ANÁLISE DE SISTEMAS ORIENTADO A OBJETO

A modelagem do sistema foi realizada utilizando diagramas UML (Unified Modeling Language), que incluem diagramas. A abordagem orientada a objetos permite representar entidades do mundo real, como administradores, funcionários, técnicos e chamados, como objetos com atributos e comportamentos definidos.

Na modelagem de requisitos utiliza-se o método ágil “SCRUM”, em que auxilia no desenvolvimento do sistema, através da separação de tarefas, e das “sprints” semanais, que são entregas recorrentes. Com isso adere-se uma metodologia flexível, em que se há necessidade de alteração, tem a facilidade de alterar dados já implementados. A qualidade do sistema é avaliada por meio de testes tanto automatizados quanto manuais, que vão desde testes de unidade até testes de aceitação pelo usuário. A verificação do sistema é feita em conjunto com os stakeholders, assegurando que as funcionalidades correspondam às exigências e expectativas previamente estabelecidas.

Implementando a UML para a modelagem de requisitos, de acordo com Grady Booch (um dos criadores da UML), ele enxerga a UML como uma linguagem padronizada para visualizar, especificar, construir e documentar os artefatos de um sistema de software — incluindo os requisitos. Utiliza-se quatro tipos de diagramas essenciais, o diagrama de caso de uso, diagrama de classes, diagrama de sequência e o diagrama de implantação. O diagrama de caso de uso é fundamental para a modelagem de requisitos, pois apresenta todos os requisitos essenciais do sistema em forma de casos de uso, e também os atores relacionados, assim facilitando a visualização da estrutura completa do sistema.

O diagrama de classe é importante, pois exibe de forma clara as classes, seus atributos, métodos e os relacionamentos, tornando mais fácil visualizar a estrutura estática do sistema, também facilita o entendimento de como uma parte do código se relaciona com outra. Já o diagrama de sequência tem como objetivo mostrar como os objetos interagem entre si para realizar os requisitos descritos nos casos de uso, utilizando as classes do sistema, ele ilustra o passo a passo a troca de mensagens entre os objetos, permitindo entender o fluxo de execução de um processo.

Por fim, o diagrama de implantação é importante para facilitar a compreensão da infraestrutura necessária para suportar um sistema, planejar a implementação e a distribuição do software em diferentes ambientes, também ajuda na identificação de possíveis falhas.

REFÊRENCIAS

Andrey Khusid – “O guia definitivo para diagramas UML” – 2025 - Disponível em: <https://miro.com/pt/diagrama/o-que-e-uml/>

Acesso: 25/04/2025.

Pearson Prentice Hall – “Modelagem de Sistemas” - Postado: 2011 – Disponível em: <https://homepages.dcc.ufmg.br/~cesarfmc/classes/es/Capitulo_05.pdf>

Acesso: 25/04/2025.

Business Machines Corporation – “Diagramas de Sequência” – Postado:05/03/2021 – Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=uml-sequence-diagrams>

Acesso: 25/04/2025.

Object Management Group -"Implantações". Linguagem de Modelagem Unificada 2.5.2 - Postado: 12/2017 – Disponível em: <https://www.omg.org/spec/UML/2.5.1/PDF>

Acesso: 25/04/2025.

PRESSMAN – “R. S. Engenharia de Software: uma abordagem profissional. 7” - Postado: 2011 – Disponível em: <https://www.scribd.com/document/347525214/Engenharia-de-Software-Pressman-2011-pdf>

Acesso em: 25/04/2025.

SOMMERVILLE, Ian – “Engenharia de software. 9” – Postado: 2011 – Disponível em: <https://www.amazon.com.br/Engenharia-Software-Ian-Sommerville/dp/8521632112>