

Documentação Técnica

1. Descrição do problema

O objetivo é consolidar o conhecimento sobre conceitos e construção de sistemas web (cliente-servidor, web/mvc e serviços) empresariais orientados a objetos em arquiteturas cliente-servidor multicamadas através da exploração dos tópicos discutidos na disciplina de Programação de Software Aplicado. Para isso será criado um pequeno site que simula o funcionamento de um comércio virtual, onde o usuário pode anunciar itens para venda e outro usuário podem comprá-los, fazer perguntas e avalia-los. Também é possível ler um QR code que é gerado no momento da solicitação de compra, para acompanhar o andamento da venda.

2. Padrões de projeto:

- Interface web

A interface web foi feita em Vue, com Bootstrap.

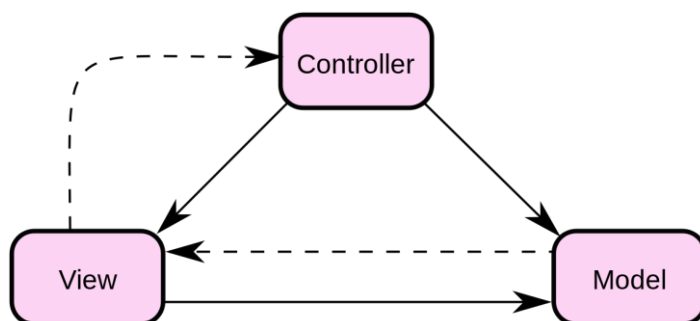
- Arquitetura limpa ou multicamada (pelo menos 3) com separação de responsabilidades.

O projeto tem a camada de apresentação, a camada de domínio, a camada de dados.

- Uso dos padrões de projeto explorados nas disciplinas de Fundamentos de Desenvolvimento de Software

Foi utilizado o padrão de repository, MVC, domain model, herança e SOLID

- Uso do padrão MVC na camada de apresentação;



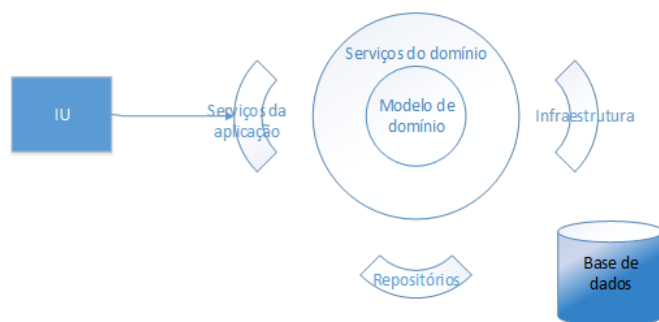
Foi criado um projeto usando .NET nesse padrão. O sistema é totalmente dividido nesse modelo. Temos a View, representada pela interface web, o controller, que é o nosso aplicativo em .Net propriamente dito e os modelos, que é o banco em POSTGRESQL.

- Uso do padrão “Facade” para isolar a camada de domínio da camada de apresentação;

É utilizado classes chamadas de Services, que isolam a camada de dados do resto, e ainda fornecem serviços que podem ser usados por todo o sistema.

- Uso do padrão arquitetural “Domain Model” na camada de domínio;

Esse padrão é totalmente utilizado, visto que temos uma interface gráfica, representado pela aplicação web, um servidor que disponibiliza serviços, através dos controllers, os repositórios que fazem a conexão com o banco e o próprio banco de dados, que usando POSTGRESQL



- Uso do padrão “DAO”/”Repository” na camada de persistência.

Foi utilizado o padrão repositoy, onde é criado uma classe principal com os métodos genéricos, e o repository de cada entidade do sistema herda dessa classe, com todos os métodos já implementados

- Persistência em base de dados relacional.

Foi utilizado o POSTGRESQL como banco de dados para armazenamento

- A camada de persistência deve ser implementada utilizando um mapeador objeto-relacional.

Foi utilizado a própria ferramenta do .Net para criar o mapeamento. Essa ferramenta é o Migrations e o LINQ, que permite que as classes no pacote Model sejam transformadas diretamente em tabelas do banco. Com o LINQ, é possível conectar ao banco e acessar esses dados

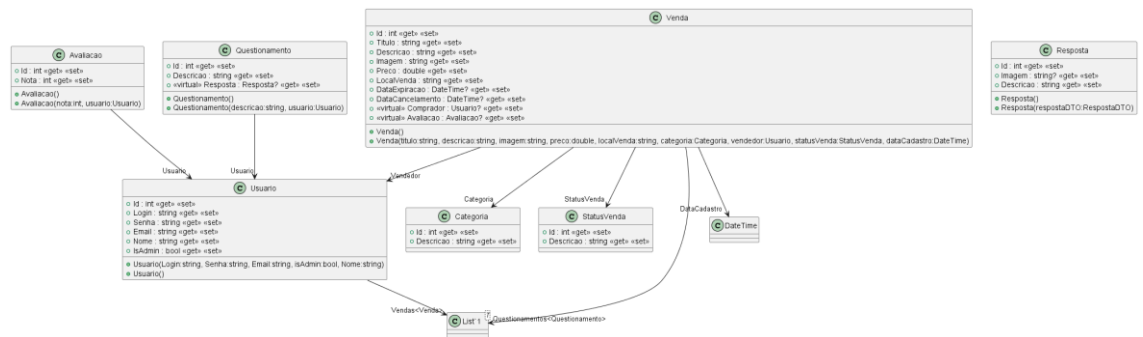
- Tratamento correto do encapsulamento de exceções entre as camadas.

Todos os erros são tratados corretamente pelo sistema, nenhum erro é jogado pra o usuário e nem erros na camada de apresentação se refletem no servidor. Um exemplo disso é quando se tenta fazer um login com uma senha errada, onde o sistema flui naturalmente.

- O acompanhamento do status de uma venda deve poder ser acompanhado via um serviço utilizando WebAPI

No momento da compra, é gerado um QR code utilizando a API gratuita disponibilizada pela Google e esse QR code é enviado por e-mail usando a biblioteca do Mailkit

3. Diagrama de classe



4. Diagrama relacional

Esse diagrama esquematiza o banco de dados, que está usando um modelo de entidade relacional.

