

UML



UML (UNIFIED MODELING LANGUAGE), é uma linguagem poderosa para comunicação em equipes de produção de software!

É uma linguagem padrão para a elaboração da estrutura de projetos de software. Pode ser utilizada para a aplicação, visualização, especificação, construção e a documentação de artefatos que façam uso de sistemas complexos de software.

UML é uma linguagem utilizada para:

- Visualizar.
- Especificar.
- Construir e
- Documentar.

DIAGRAMAS DA UML:

DIAGRAMAS ESTRUTURAIS E COMPORTAMENTAIS

Usar a ferramenta certa, para a atividade certa. Agir deste modo, com certeza, facilita e muito o trabalho, você concorda?

Com esse objetivo de facilitar o trabalho e aumentar a produtividade, bem como o entendimento das pessoas envolvidas em cada fase do projeto, é que a **UML desenvolveu um diagrama para cada função**. Nem sempre se usa todos em todos os projetos, mas é necessário que se tenha o conhecimento do que eles fazem para tirar o melhor proveito e obter sucesso nos projetos.

DIAGRAMAS

Um diagrama é a apresentação gráfica de um conjunto de elementos, geralmente representada como gráficos de vértices (itens) e arcos (relacionamentos). São desenhados para permitir a visualização de um sistema sob diferentes aspectos.

Talvez ainda surja o questionamento: “Para que tantos diagramas? ”

O objetivo é fornecer múltiplas visões do sistema que está sendo modelado, analisando-o e modelando-o sob diversos aspectos, e procurando, assim, atingir a totalidade da modelagem, permitindo que cada diagrama complemente os outros. Em cada um dos diagramas da UML é analisado o sistema, ou uma parte dele, de um determinado ponto de vista; é como se o sistema fosse modelado em camadas.

Em alguns diagramas o sistema é visto de forma mais ampla, sem muitos detalhes técnicos, apresentando uma visão externa do sistema e de fácil entendimento para pessoas que não têm muito conhecimento técnico, **como é o objetivo do Diagrama de Caso de Uso.**

Diferente de outros diagramas que oferecem uma visão um pouco mais profunda do software, eles têm por objetivo apresentar um aspecto mais técnico ou ainda visualizar apenas uma característica específica do sistema ou um determinado processo.

A utilização de diferentes diagramas permite que falhas possam ser encontradas nos diagramas anteriores e assim diminuir a possibilidade da ocorrência de erros durante a fase de desenvolvimento do software.

Vale ressaltar, como já foi dito anteriormente, que mesmo que cada diagrama tenha sua função, nem sempre é necessário modelar um sistema utilizando-se de todos os diagramas, pois alguns deles possuem funções muito específicas, como é o caso do Diagrama de Tempo, que, por exemplo, tem como objetivo focalizar o tempo ou duração da mensagem ou condições em mudança em uma linha de tempo no diagrama.

Veremos os seguintes Diagramas:

DIAGRAMAS ESTRUTURAIS

DIAGRAMAS COMPORTAMENTAIS

DIAGRAMAS DE INTERAÇÃO

DIAGRAMAS ESTRUTURAIS

De uma maneira resumida, os diagramas estruturais definem a estrutura do sistema tanto na parte de software, quanto de hardware.

Diagrama de Objetos

Este diagrama está amplamente associado ao Diagrama de Classes (detalhado mais adiante). Ele consiste em uma variação do Diagrama de Classes, em que, em vez de classes, são representadas instâncias e ligações entre instâncias.

O Diagrama de Objetos é praticamente um complemento do Diagrama de Classes, sendo bastante dependente desse. Este diagrama fornece uma visão dos valores armazenados pelos objetos das classes contidas em um Diagrama de Classes em um determinado momento da execução de um processo. Em outras palavras, ele modela as instâncias das classes contidas no Diagrama de Classes, isto é, o Diagrama de Objetos mostra um conjunto de objetos e seus relacionamentos no tempo. Normalmente, são compostos por: objetos e vínculos.

Para facilitar o entendimento, a figura abaixo mostra um Diagrama de Objetos para o **cliente Silva** e seus dois **pedidos na COMPRAS & COMPRAS**.

Podemos ler o diagrama como:

- O objeto Silva da classe Cliente está associado aos objetos 991 e 774, ambos da classe Pedido.

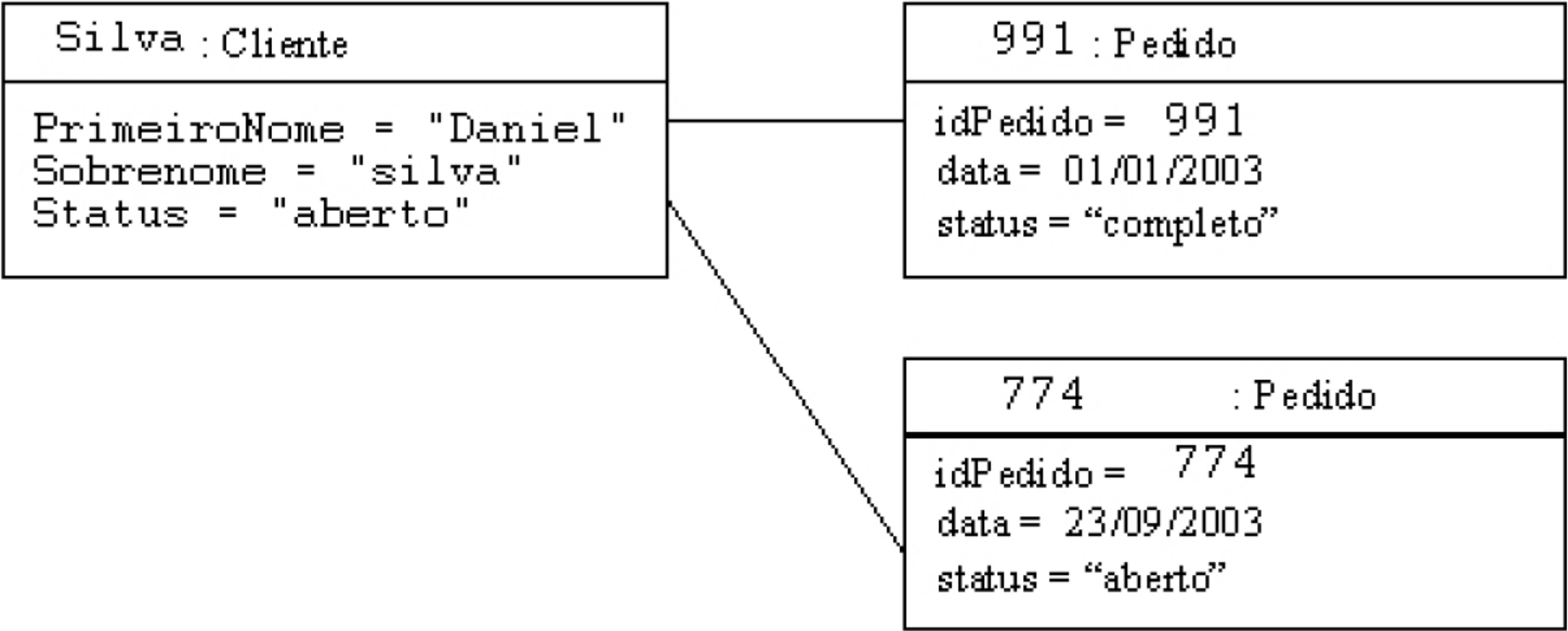


Diagrama de Classes

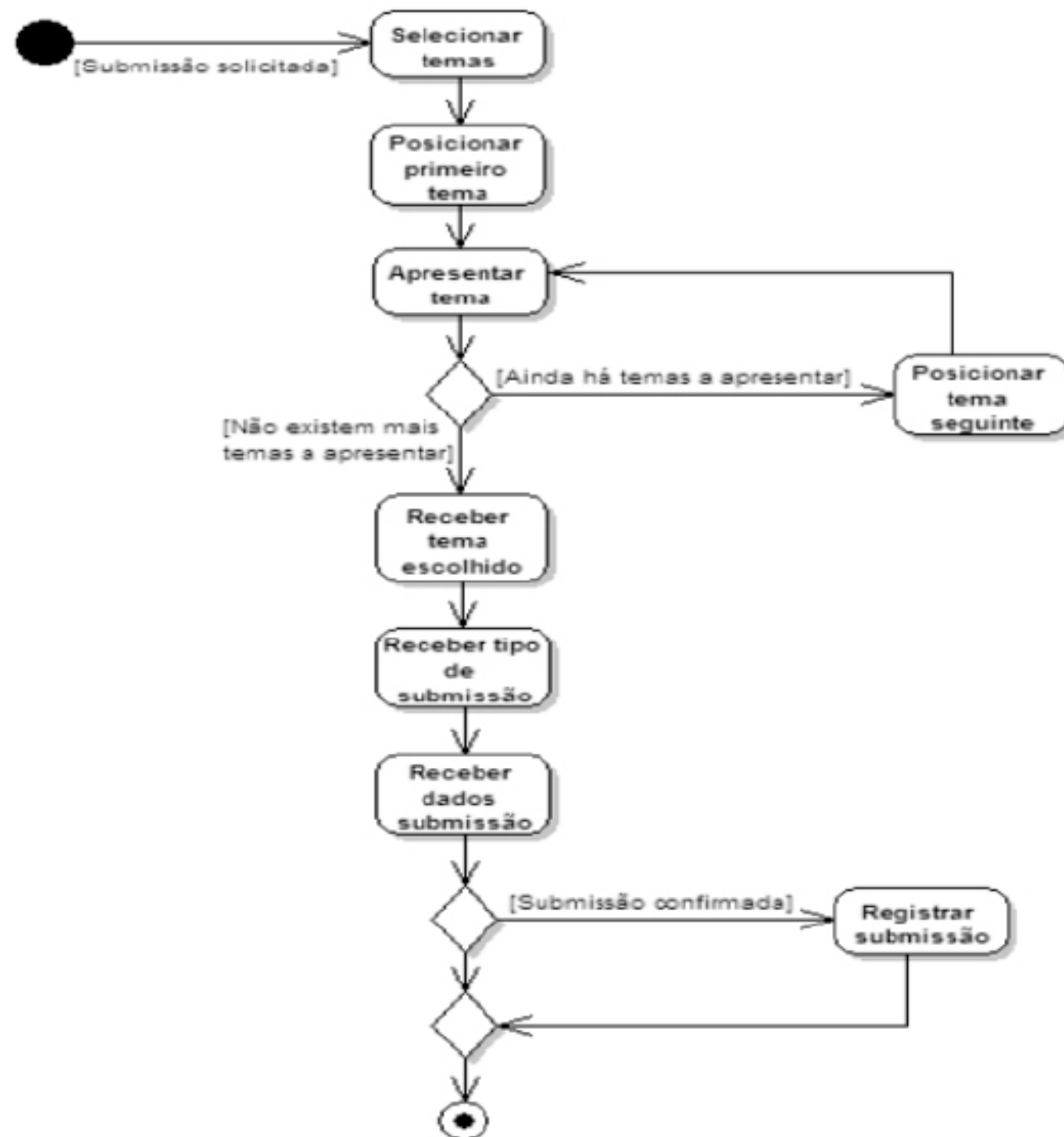
Considerado um dos diagramas mais importantes da UML e um dos mais utilizados, ele serve de apoio para a maioria dos outros diagramas. Como o próprio nome já diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si.

Devido à sua importância, o detalharemos mais adiante.

DIAGRAMAS COMPORTAMENTAIS

Diagramas de atividade

Se preocupa em descrever os passos a serem percorridos para a conclusão de uma atividade específica, muitas vezes representada por um método ou algoritmo com um certo grau de complexidade, podendo, no entanto, modelar um processo completo. Concentra-se na representação do fluxo de controle e no fluxo de objeto de uma atividade.



Diagramas de Caso de Uso

É o diagrama mais geral e informal da UML, sendo utilizado especialmente para auxiliar no levantamento e na análise dos requisitos, em que são determinadas as necessidades do usuário, e na compreensão do sistema como um todo, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para a maioria dos outros diagramas.

Esse diagrama utiliza uma linguagem simples e de fácil compreensão, para que os usuários possam ter uma ideia geral de como o sistema irá se comportar. Ele procura identificar os atores (usuários, outros softwares que interajam com o sistema ou até mesmo algum hardware especial), que utilizarão de alguma forma o software, bem como as funcionalidades, ou seja, as opções que o sistema disponibilizará aos atores, conhecidas nesse diagrama como Casos de Uso.



DIAGRAMAS DE INTERAÇÃO

Diagrama de Temporização

Também conhecido como Diagrama de Tempo, descreve a mudança no estado ou na condição de uma instância de uma classe ou seu papel durante um tempo. É tipicamente utilizado para demonstrar a mudança no estado de um objeto no tempo em resposta a eventos externos.

