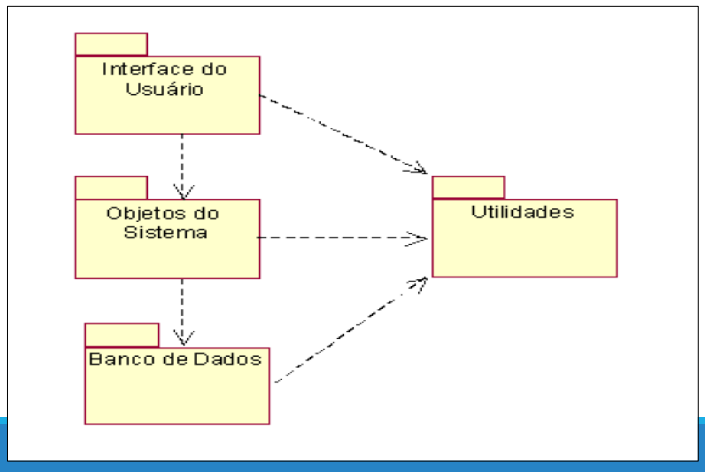
31

UNINASSAU

UML

Modelo de elementos:

Pacotes:



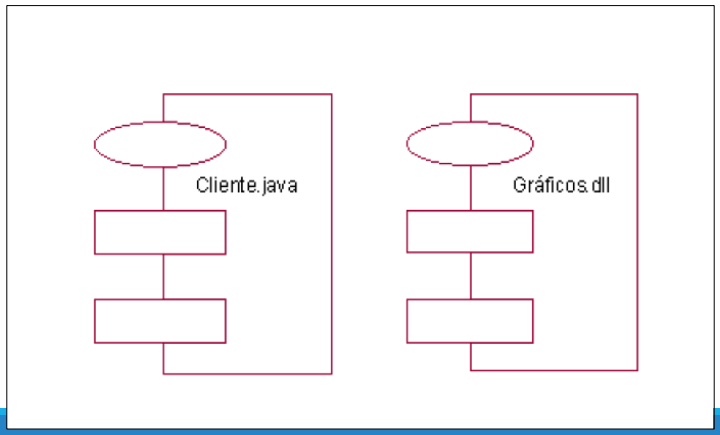
32

UNINASSAU

UML

Modelo de elementos:

Componentes:



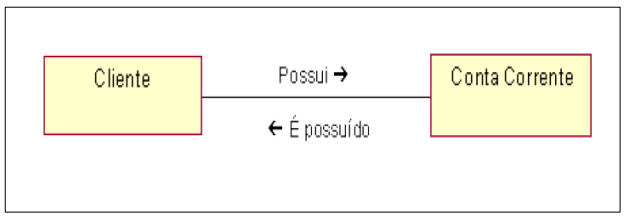
33

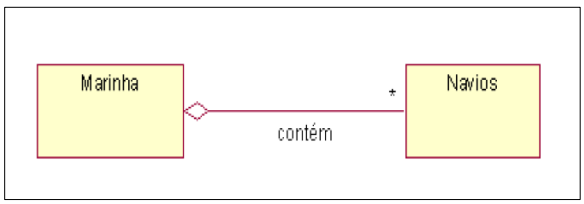
UNINASSAU

UML

Modelo de elementos:

Associações:





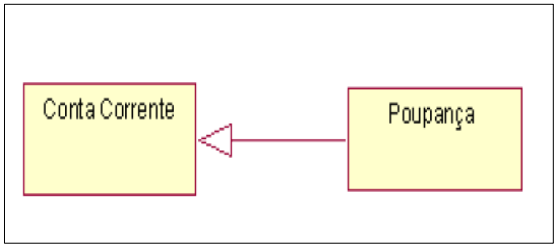
34

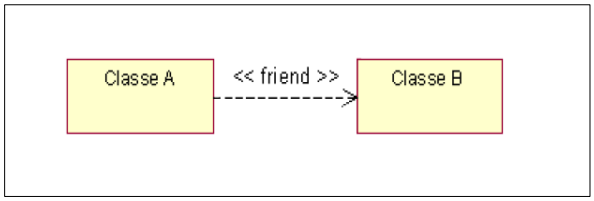
UNINASSAU

UML

Modelo de elementos:

Generalização:





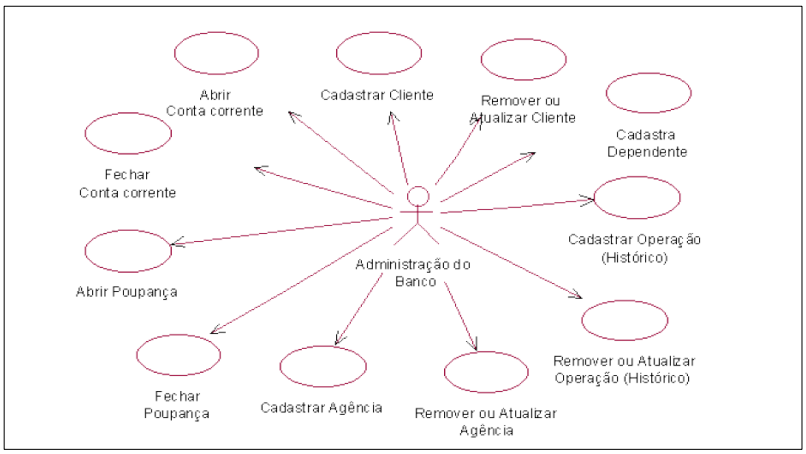
35

UNINASSAU

UML

Diagramas:

Casos de uso



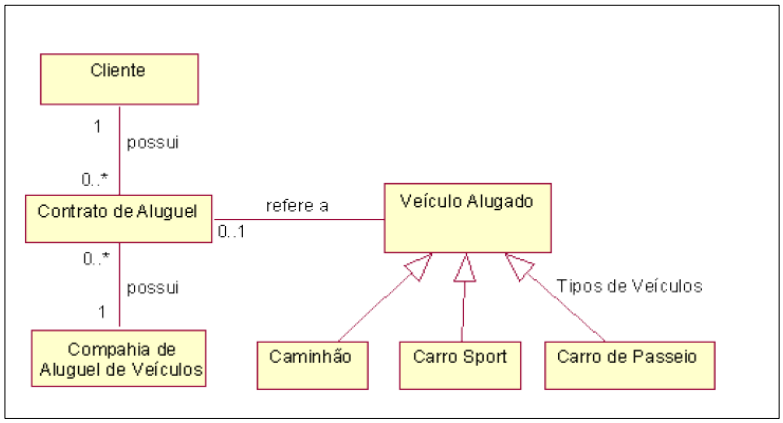
36

UNINASSAU

UML

Diagramas:

Classes:



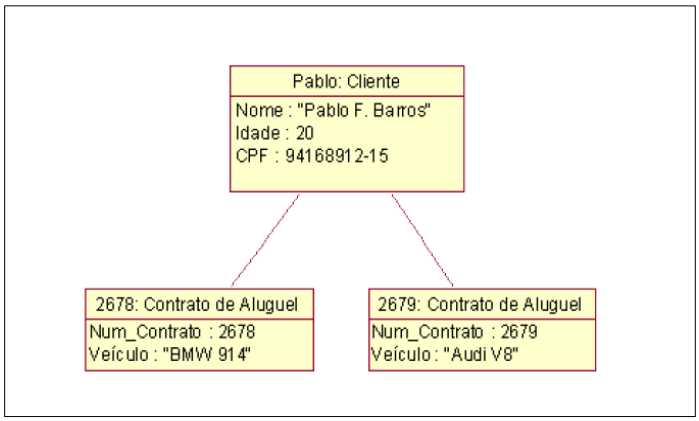
37

UNINASSAU

UML

Diagramas:

Objetos:



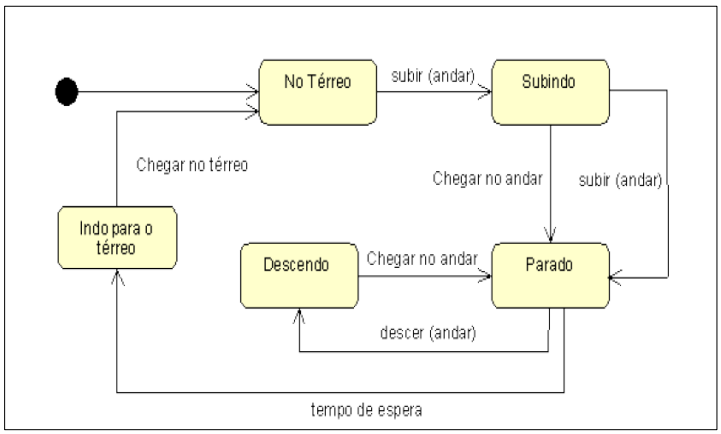
38

UNINASSAU

UML

Diagramas:

Estado:



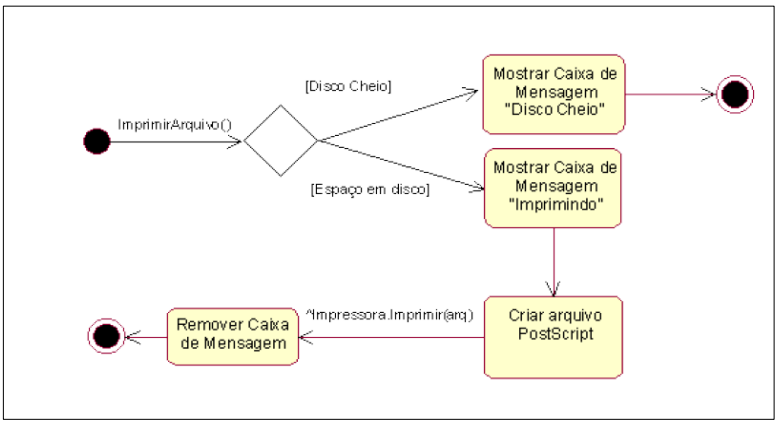
39

UNINASSAU

UML

Diagramas:

Atividade:



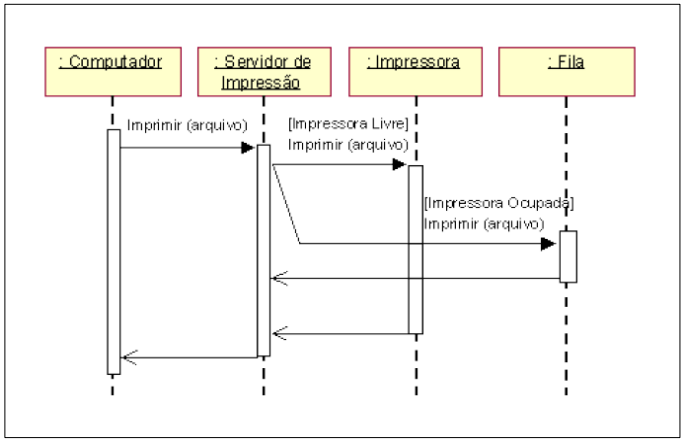
40

UNINASSAU

UML

Diagramas:

Sequência:



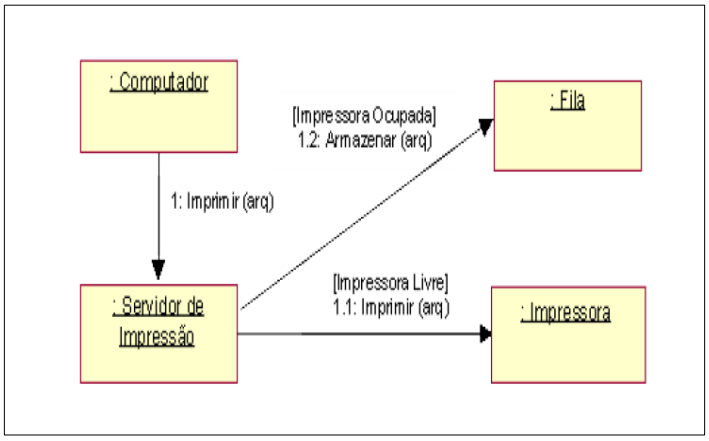
41

UNINASSAU

UML

Diagramas:

Colaboração:



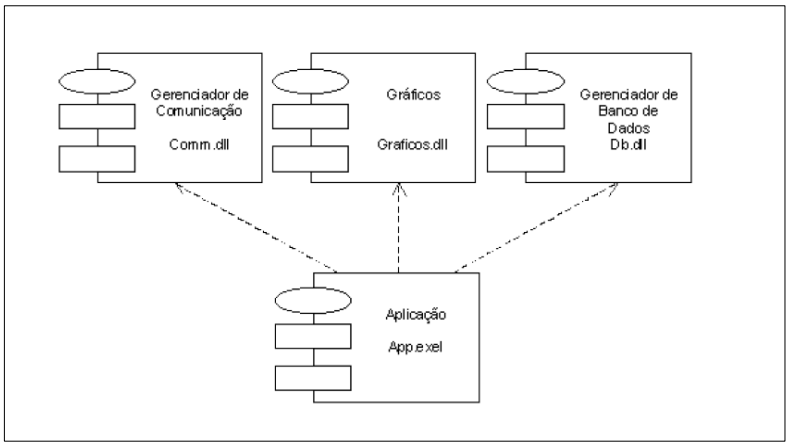
42

UNINASSAU

UML

Diagramas:

Componentes:



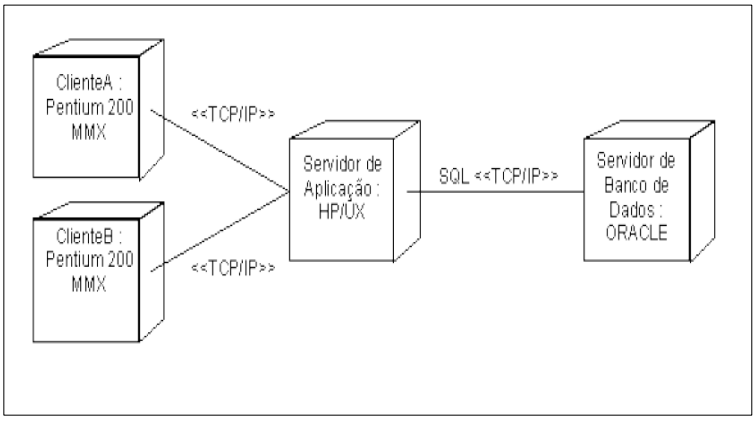
43

UNINASSAU

UML

Diagramas:

Execução:

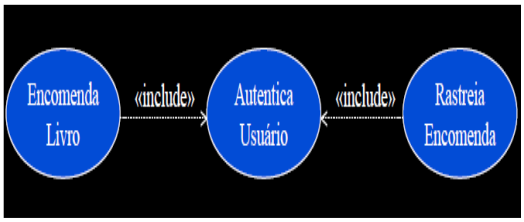


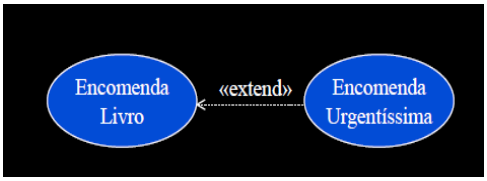
44

UNINASSAU

UML

Diagramas:



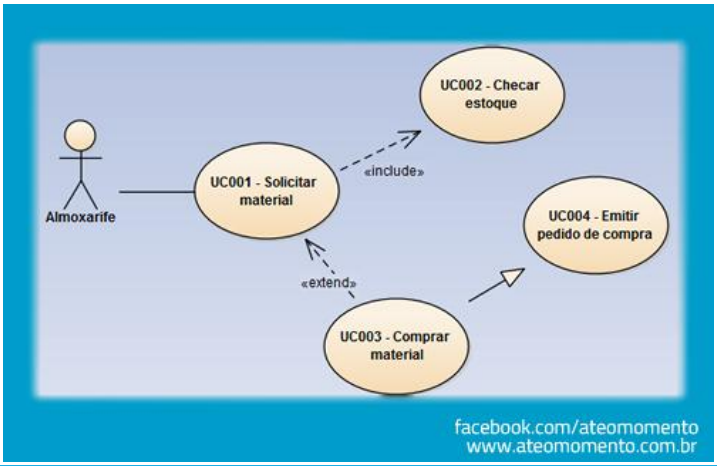


45

UNINASSAU

UML

Diagramas:



46