

PROJETOS DE SISTEMAS ORIENTADOS A OBJETOS

Ao projetar e implementar um sistema de administração para um pequeno empreendimento, especialmente um sistema de automatização total da folha de pagamento, é possível se beneficiar de diversos recursos e elementos essenciais orientados a objetos. (Apex, 2017)

Os principais dos recursos são as Classes e Objetos, são os fundamentos da programação orientada a objetos, permitindo a modelagem de entidades do sistema. As classes definem atributos e métodos, enquanto objetos são instâncias dessas classes, ambos facilitam a representação de entidades do mundo real, como funcionários, pagamentos e impostos e a evolução pode envolver a adição de novos atributos e métodos conforme as necessidades do sistema crescem. (FARINELLI, 2007)

O encapsulamento é outro elemento essencial, ele protege o estado interno dos objetos, ocultando detalhes de implementação e melhora a manutenibilidade do código, permitindo alterações internas sem afetar o uso externo. Os atributos são definidos como privados e o acesso a eles é feito por métodos públicos (getters e setters), à medida que o sistema cresce, novos atributos podem ser encapsulados para manter a integridade dos dados. (JUNGTHON & GOULART, 2009)

A herança permite criar hierarquias de classes, compartilhando características comuns entre elas. Nela uma classe filha (subclasse) herda atributos e métodos da classe pai (superclasse), reduzindo a duplicação de código, além de promover a reutilização e permitir modelar relacionamentos hierárquicos, como diferentes tipos de funcionários. O polimorfismo permite que objetos de classes diferentes respondam ao mesmo método de maneira única através da implementação de métodos com a mesma assinatura, e vale ressaltar que facilita a criação de código genérico e flexível, simplificando a lógica de processamento. (Aguiar, 2015)

Outro ponto importante é a interface, a qual define contratos que as classes devem cumprir, permitindo a criação de código mais flexível, melhorando a criação de componentes reutilizáveis, tornando o sistema mais modular. Neste caso as classes podem implementar múltiplas interfaces, obrigando-as a fornecer implementações para os métodos definidos na interface e conforme

novas funcionalidades são adicionadas, novas interfaces podem ser introduzidas. (BARANAUSKAS, 1993)

Já as coleções, como listas, conjuntos e mapas são essenciais para armazenar e manipular dados em grande quantidade e como característica técnica é possível destacar que diferentes tipos de coleções têm diferentes estruturas e métodos. Uma das vantagens é que simplificam a gestão de dados, especialmente em casos como o armazenamento de históricos de pagamento e à medida que o volume de dados cresce, a escolha da estrutura de coleção mais apropriada pode evoluir. (Apex, 2017)

A persistência de dados é necessária para armazenar informações importantes, como registros de pagamento e históricos, pode ser implementada com bancos de dados relacionais ou não relacionais, porém no caso deste projeto foi implementada com banco de dados relacional MS SQL Server. Garante a integridade e a disponibilidade dos dados, e futuramente pode ser necessário escalar a infraestrutura de armazenamento de dados. (Melo, 2016)

Vale ressaltar que os frameworks e bibliotecas orientados a objetos podem acelerar o desenvolvimento e fornecer soluções predefinidas para problemas comuns, variam dependendo da linguagem e do contexto. Para este caso foram utilizados os frameworks ASP.NET, Ionic e Angular, os quais economizam tempo de desenvolvimento e garantem boas práticas, além de gerar o aproveitamento de novos recursos e melhorias futuramente. (Souza & Silva, 2021)

Por fim, uma estratégia extremamente importante é a documentação clara, essencial para entender o funcionamento do sistema e facilitar a colaboração, onde foi realizada a inclusão de comentários no código, documentação de API e manuais de usuário. Ela melhora a comunicação entre os membros da equipe e reduz o tempo de treinamento, mas deve ser atualizada à medida que o sistema é aprimorado e novos recursos são adicionados. (JUNGTHON & GOULART, 2009)