

Analise de Sistemas

Processo da fase de analise

- Teste de viabilidade;
- Conceitos;
- Definição do que o **Analista** faz;



Reunir e coletar dados, que são as **Necessidades** do cliente.



As necessidades são os problemas que o cliente traz e cabe ao analista e a empresa de desenvolvimento de software trazer uma solução mais adequada.

As informações são coletadas através de uma **Técnica de levantamento**.



Através do resultado, tem as **Coletas de dados**.



Se torna um **Levantamento**.



É feito através de **Estudo de caso**.



É definido os **Requisitos funcionais** e não funcionais.



Elaboramos os diagramas de casos de uso.

UML – Diagrama de casos de uso

Os diagramas de casos de uso, é um diagrama global, que mostra como vai ficar o sistema, ou seja, como o sistema vai se comportar. É um diagrama sem detalhes (atributos), bem objetivo.

Elementos que compõem os diagramas de casos de uso:

- Atores;
- Casos de uso (Componentes);
- Relacionamentos que vão conecta os Atores com Casos de uso e os Casos de uso para Casos de uso. São:
 - Associação;
 - Caso de herança (Generalização);
 - Dependência: Extensão e inclusão.
- Fronteira do sistema (Colocado no final do diagrama).

Atores

São conhecidos como bonecos magros, eles representam um papel naquele momento no sistema, eles são entidades externas que interagem com o sistema.

Se a informação for algo que interagem e tem acesso ao sistema, é chamado de atores. Atores são todas aquelas entidades externas que interagem com o sistema. Ou seja, para ser considerado atores, tem que interagir com o sistema. No mínimo tem que ter um Ator e no máximo vai depender do sistema.

Por exemplo: O funcionário é um ator.

CASO DE USO

São todas as funcionalidades do sistema, sempre é iniciado por um verbo.

Exemplo: Cadastra, consulta.

RELACIONAMENTO DE ASSOCIAÇÃO

Só existe um relacionamento entre atores e Caso de uso, que é chamado de Associação ou Comunicação. É representado por uma linha continua. As associações dirigidas, tem uma setinha na ponta.

Obs: Não pode ter palavras repetidas no diagrama casos de uso. No máximo duas palavras. A organização do diagrama tem que ser clara porque conta muito.

RELACIONAMENTO DE GENERALIZAÇÃO

Generalização de atores

Chama-se também de Herança, ocorre entre atores e casos de uso. Identificamos relacionamento de herança, quando dois atores tem uma função em comum. É uma hierarquia, do maior para o menor. A seta sempre é apontada para o Pai.-

Generalização de casos de uso

Para ser considerado um relacionamento de Herança entre casos de uso vai envolver sempre 3 casos de uso, o principal fica em cima (Pai) e os dois embaixo é os especializados, eles estão herdando alguma coisa do Pai. Identificamos primeiro se nos 3 casos de uso tem alguma informação semelhante, ou seja, a mesma informação nos 3 porem cada um tem que ter uma informação diferente. A setinha é sempre apontada para o Pai.

Exemplo 1: O pai (Estilo Musica), Filho1 (Funk) e Filho2 (Sertanejo).

Exemplo 2: Pai (Pagamento), Filho1 (Boleto Bancário) e Filho2(Cartão).

Eu posso ter dois relacionamentos no mesmo casos de uso, um filho pode virar um Pai.

EXTENSÃO

EXTEND

Extend e includes só vai ocorrer entre casos de uso, relacionamento tracejado só em casos de usos. Jamais vai acontecer o relacionamento tracejado em atores e casos de uso.

Relacionamento de Extend, é quando envolve um cenário opcional. O Extend, mostra uma condição, que pode ser satisfatória ou não. E sempre colocar o nome do relacionamento (Extend). A seta do Extend, é direcionado ao primeiro caso de uso.

Exemplo 1: Caso de uso (Cadastro) e outro caso de uso (Atualização de dados). A seta seria direcionada ao Cadastro

Exemplo 2: Casos de uso (Sorvete) outro caso de uso (Sabores), utilizamos aí o Extend.

INCLUDES

É uma variação do Extend, no include tem um cenário obrigatório, um caso de uso só existe se outro caso de uso existir. A seta no Includes é virada para o segundo caso de uso.

Exemplo 1: Caso de uso (Cadastro) e outro caso de uso (Atualização de dados). Só que é definido pelo sistema que a cada 15 dias fosse feita a atualização de dados.

Exemplo 2: Para ter um acesso ao sistema, é preciso fazer um cadastro. Caso de uso (Cadastro) e o outro caso de uso (Login). No caso a seta é virada para o Login.

Obs: Não pode ter um Extend e um Includes no mesmos caso de uso.

FRONTEIRA DO SISTEMA

A fronteira do sistema é representada por um quadrado. As entidades do sistema, que interagem com o sistema não podem se misturar com a funcionalidade do sistema. A funcionalidade do sistema é representada através dos casos de uso e esses casos de uso vai ficar dentro desse quadrado e no lado de fora desse quadrado vai estar os atores nas extremidades. No lado esquerdo, pode ir colocando quem representa a empresa e no direito, quem representa o cliente.

UML - Diagrama de classes

O diagrama de classe é como se fosse a visão do programador e o diagrama de caso de uso é como se fosse mais a visão do usuário. O diagrama de classes é composto pelas classes que são tabelas que tem três divisões. Então, o conjunto dessas informações recebe o nome de classes, então ela tem essas três divisões:

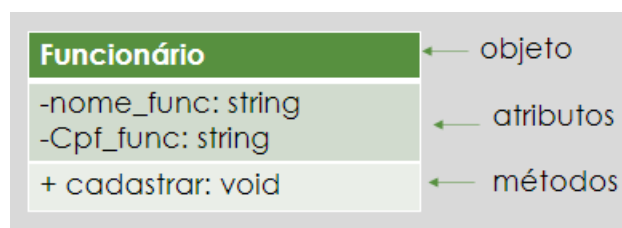
- Nome
- Atributos
- Operações (que seria os métodos)

Atributos e Métodos

Então na primeira divisão, tem o objeto ou nome da classe;

Na segunda divisão, tem os atributos que são as características do objeto;

Na terceira divisão, os métodos que são as ações ou comportamentos dos objetos



Nas tabelas, tem a visibilidade e os tipos de Dados. A visibilidade vem na frente tanto dos métodos quanto dos atributos. O - e o + são as visibilidades, que é como as classes conseguem enxergar umas às outras, é conseguir enxergar as informações que estão numa classe ou em outra.

No lado esquerdo da tabela, temos os tipos de dados depende da linguagem de programação, então todos tem um tipo de dado. Os métodos na maioria das vezes retornam algo por isso quase sempre é void.

Atributos e operados

Os atributos e operados que possuem uma visibilidade, a visibilidade indica se o atributo pode ser visualizado por outras classes além da própria, que pode ser:

- + símbolo do público, ele pode ser visto e usado por outras classes;
- - Símbolo do privado, ele não pode ser acessado por outras classes;
- # símbolo de protegido, ele pode ser acessado por subclasses da classe e que a pertence;

- ~ símbolo do pacote, significa que as classes de um pacote podem ser usadas. É pouco utilizada.

O que na maioria das vezes é utilizado e padronizado são os símbolos do público (+) e do privado (-).

O público é mais utilizado para os métodos, que geralmente é chamada de outros métodos, tem um retorno e geralmente os atributos são privados (-) e tem atributos que são protegidos (#). Geralmente os atributos são privados (-) e os métodos são públicos (+). Tanto os métodos quanto aos atributos eles possuem a visibilidade.

Associação Binária

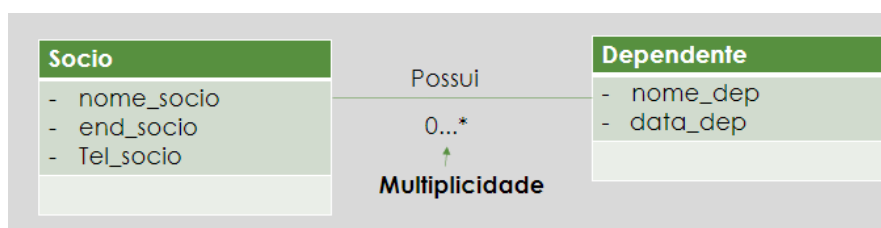
O primeiro relacionamento que existe entre as classes, é o relacionamento de associação Binária. A associação Binária, é um relacionamento padrão, ele está ligado porque ele tem que estar ligado mais não tem vínculo nenhum. A ligação dele é parecida com a do diagrama de caso de uso, aquele da ligação do ator com os casos de uso, aquela linha contínua.

A notação da associação Binária é aquela linha contínua, que conecta as classes, porém não tem vínculo nenhum. Ela é muito utilizada dentro do sistema, para cadastrar algo, consulta algo, onde uma classes estar ligado para Consulta, cadastra, esse são relacionamentos padrão.

A ligação de associação Binária ocorre quando são identificados relacionamentos entre objetos e duas classes diferentes. Esse tipo de associação é a mais utilizada.

Multiplicidade

A multiplicidade é basicamente a mesma coisa que a cardinalidade no banco de dados, a diferença é que invés de utilizar o N para representar muitas coisas, aqui utilizamos o símbolo *. Toda vez que for pensar em algo, pensar no mínimo e no máximo. Não pode repetir as palavras entre as classes. Quando não tiver método, tem que manter a última divisão deixando vazio. O mínimo pode começar por qualquer número, vai depender da regra de negócio.



Associação Ternaria ou N'aria

São associações que conectam objetos entre três a quatro classes. São representadas por um losango para onde convergem todas as ligações da associação. Esse relacionamento não muito utilizado, quando utilizar tomar cuidado porque ele envolve no mínimo três e no máximo quatro classes, o losango serve apenas para ligar as classes, as classes devem estar interligadas, ou seja, estar no mesmo ciclo para poder usar esse relacionamento. Normalmente, a multiplicidade, ela é a mesma para todos, na leitura não importa por onde começar porque sempre vai dá na mesma, vai dá certo.



Generalização/Especialização

Esse relacionamento é de herança, igual o do diagrama de casos de uso. O objetivo desse relacionamento é representar a ocorrência de herança entre as classes, identificando as classes-mães (ou superclasses), chamadas gerais, e classes-filhas (ou subclasses), chamadas especializadas, demonstrando a hierarquia entre as classes.

São três classes envolvidas, as classes filhas em baixo e a classes geral em cima. Elas têm que ter algo em comum. Não precisa repetir os métodos e os atributos da classe geral nas classes filhas. Esse relacionamento não precisa ter as palavras e a multiplicidade.



Agregação

A agregação é um tipo de relacionamento que tem um vínculo, apresento a classe todo e a classe parte, eles estão conectados através da agregação. A agregação serve para agregar para outros fins. A classe parte é como se fosse um complemento da classe toda, ele possui atributos e métodos. O carro pode ficar sem as rodas e as rodas podem ficar sem o carro, isso é uma forma de se pensar. Ele possui um símbolo, um losango na extremidade da classe que contém os objetos-todo.

A agregação é um tipo especial de associação onde se tenta demonstrar que as informações de um objeto (chamado objeto todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos parte). Esse é um tipo de relacionamento que tenta demonstrar uma relação todo/parte entre os objetos associados.



Composição

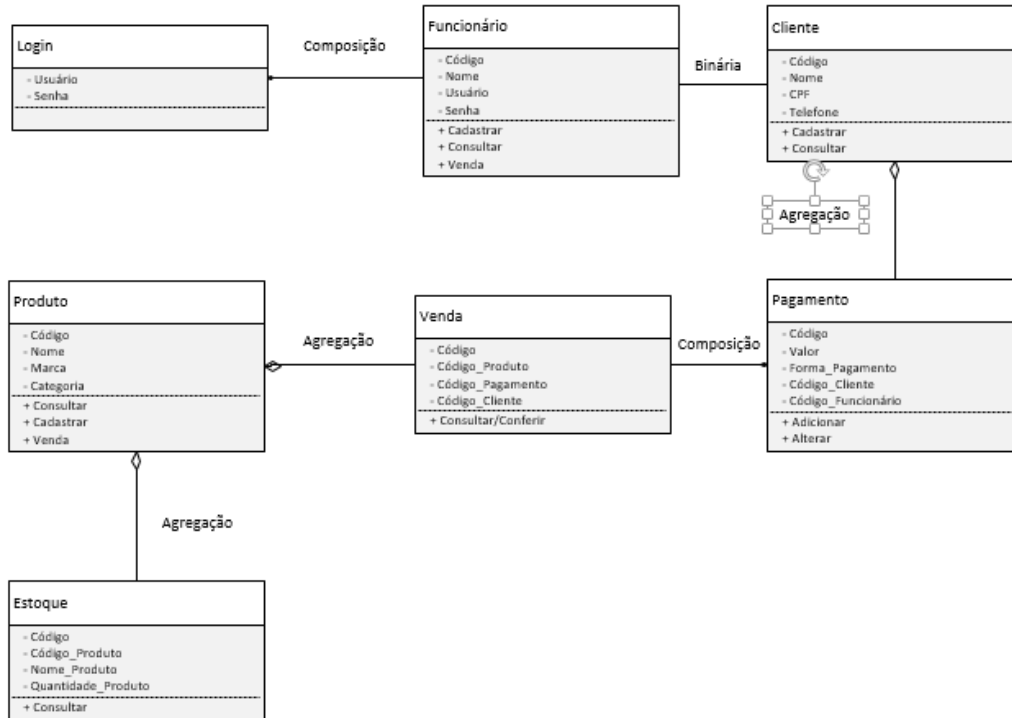
Constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre os objetos todo e os objetos parte, procurando demonstrar que os objetos-parte têm de estar associados a um único objeto todo. O símbolo de composição diferencia-se graficamente do símbolo de agregação por utilizar um losango preenchido.

A composição é como se fosse um include, o objeto todo depende exclusivamente do objeto parte e vice-versa. Um não existe sem o outro.



Atividade Prática

Os tipos de relacionamentos e visibilidade utilizados:



Tipos de dados e as multiplicidades utilizados:

