

PROJETO DE SISTEMAS ORIENTADO A OBJETOS

Nesta fase, aqueles princípios, métodos e boas práticas da disciplina Projeto de Sistemas Orientado a Objetos foram implementados, tomando Versolatto (2015) e Pressman (2006) como referência para estruturar tecnicamente o desenvolvimento do Sistema Integrado de Gestão de Chamados e Suporte Técnico com Apoio de IA.

A meta era converter os requisitos do PIM III em uma solução de software bem organizada, modular, além de fácil manutenção, assegurando que o sistema funcionasse de maneira eficiente, segura e, ademais, escalável.

Aplicação dos Conceitos Fundamentais no Projeto

Durante a fase de design, os conceitos de abstração, modularidade, encapsulamento e também a independência funcional, foram implementados na prática na definição da arquitetura e dos componentes do sistema.

Abstração: cada entidade do sistema foi modelada representando só as informações essenciais para o seu bom funcionamento. Como exemplo, a classe Chamado resume os principais atributos de um atendimento (ID, data, status, descrição e técnico responsável), enquanto detalhes menos importantes foram isolados em outras classes auxiliares, Histórico e Categoria por exemplo.

Modularidade: O sistema foi dividido em módulos independentes tipo, Módulo de Usuários, Módulo de Chamados, e mais alguns tipo Módulo de FAQ Inteligente e o de Relatórios. Essa divisão, ela assegura um acoplamento baixo, uma coesão alta, para fazer a manutenção do sistema.

Encapsulamento: Cada classe, ela tem seus próprios métodos e os atributos privados, que são acessados por interfaces e métodos públicos. Isso trouxe mais segurança e isolou a lógica do negócio.

Independência Funcional: Os componentes foram planejados para cada um fazer o seu trabalho, por exemplo, o módulo de IA de Suporte que serve só para sugerir soluções automáticas, sem interferir nos chamados.

Esses princípios, seguindo a definição lá do Versolatto em 2015, ajudaram a deixar o sistema mais organizado, diminuindo os erros e facilitando a manutenção.

Fases do Projeto Implementadas no Sistema

Inspirando-se no modelo de projeto proposto por Pressman (2006) e corroborado por Versolatto (2015) as quatro fases de projeto orientado a objetos foram aplicadas desta forma

a) Projeto de Dados e Classes

Elaborou-se o Diagrama de Classes UML que mostra as entidades chave: Usuário Técnico Chamado Categoria Histórico FAQ e IA.

Cada classe detalha seus atributos e métodos e as relações de herança e associação foram utilizadas para ilustrar a hierarquia entre usuários e técnicos.
O banco de dados SQL Server foi modelado a partir desse diagrama, a fim de garantir a ligação entre as classes do sistema e as tabelas relacionais.

b) Projeto Arquitetural

A arquitetura escolhida foi multicamadas, com camadas de apresentação de negócio e persistência aderindo ao padrão Model-View-Controller MVC.
Essa estrutura propicia a divisão da lógica de negócio da interface e do banco de dados otimizando o desenvolvimento em grupo e a execução de testes.
Cada camada se comunica exclusivamente por meio de interfaces claras e bem definidas favorecendo o baixo acoplamento e o alto reaproveitamento de código.
c) Projeto de Interfaces

As interfaces, pensadas cuidadosamente, almejam proporcionar uma usabilidade impecável e uma vivência do usuário bem agradável, seguindo os fundamentos de usabilidade detalhados por Versolatto (2015) e Nielsen (1993).
Elas se dividem em:

Interface Web, feita com ASP NET e Blazor, com um design responsivo e autenticação para garantir a segurança;

Interface Desktop, focada nos administradores, ela é em C# com Windows Forms;

Interface Mobile, foi feita pro Android, e que permite a abertura e o acompanhamento dos chamados.

d) Projeto de Componentes

Cada módulo funcional foi retratado por um diagrama de componentes UML, onde a comunicação entre as partes do sistema fica bem clara.
Os componentes principais são:

Componente de Gerenciamento de Chamados;

Componente de FAQ Inteligente, que usa IA pra dar umas respostas;

Componente de Relatórios, responsável por fazer os gráficos e as estatísticas;

Componente de Banco de Dados, para salvar tudinho usando SQL Server.

3 Aplicação dos Atributos de Qualidade ISO 25010

O projeto abraçou as diretrizes da ISO/IEC 25010, visando qualidade em varios aspectos do sistema.

Funcionalidade: o sistema cumpre os requisitos de registro, acompanhamento, e solução de chamados.

A confiabilidade de um banco de dados centralizado mantém a integridade das informações.

Usabilidade: interfaces padronizadas e intuitivas facilitam uma navegação suave.

Eficiência: otimização do código com consultas SQL parametrizadas e carregamento assíncrono.

Manutenibilidade: arquitetura modular permite alterações, bem localizadas.

Portabilidade: o uso de .NET e Blazor possibilitam uma futura migração para várias plataformas.

Estas práticas garantem a evolução do sistema, sem prejudicar a estabilidade e desempenho.

Integração com Inteligência Artificial

No contexto do projeto, o paradigma orientado a objetos foi essencial para integrar o módulo de IA.

A classe IAFAQ foi projetada, como um componente independente, capaz de analisar o histórico de chamados, e sugerir respostas automáticas.

O módulo emprega polimorfismo, adaptando respostas ao tipo do problema relatado, aplicando herança para comportamentos específicos de classes genéricas

RespostaAutomatizada e SugestãoIA.

Dessa maneira, a IA se encaixa perfeitamente na estrutura orientada a objetos do sistema.

Conclusão da Aplicação

Aplicar os conceitos do Projeto de Sistemas Orientado a Objetos possibilitou o desenvolvimento organizado, escalável e alinhado com as boas práticas de engenharia de software do sistema.

As escolhas arquitetônicas com base em UML, modularidade e camadas, foram chave para clareza na comunicação entre os devs e para facilitar a manutenção.

Em suma, o projeto transforma a disciplina em ação, exibindo como o planejamento técnico e a modelagem orientada a objetos são cruciais para o sucesso de um sistema real de suporte técnico com inteligência artificial.

PRESSMAN, Roger S. *Engenharia de Software: uma abordagem profissional*. 6. ed. São Paulo: McGraw-Hill, 2006.

VERSOLATTO, Fábio Rossi. *Projeto de Sistemas Orientado a Objetos*. São Paulo: Editora Sol, 2015.

SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2011.

ISO/IEC 25010:2011. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)*.

NIELSEN, Jakob. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1993.