

Nos dias de hoje o abuso do vocábulo sistema faz com que repliquemos o termo em diversos ramos diferentes, como sistema político, sistema digestivo, sistema de fabricação entre outros, também temos sistemas baseados em computador. Este ultimo pode ser visto como um arranjo de coisas relacionadas para atingir alguma meta por meio de processamento de informação, envolvendo software, hardware, pessoas, banco de dados, documentação e procedimentos. **A análise de sistemas implica em realizar um estudo profundo de forma a entender como o sistema funciona e verificar se é necessário introduzir melhorias, sendo acima de tudo uma atividade de resolução de problemas.** Esta atividade envolve desenvolver um conjunto de modelos gráficos que atua como especificação do sistema, ajudando a descrever seu comportamento e confiabilidade requerida.

A análise de sistemas é uma série de tarefas de modelagem que levam a especificação completa dos requisitos e à representação abrangente do projeto a ser construído. Basicamente seu objetivo é descrever o que o cliente exige; estabelecer a base para criação do software; e, definir um conjunto de requisitos que podem validar o que foi construído. Cabendo ao analista focalizar nos requisitos fornecendo uma visão aprofundada do domínio do negócio, ao ponto de representar os relacionamentos entre classes e funções. Ao analisar os requisitos em domínios específicos de negócio, a engenharia de software observa que padrões se repetem entre as aplicações de forma a serem definidos e categorizados, permitindo ao analista quando reconhece-los poder reutiliza-los.

A criação de modelos de análise somado a utilização de componentes reutilizáveis melhora os prazos de colocação no mercado e reduz o custo de desenvolvimento. Contudo o modelo deve ser simples o suficiente e agregar valor aos interessados, podendo ser aplicado em diversos seguimentos como, projetos, orçamentos, garantia de qualidade. A análise orientada a objetos (OO), é uma atividades de construção de modelos e documentação, tem foco no mapeamento de uma solução sistêmica para processos de negócios. Todos os modelos aplicam os conceitos de orientação a objetos na construção dos modelos.

A técnica de análise Orientada a Objetos permite representar as informação composta que devem ser compreendida pelo software. Os objetos podem ser qualquer coisa do mundo real, se você esta construindo uma casa, paredes, portas e janelas são alguns itens ou objetos que serão importantes para você. Alem de poderem ser diferentes, cada uma dessas coisas também possuem comportamentos distintos e um conjunto de propriedades que os caracteriza. As portas tem altura, largura e são solidas e possuem o comportamento adicional que permitem serem abertas em uma direção. As janelas são semelhantes às portas, mas têm propriedades ligeiramente diferentes, seu objetivo não é permitir que as pessoas a usem como passagem mas sim que olhem através delas. Portanto vemos que os objetos, possuem características e comportamentos que podem ser descritos, a esta descrição conceitual da-se o nome de Classe.

Na orientação a objetos (OO), se você pensar na classe Pessoa, ela terá alguns atributos comuns a qualquer pessoa como, nome, idade, sexo e uma série de comportamentos como, correr, pular, dormir. Uma classe define o que é uma pessoa, e uma Instância dessa classe representa uma pessoa no mundo real, como Maria e Pedro. A classe não é o objeto propriamente dito, mas sim, um template para a construção de qualquer objeto, como, Maria e Pedro são instâncias da classe Pessoa. Uma outra característica da classe é sua identidade, que deve ser única dentro do sistema para que não haja conflito. A nomenclatura da classe deve possuir uma semântica dentro do contexto da aplicação e por convenção o nome da classe deve iniciar sempre com a inicial em maiúscula (Pessoa) e se o nome for composto, as iniciais de cada palavra em maiúscula (PessoaFisica, PessoaJuridica) e não usar ifém, underline ou espaço para separar as palavras, elas devem está unidas. Nela conterá todos os atributos e métodos de um determinado objeto. Os atributos são as propriedades e os métodos os comportamentos de um objeto, ao implementar a classe Pessoa, podemos identificar alguns atributos como: nome, idade, sexo, entre outros e alguns métodos como: andar, falar, correr, entre outros. O processo de mapear todos os atributos e métodos necessários para uma classe é chamado de Abstração. Outro aspecto importante da POO que merece ressalva, permite otimizar a produção da aplicação em tempo e linhas de código, e consiste no reuso de código já inscrito, de forma que as classes podem herdar características de outras classes pais ou superclasse, e por sua vez vão passar para seus filhos.

Todo esse contexto técnico e complexo da POO somando a todos processo de análise, pode acarretar na dificuldade de compreensão por parte de outros membros da equipe do projeto e também do cliente. Neste sentido se faz uso da Linguagem UML (Unified Modeling Language) uma linguagem gráfica para modelagem de software e processos de negócios. A UML abrange elementos conceituais, como processo comerciais, funções de sistemas e elementos concretos como classes de linguagens de programação e orientação a objetos. É independente de processos, apesar de ser utilizada em processos orientados a caso de uso, centrados na arquitetura, interativos e incrementais, também abrange a documentação e todos os seus detalhes. A UML fornece um conjunto de diagramas, mas iremos abordar apenas dois, que são, diagrama de caso de uso e diagramas de classe.

O diagrama de caso de uso mostra o que o sistema faz do ponto de vista do usuário, descrevendo as principais funcionalidades do sistema porém, não nos aprofundamos em detalhes técnicos. Ele é composto por um Cenário (conjunto de eventos que acontecem quando o usuário interage com o sistema), um Ator (um tipo de usuário, graficamente representado por o um homem palito), um Caso de Uso (uma tarefa realizada por um ator, representada por uma elipse) e a Comunicação que liga o autor ao caso de uso representado por setas. O diagrama de caso de uso permite modelar o contexto de um sistema dando ênfase ao atores ao redor do sistema, em um sistema bancário por exemplo, temos coisas internos ao sistema, como gerentes de contas, validação de conta, cartão de crédito. Do mesmo modo, fora do sistema temos coisas como cliente de conta, cliente de crédito. Todas essas coisas externas e internas se relacionam entre se e fazem parte do sistema. Identificar,

organizar criar estereótipos para os atores bem como preencher o caso de uso para cada ator permite decidir o que está incluído e o que será restringido no sistema, a partir daí podemos estabelecer quais são os requisitos do sistema declarando um tipo de contrato com o que se espera que o sistema faz.

A representação gráfica das classes em UML permite visualizar uma abstração independente de qualquer linguagem de programação, dando ênfase a partes importantes como, nome, atributos e comportamentos. A representação gráfica de uma classe é na forma de um retângulo dividido em 3 partes, a primeira é um cabeçalho com o nome da classe e estas devem ter nomes diferentes uma das outras, a segunda parte contém a lista dos atributos, por fim, a última contém a lista dos métodos que determinam o seu comportamento. Em um sistema bancário um cliente possui uma conta, ou seja, a classe Cliente se relaciona com a classe Conta. Os relacionamentos são os tipos de herança mais importantes entre classes são, as dependências, generalizações e as associações. As dependências são relacionamentos de utilização e determinam a modificação de um item e não o seu inverso, essa relação é representada por linhas tracejadas com seta apontando o item do qual o outro depende. A generalização pode ser definida como um relacionamento de herança do tipo, subclasse filha e classe mãe, por exemplo as classes conta corrente e conta poupança tem algumas características e propriedades comuns a classe conta, dessa forma, Conta Corrente e Conta Poupança possuem uma generalização com a classe Conta, é representada por uma linha sólida que aponta para a classe mãe. As associações são relacionamentos estruturais entre instâncias de classes, tomando como exemplo o sistema bancário, isso quer dizer que o Cliente João vai ter uma associação com a conta uma corrente corrente ou poupança.

A análise é uma atividade de investigação do problema e consiste das atividades necessárias para entender o **domínio do problema** descrevendo tudo o que deve ser feito de modo que cliente e equipe do projeto tenha conhecimento do que vai ser feito, assim que aprovado. O projeto descreve a fase de desenvolvimento e por sua vez, inclui as atividades que resultam em informação mais direcionada a equipe de desenvolvimento como modelos, arquiteturas e como pode ser feita a solução o que consiste das atividades de criação. A perspectiva empregada de objetos durante a Análise OO, está em descrever objetos no domínio do problema. Durante o projeto o mesmo objeto, a ênfase está em construir os objetos lógicos de software que poderão ser eventualmente implementados usando uma linguagem de programação OO. Durante a construção (programação OO), os objetos do projeto são implementados e deseja-se que sejam rigorosamente testados. Algumas ferramentas de desenvolvimento são capazes de utilizar a modelagem construída no processo de análise para gerar código fonte de algumas linguagens de programação. De qualquer forma quanto melhor modelado for a documentação, melhor será o entendimento e compreensão do problema a ser resolvido.

O trabalho investigativo da Análise aplicado a Projetos Orientados a Objetos e o uso de ferramentas cases como a UML permite criar uma variedade de representações em sua maioria de forma gráfica

que demonstre as características, comportamentos, requisito, arquitetura, processos e etapas de processos de software. Esse arranjo é consolidado no mercado. Como já abordado a UML fornece ferramentas que constroem modelos tanto para o processo de negócio quanto modelos complexos de desenvolvimento, utilizados para visualizar melhor algumas estratégias de arquitetura de um projeto de sistema.