Classes
Modelagem UML

P. O. O.

Prof. Grace

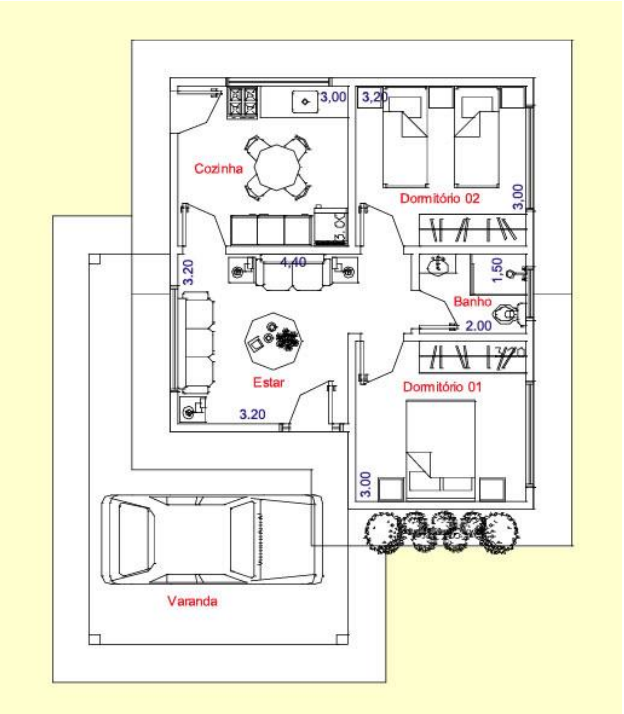
MODELAGEM DE SISTEMA

Definição de classes

- Quem define que classes criar durante o desenvolvimento de um Sistema?
- Que métodos criar para cada classe?
- E quanto aos atributos? Que tipos de dados eles devem armazenar? Quantos devo criar?
- Atividade: ***Modelagem de sistema***
  - Realizada para elaborar e representar as soluções dadas ao sistema (análise e projeto)

O que é Modelagem?

- Ato de criar modelos;
- Representação de algo do mundo real por meio de modelos;
- Simplificação da realidade:
  - Ex.: planta de uma casa representa, mas não é a casa.
- Modelagem de sistemas:
  - representações do sistemas que desenvolvemos;



É importante? Por quê?

Ajuda a:

1. Compreender o sistema;
2. Sair do mundo das ideias e para algo “palpável”;
3. Perceber furos de entendimento antes de implementar;
4. Ter compreensão única do sistema (equipe e cliente);

Modelagem de sistemas tem como objetivos:

1. Visualizar o sistema;
2. Especificar estrutura e/ ou o comportamento do sistema;
3. Proporcionar guia para construção;
4. Documentar tomadas de decisões;

Escolha de modelos

- A escolha dos modelos pode influenciar na definição da solução:
  - Modelos estruturados: soluções estruturadas
  - Modelos orientados a objetos: soluções O. O.
- Nenhum modelo único é suficiente.

Planta de Alvenaria

Planta Hidráulica

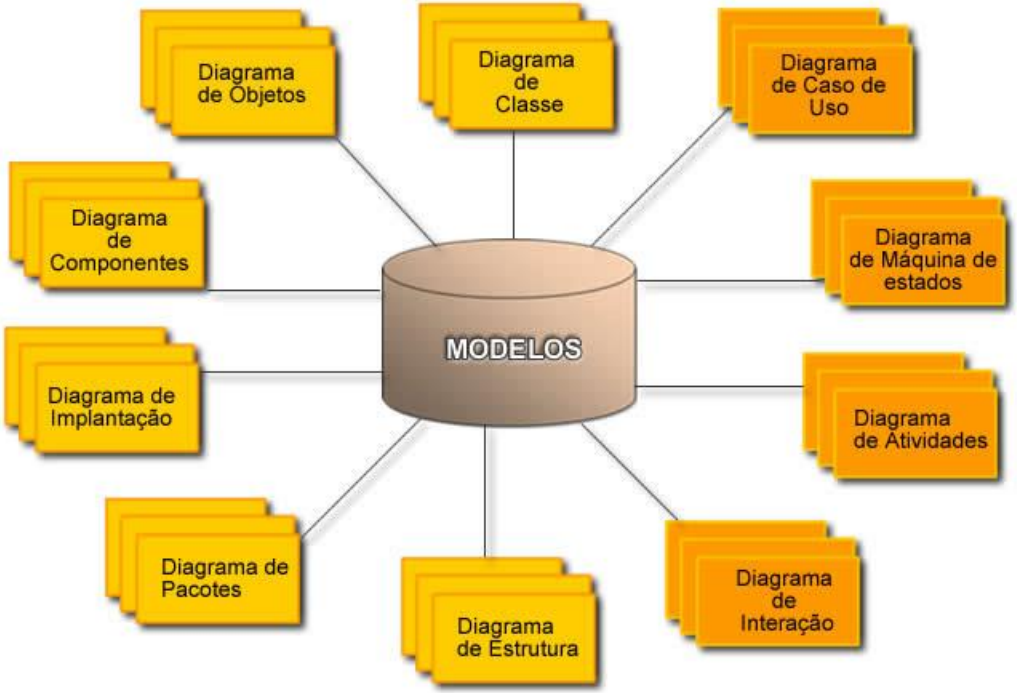
Planta Elétrica

Foco da disciplina: Modelagem O. O.

- Representação mais próxima da nossa realidade;
- Elementos base: objetos ou classes;
  - Classes: representam um conjunto de objetos;
  - Objetos possuem:
    - Identidade (nome/ instância);
    - Comportamento (métodos);
    - Estado associado a ele (valores que pode assumir);
- Abordagem bastante utilizada na construção de sistemas das mais variadas áreas, dimensões e complexidades: uso da linguagem UML.

UML (Unified Modeling Language)

- Linguagem para:
  - Visualização
  - Especificação
  - Construção e
  - Documentação



de artefatos de sistema de software

Artefatos: Requisitos do sistema, modelos, código de programa, componentes executáveis, arquitetura, etc.

Diagramas UML

- Diagramas de estrutura (6 diagramas):
  - Representam a visão estática da aplicação.
  - São eles: Classes; Objetos; Componentes; Estrutura composta; Pacotes; Implantação;
- Diagramas de Comportamento (3 diagramas):
  - Aspectos gerais de comportamento;
  - São eles: Casos de uso; Atividades; Máquina de Estados;
- Diagramas de Interação (4 diagramas):
  - Representa diferentes aspectos de interação entre classes e objetos; deriva dos anteriores;
  - São eles: Sequência; Comunicação; Tempo; Diagrama geral de interação

Fonte: http://www.uml.org/

Por que tantos diagramas?

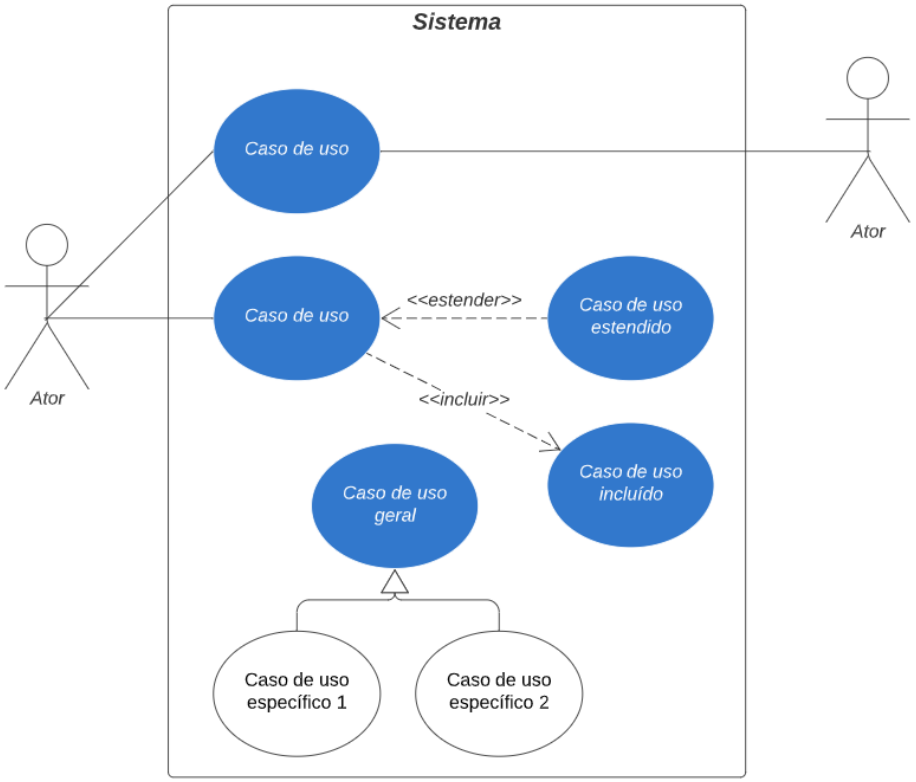
- Preciso implementar todos?
- Representam as múltiplas visões do sistema;
- Visão mais geral ou mais específica de acordo com o diagrama;
- Descoberta de falhas anteriores;
- Cada diagrama tem sua finalidade;
  - Ex.: Caso de uso, classe, sequência e atividade

1) Diagrama de Caso de uso

- Diagrama mais geral da UML;
- Base para desenvolvimento de outros diagramas;
- Auxilia no levantamento e análise dos requisitos;
- Linguagem simples e fácil compreensão para usuários do sistema;
- Ideia geral de como o sistema se comporta;
- Identifica atores do sistema (usuários e outros softwares);

Elementos básicos

- Atores = clientes/ sistemas
- Casos de uso = funcionalidades do sistema
- Relacionamento = relação entre as funcionalidades
  - Associação (simples)
  - Inclusão (caso incluído sempre ocorre)
  - Extensão (caso estendido pode ocorrer ou não)
  - Generalização (herança)

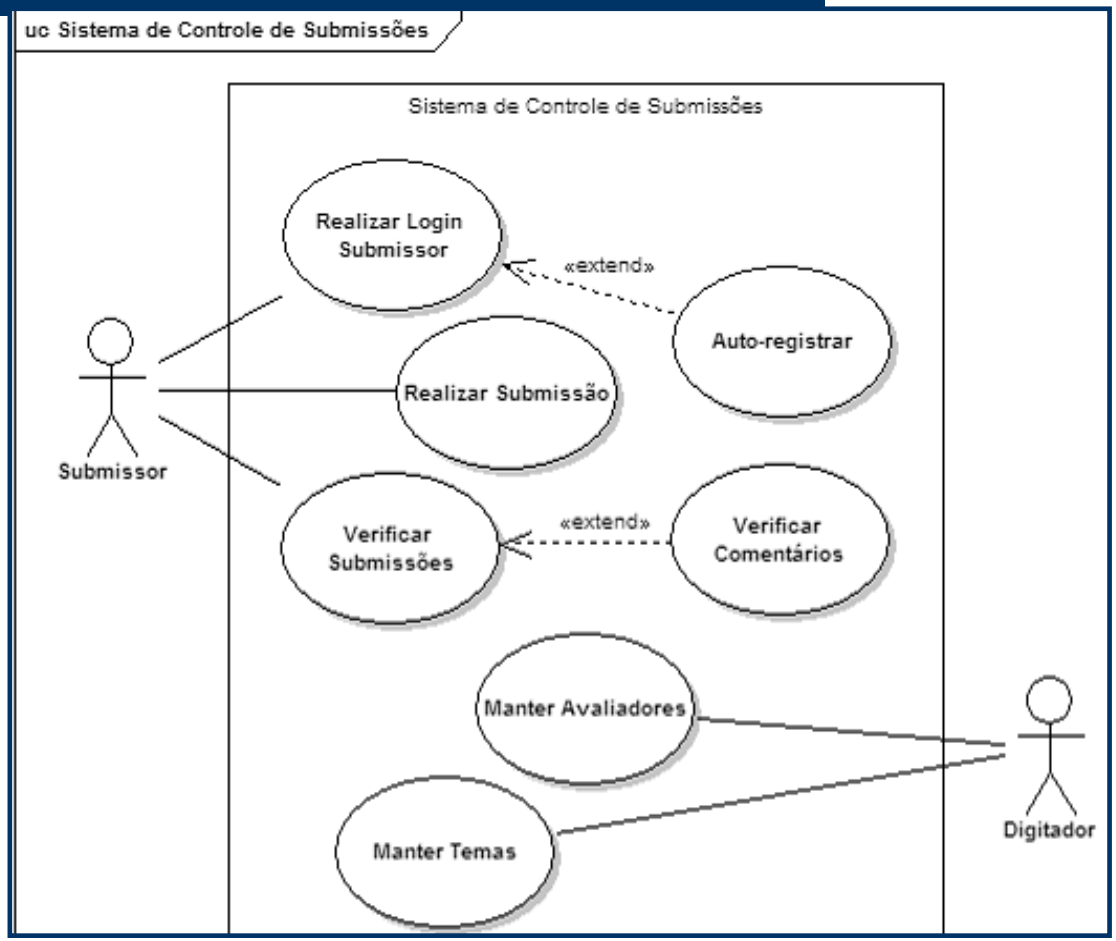


Exemplo: Submissões para Congresso

- Congressos são encontros para comunicar trabalhos de pesquisadores: artigos, minicursos, palestras, etc.
- Os trabalhos, ao serem submetidos pelos pesquisadores, passam por avaliadores que podem aceitar, rejeitar ou sugerir alterações para os trabalhos.
- Os seguintes exemplos são “recortes” de um sistema de Submissão para Congresso.

Exemplo: Sistema de controle de Submissões para Congresso

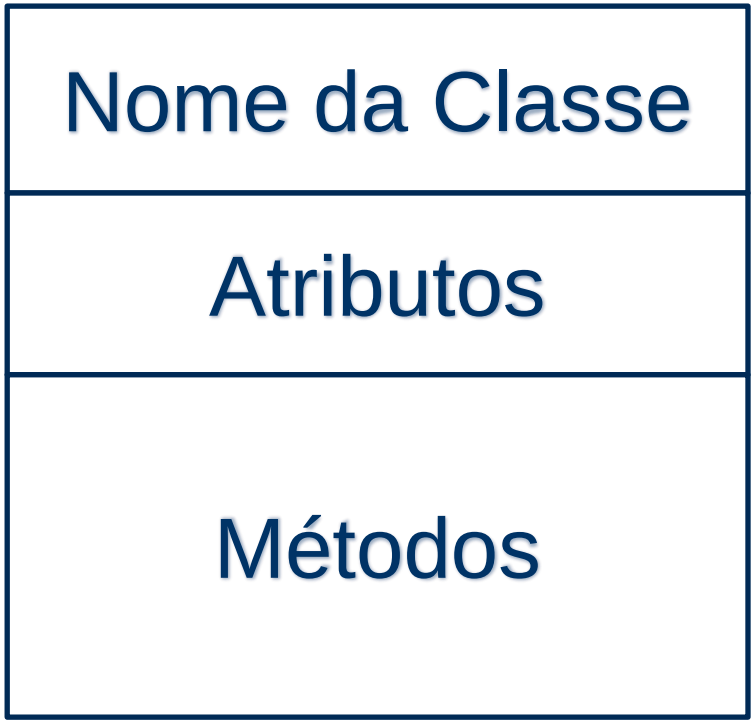
- Atores
- Casos de uso
- Relacionamentos:
  - Associação
  - Extensão



2) Diagrama de Classes

- Define a estrutura das classes utilizadas pelo sistema;
- Determina os atributos e métodos de cada classe;
- Estabelecer como as classes se relacionam entre si;
- Apoio para a maioria dos outros diagramas;
- Mais utilizado e importante da UML.

Elemento básico: Classe



Exemplo: Controle de Submissões

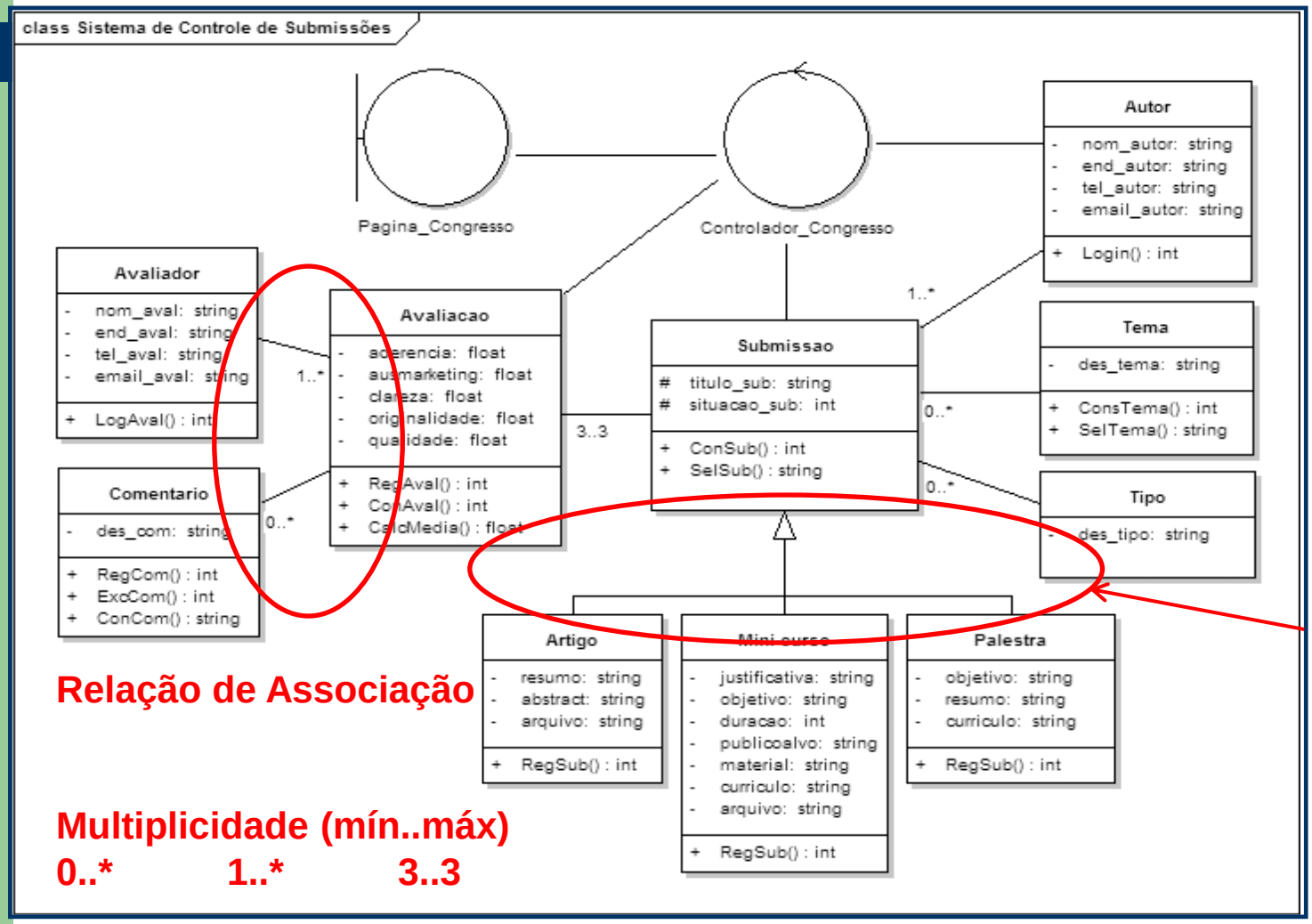
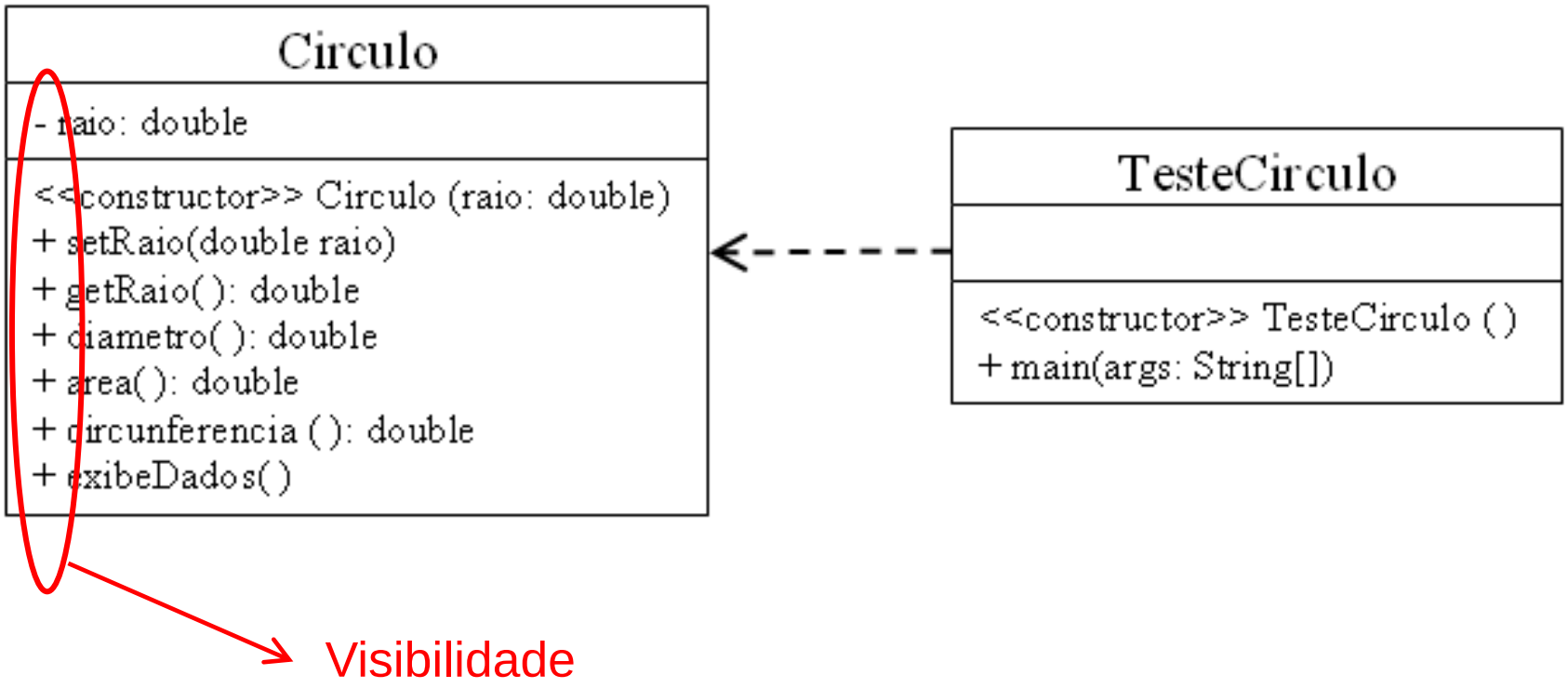
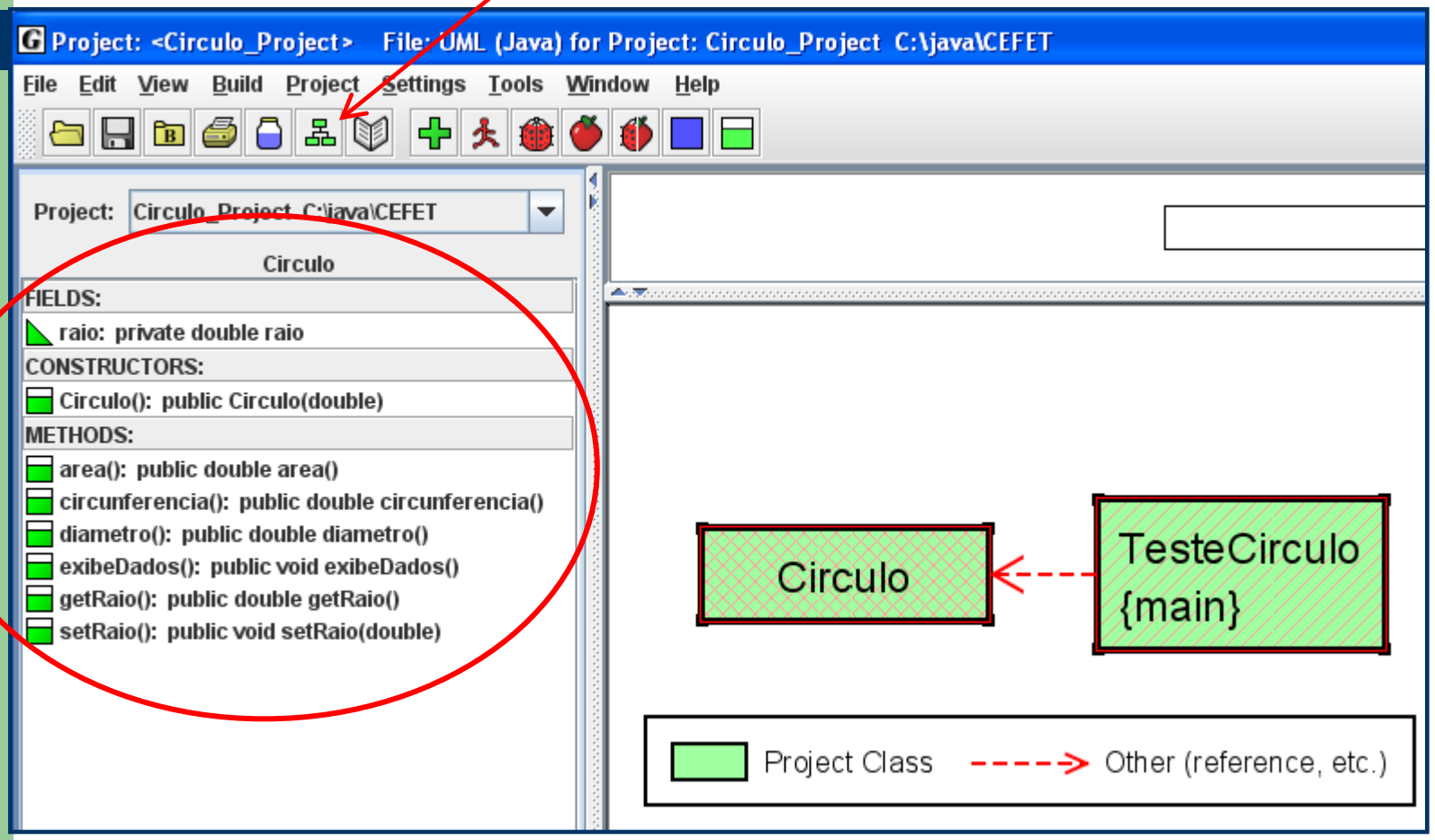


Diagrama de Classe
Exemplo: Círculo



No jGrasp

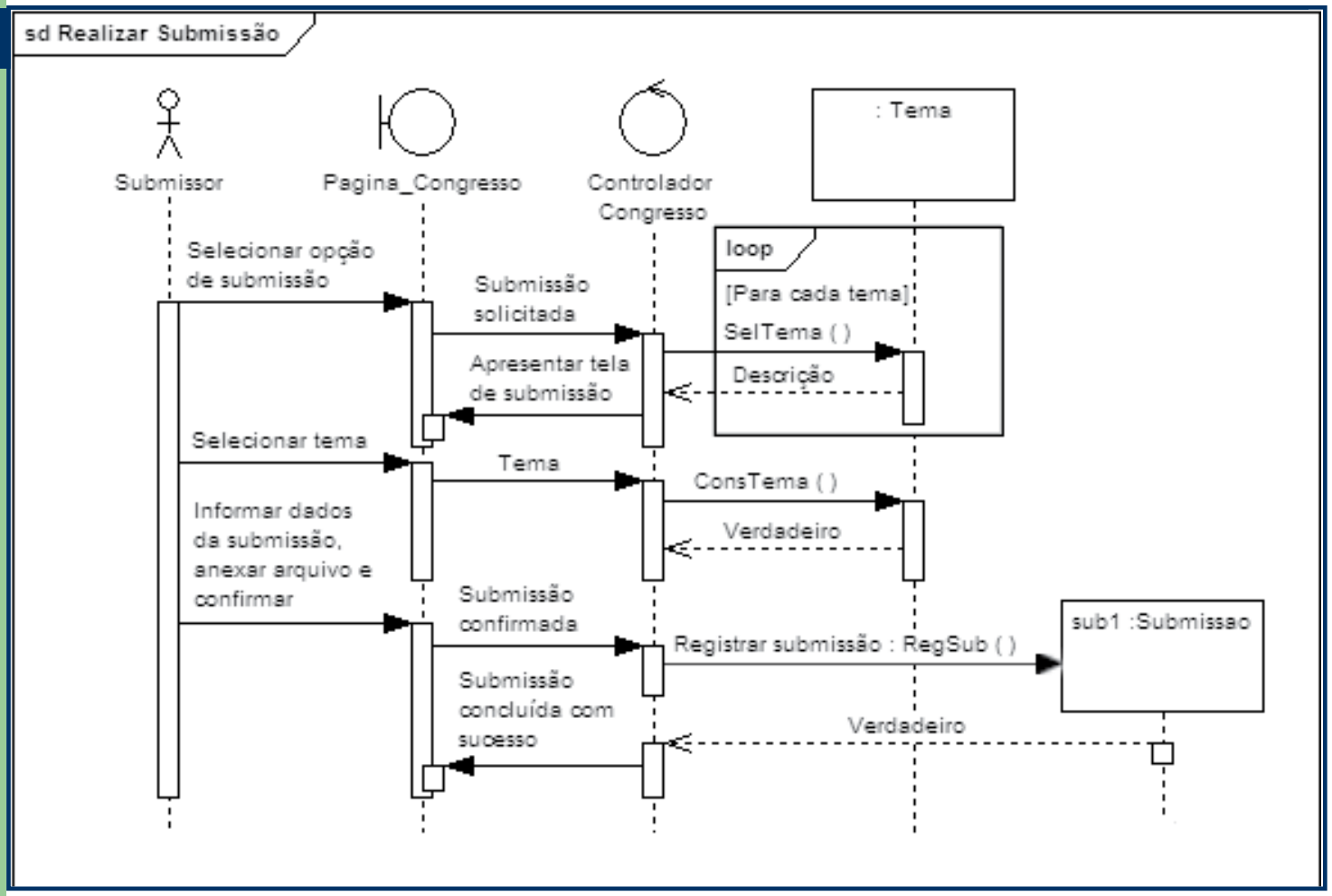
Crie um projeto e adicione as classes que fazem parte do projeto



3) Diagrama de sequência

- Preocupa-se com a ordem em que as mensagens são trocadas entre os objetos de determinado processo;
- Baseia-se em um Caso de Uso definido anteriormente;
- Apoia-se no Diagrama de Classes para determinar objetos envolvidos e os métodos disparados;

Exemplo Diagrama de Sequência
Caso de uso: Realizar Submissão

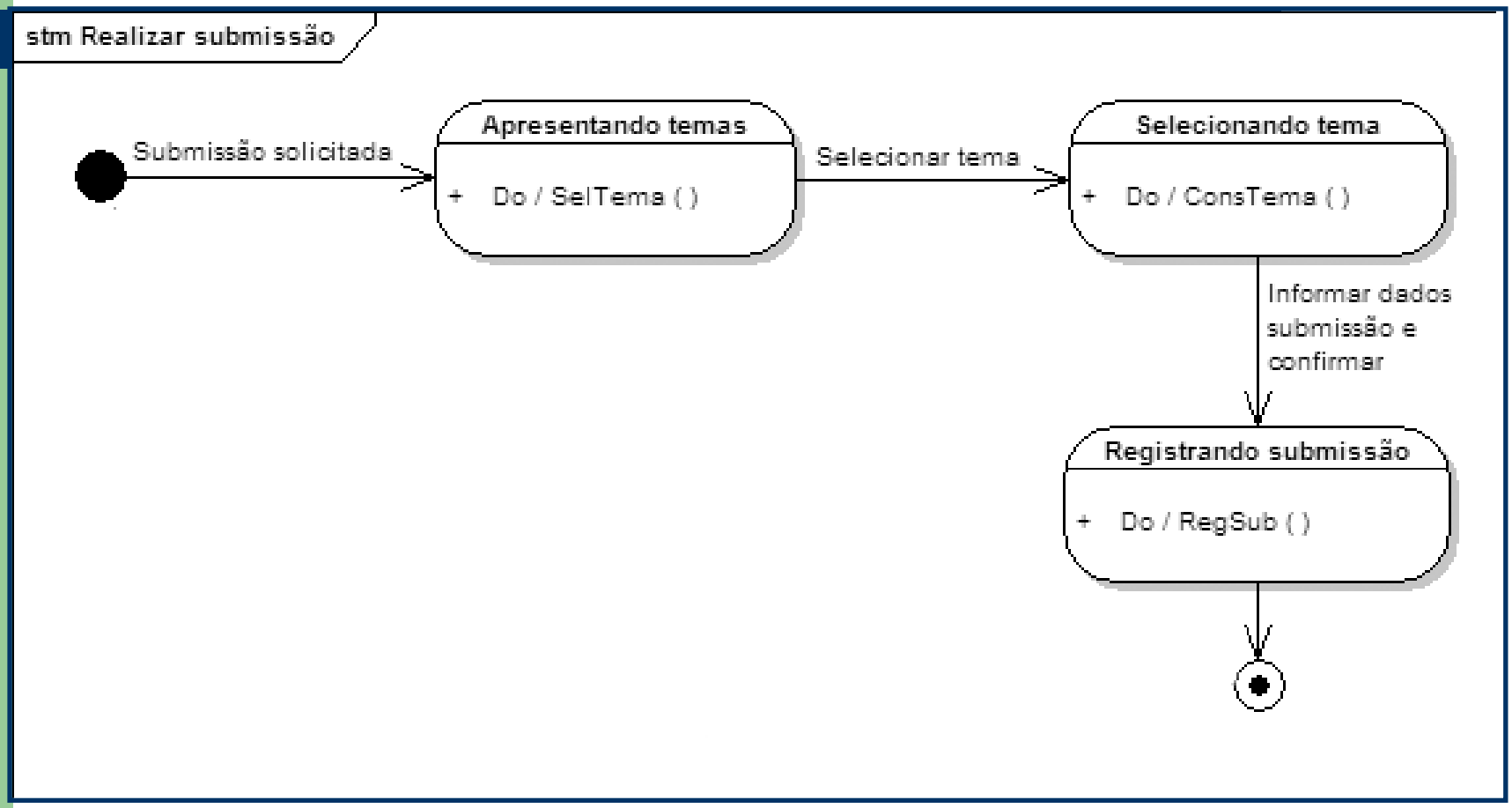


4) Diagrama de estados

- Identifica mudanças sofridas nos estados de:
  - instância de uma classe ou
  - caso de uso ou
  - subsistema ou
  - sistema completo.
- Baseia-se em um Caso de Uso
- Apoia-se no Diagrama de Classes

Exemplo: Estados Semáforo

Exemplo Diagrama de Estados
Caso de Uso: Realizar submissão



Atividade – Classe Retângulo

- Modele e implemente uma classe que represente um Retângulo a partir de seus atributos: **base** e **altura**
- Ela deve ser capaz de instanciar objetos a partir de 2 parâmetros (base e altura)
- Caso base e altura não sejam informados, instancie um retângulo de base = 2 e altura = 1;
- Cada instância deve ser capaz de:
  - Devolver os valores de: Área; Perímetro; Base; Altura;
  - Informar se o objeto é um quadrado (boolean)
  - Imprimir todas as informações sobre o objeto instanciado
- Elabore o diagrama de Classes (jgrasp ou word ou qualquer editor).