 

**INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS**

**Campus Almenara - MG**

**Análise e Desenvolvimento de Sistemas**

**CARLOS ROBERTO NEVES LACERDA**

**BACKLOG 27.02.2023**

Almenara-MG

2023



**CARLOS ROBERTO NEVES LACERDA**

**BACKLOG 27.02.2023**

Trabalho para apresentação da disciplina:

Análise de Projeto de Software.

Professor(a): Pablo Castro.

Almenara-MG

2023

 

**Nas aulas anteriores foi discutido sobre os requisitos do sistema, o que eles são e pra que eles servem. Desse modo, é considerado o nível de abstração próximo do problema e de seus usuários. Contudo, podemos identificar o código fonte como uma representação de baixo nível e executável do comportamento de um sistema. Com isso, vemos que existe uma lacuna entre os requisitos e código fonte. Para solucionar esses problemas, alguns Engenheiros de Software investiram na inversão de modelos padrões, na qual, foram criados para ajudar na compreensão e na análise de um sistema. Para isso, foi definido que os modelos deveriam ser mais detalhados que os requisitos e mais simples que o código fonte. Segundo o estatístico britânico George Box “A complexidade de um software é uma propriedade essencial e não acidental. Portanto, representações de uma entidade de software que abstraem sua complexidade normalmente também abstraem sua essência [...]. Esse paradigma funcionou porque as complexidades ignoradas não são propriedades essenciais do fenômeno sob estudo. Porém, essa abordagem não funciona quando as complexidades são essenciais”. Essa frase se refere aos modelos matemáticos que foram baseados nos modelos de engenharia de software, porém ela se aplica muito bem a ambos, visto que os modelos se tornam errados, pois tentam simplificar ou retratar a realidade. Por isso, a principal questão consiste em esmar se o modelo compete as abstrações do sistema mesmo com a simplificação do mesmo.**

**Com a evolução dos modelos de software foi desenvolvido o modelo UML, que pode ser defini como uma forma de modelagem de software em diagramas. Essa documentação auxilia no desenvolvimento do design de sistemas, ela faz funciona como esqueleto no qual os desenvolvedores iram trabalhar assimilando.**

Almenara – MG

2023

** **

**Quando a UML surgiu na década de 80 tinha o propósito de criar modelos gráficos, que seriam um auxílio para os programadores, com intuito de serem convertidos em código fonte. Para efeito de compreensão, podemos fazer uma analogia entre o desenho feito no Auto Cad pelo os Engenheiros Civis e o diagrama de UML feito pelos Engenheiros de Software ambos desenvolvidos para o auxílio no desenvolvimento do projeto.**

**O diagrama UML é dividido em dois grupos, sendo eles;** **Diagramas Estáticos e Diagramas Dinâmico. Os Diagramas Estáticos são referentes a modelagem da estrutura e da organização do sistema, nele são inclusas as informações sobre os atributos, as classes, os métodos e etc. Por outro lado, temos os Diagramas Dinâmicos que remetem a modelagem de eventos podem ocorrem durante a execução de um sistema. Além disso, temos outras espécies de diagramas, como por exemplo o Diagrama de Classes que fornecem uma representação gráfica do conjunto de classes, contendo as informações dos atributos, dos métodos e dos relacionamentos entre as classes. Contudo, o diagrama UML possui as Associações entre as classes que basicamente é o compartilhamento de atributos, na qual uma classe A possui um objeto b de uma classe B, nesse caso, A está associado a B. Além disso, podemos identificar a Herança presente no diagrama de UML que está relacionada ao que uma subclasse herdará de uma classe, nesse caso uma classe filha herdará atributos de uma classe pai. Por fim, temos a Dependências que ocorre quando um método de uma classe A declara um parâmetro ou variável local do tipo B, nesse caso ocorre a dependência do mesmo.**

**Com isso, podemos concluir que o diagrama UML é estritamente importante no desenvolvimentos de um sistema visto que ele auxilia os desenvolvedores na criação do código fonte.**

Almenara – MG

2023

** **

**REFERÊNCIAS**

https://engsoftmoderna.info/cap3.html