

Documento de Arquitetura do Sistema Pizza.Net

Marcus Vinicius Marchiori Ribeiro - 11811BSI224
Matheus José dos Reis – 11811BCC027
Matheus Humberto G. F. Costas - 11811BCC006
Nayara Terezinha Nunes – 11911BCC006
Thiago Antônio Moreira -11411BSI235
Victor Hugo Marra Araújo - 11721BCC041

Uberlândia – MG
Setembro de 2020

1. Sistema PizzaNet

O sistema permitirá a realização, gerenciamento e entrega de pedidos da Pizzaria Entrega Rápida, o sistema poderá ser acessado pela web ou por aplicativo para dispositivos móveis. Possui fácil utilização, devido ao público alvo possuir pouco conhecimento técnico em informática. A ferramenta não visa substituir profissionais, mas sim facilitar o trabalho dos funcionários e aumentar a eficiência do serviço prestado.

2. Necessidades do sistema

As principais necessidades que o sistema deve atender compreendem:

- **Gerente:** pode manter o cadastro de pizzas e bebidas com preço, sabores e tamanho, acompanhar o andamento de todas as entregas em tempo real pelo Maps.
- **Atendente:** pode registrar o recebimento de pedidos por telefone, alocar um entregador ao pedido do cliente.
- **Cliente:** pode montar sua pizza selecionando os ingredientes que desejar, e o valor a ser pago pela pizza será calculado a medida que os ingredientes forem selecionados.
- **Pedidos:** pela internet, encaminharão o cliente para preencher seu cadastro previamente por meio de telefone. Pedidos por telefone devem ser pagos em dinheiro, o sistema calculará e encaminhará o valor da conta e troco. Pedidos feitos pela internet terão pagamento diretamente na Web por meio de cartão de crédito, antes da finalização do pedido. Pedidos de pizza com mais de um sabor devem ser cobrados pelo maior valor entre os tipos de pizza.
- **Entrega:** preço será calculado de acordo com a distância do local de entrega do pedido. A cobrança feita é de R\$ 0,50 para cada quilometro. A distância para entrega deve ser obtida pela API do Google Maps.

3. Tecnologias

Para o desenvolvimento do sistema serão utilizadas as seguintes especificações das tecnologias:

- O back-end do sistema será hospedado em provedor que mantém 96% de disponibilidade como a AWS, Azure e etc;
- O back-end será feito em Java com Spring Boot;
- Persistência do banco Spring Data, JPA e Hibernate;
- Projeto de padrão REST;
- Arquitetura MVC;
- O banco de dados será o PostgreSQL;
- O app será feito em Dart com o framework Flutter;
- A documentação do sistema deve seguir o padrão Swagger;
- Metodologia orientada a objetos.

3. Requisitos de software

Essencial	Funcionário cadastra cliente
Essencial	Gerente mantém produtos (cadastrar e alterar)
Essencial	Atendente registra pedidos de clientes
Essencial	Cliente registra pedido pela Web
Essencial	Gerente mantém entregadores
Essencial	Cliente cadastra "sua pizza"
Essencial	Cliente acompanha entrega
Essencial	Gerente acompanha entrega
Essencial	Funcionário aloca entregador
Importante	Entregador registra entrega do pedido
Importante	Cliente altera pedido
Importante	Funcionário registra alteração no pedido
Importante	Funcionário cancela pedidos
Desejável	Cliente consulta prazo previsto de entrega
Desejável	Funcionário consulta histórico de pedidos
Não Funcional - Essencial	Pedidos por telefone devem ser pagos em dinheiro

Não Funcional - Essencial	Executar app em dispositivo móvel
Não Funcional - Essencial	O sistema deve ser acessado pela web
Não Funcional - Essencial	Usar o Maps para fazer localização de entregadores e local de entrega
Não Funcional - Essencial	O usuário devem fazer login no sistema (cliente, funcionário, gerente e entregador)
Não Funcional	Gerente cadastra funcionário

Tabela01 – Requisitos PizzaNet

4. Casos de Uso

Os casos de uso apresentam serviços ou funções que podem ser utilizados por usuários do sistema, e relação entre eles.

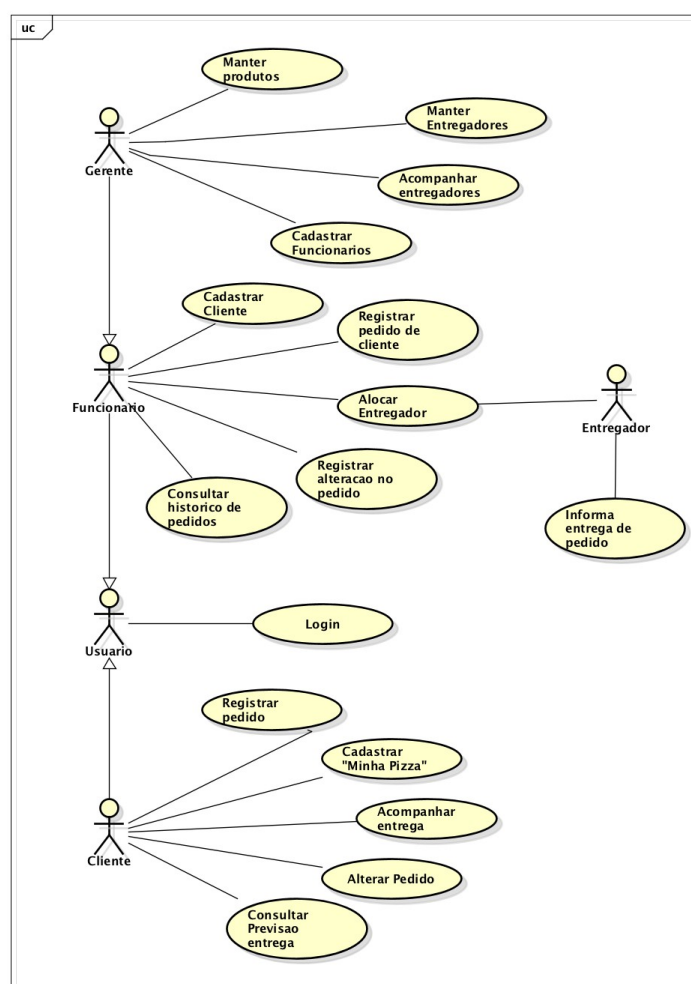


Figura01 – Casos de uso PizzaNet

5. Diagrama de classe de análise

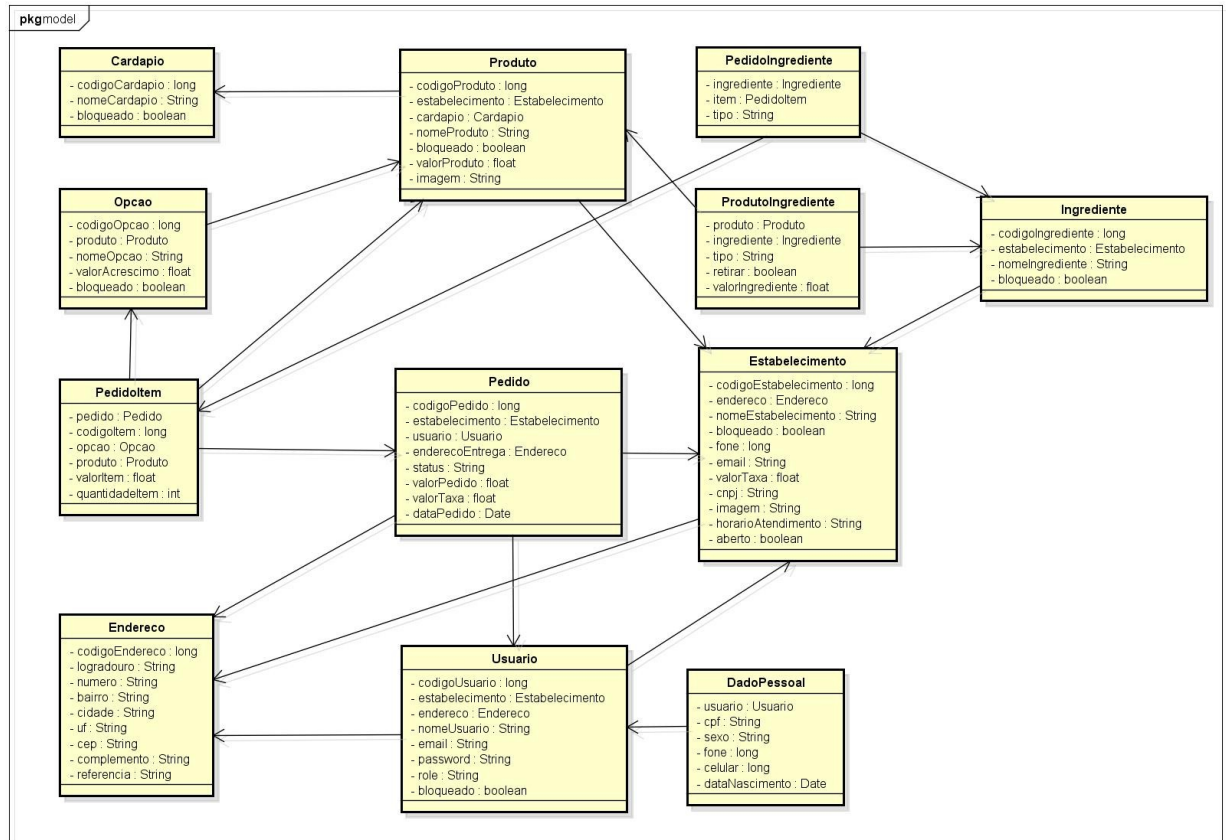


Figura 02 - Diagrama de classe de análise do sistema

6. Diagrama entidade relacionamento

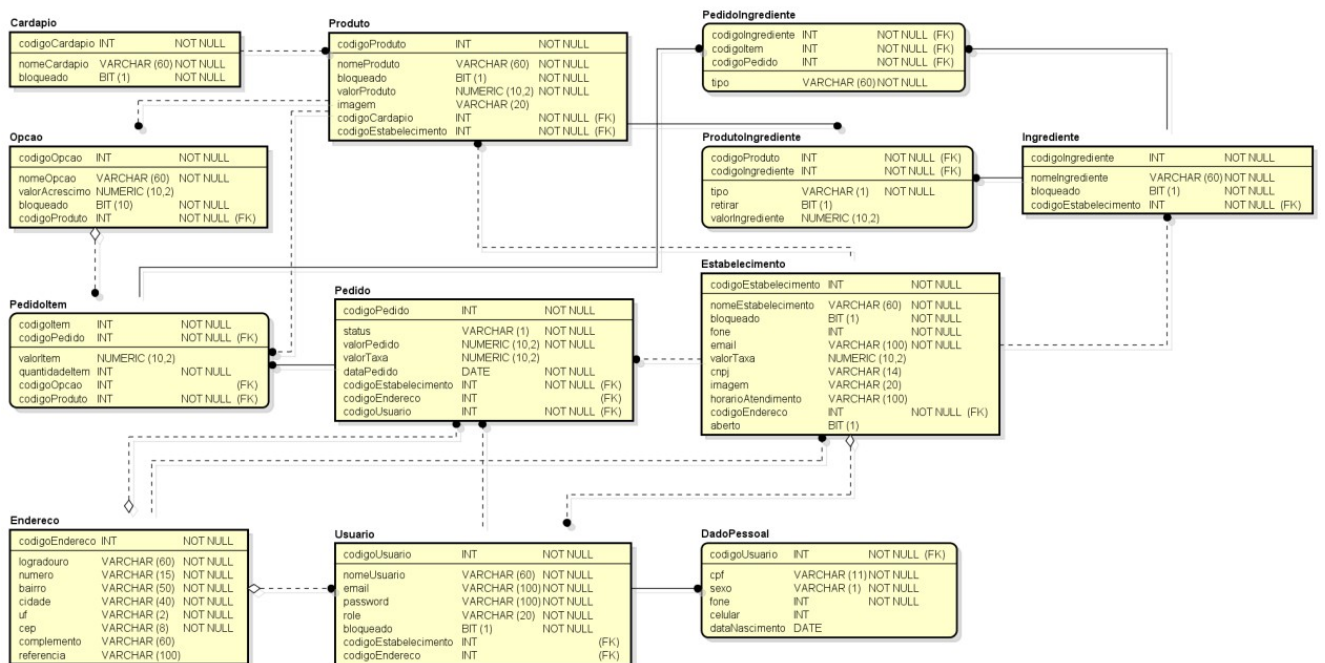


Figura 03 - Diagrama ER do sistema

7. Divisão dos pacotes seguindo MVC

Os pacotes de view irão conter as classes e projetos do front-end/web/mobile, todas as classes que serão expostas e apresentadas para os usuários do sistema;

Os pacotes de model irão conter a camada de serviço e modelagem, regras de negócios e cumprimento dos requisitos funcionais;

Os pacotes de controller, irão conter os web containers e JSP servlets de páginas HTML.

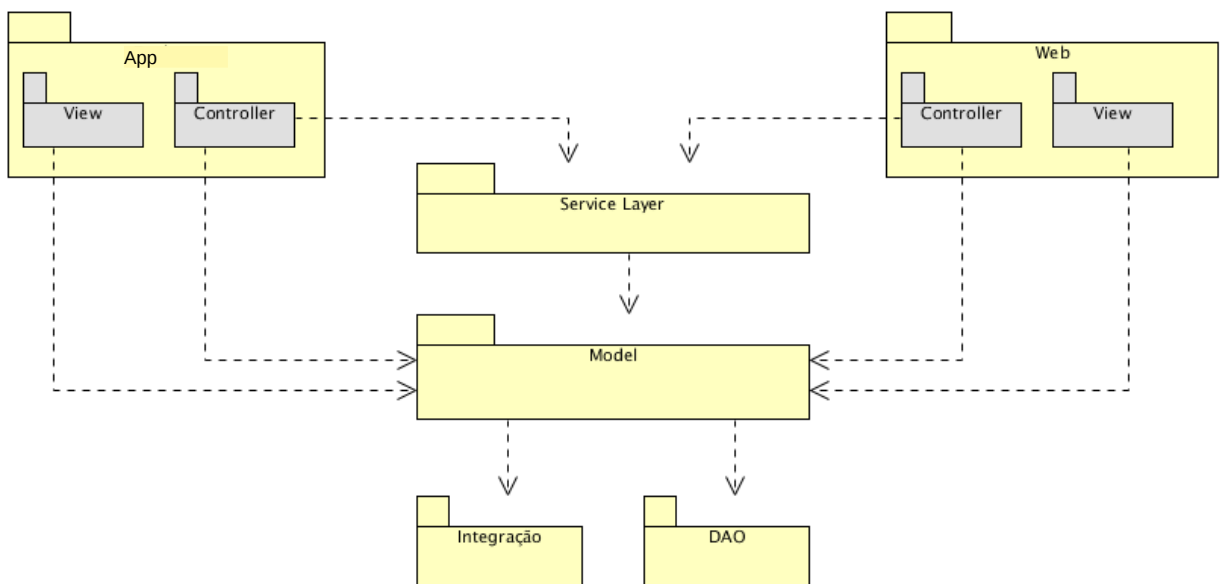
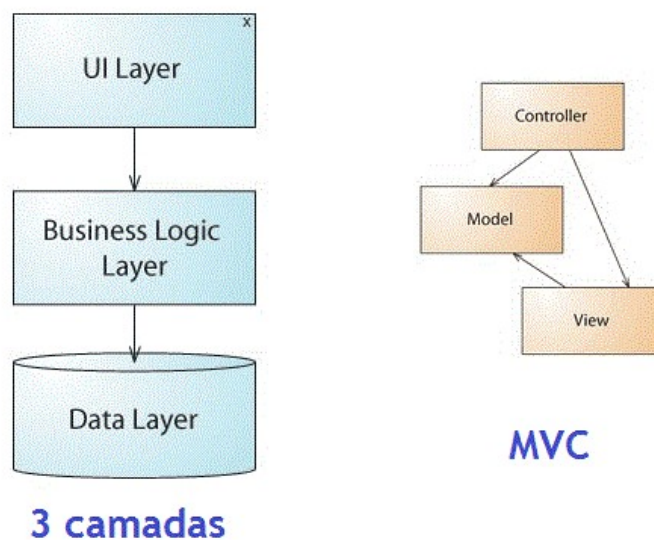


Figura 04 – Modelo Visualização Controle



3 camadas

MVC

Figura 05 – Padrão MVC

8. Diagrama de componentes

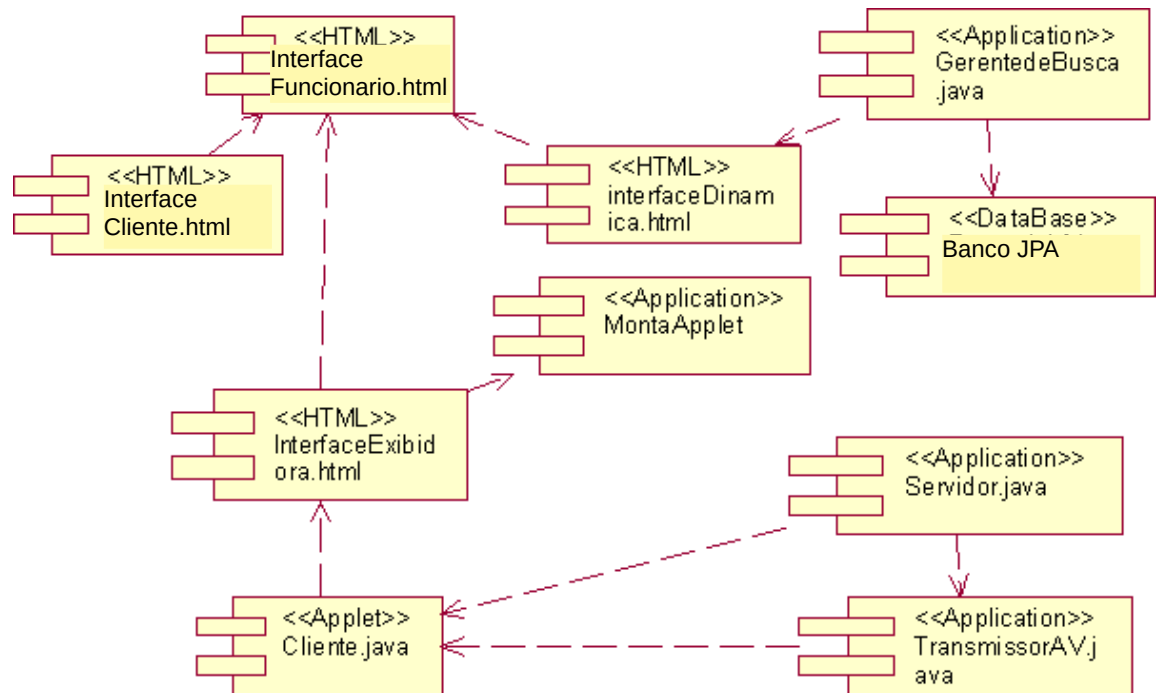


Figura 06 – Componentes

Referências bibliográficas:

- AMARAL, Rafael. Sistema web para pedidos de delivery, 2014. Acesso em: 27/09/2020.
- GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec, 2009. Acesso em: 27/09/2020
- LARMAN, Craig. Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado. 2. ed. Porto Alegre: Bookman, 2004. Acesso em: 27/09/2020