

Danilo Carvalho⁽¹⁾, Gabriel Mendes⁽²⁾, Guilherme Rodrigues⁽³⁾, Jorge Lucas⁽⁴⁾, Marcelo D Angelo⁽⁵⁾, Nicholas Domingues⁽⁶⁾, Professora Orientadora: Viviane Gerardi - ⁽¹⁾R.A 00335596, ⁽²⁾R.A 00250615, ⁽³⁾R.A 00249962, ⁽⁴⁾R.A 00264010, ⁽⁵⁾R.A 00250627, ⁽⁶⁾R.A 00244658.

Resumo

O artigo a seguir apresenta a criação de um aplicativo no qual a problemática são as dificuldades financeiras e sociais que a Covid-19 impôs aos comércios afetando assim, seu funcionamento. Para a solução, é a criação de um aplicativo onde a padaria cadastrará seu comércio e catalogue seus produtos no sistema, para então, o cliente compra-lo online. Desta forma, as vendas serão feitas seguindo as regras de distanciamento social e facilitará a vida do comerciante que tem dificuldades de trabalhar sem interação social. Também aborda análise e desenvolvimentos com linguagem de programação em C.

Palavras-Chave: Linguagem C, análise e desenvolvimento de sistemas, tecnologia da informação.

1. Introdução

As padarias são um serviço essencial, presente em cada cidade, distrito ou bairro, independente das características econômicas, estão passando por momentos sensíveis diante da pandemia do novo coronavírus [3]. Elas atuavam 80% offline e precisam converter e agregar urgentemente para o modelo online. A padaria aumentou com o passar dos anos sua estrutura de atendimento chegando a oferecer refeições diárias no almoço e jantar, além de produtos variados se tornando uma loja de conveniência onde se acha praticamente tudo que um mercado tem ou que você necessita. O departamento de serviços de alimentação era o que mais crescia dentro da padaria e ele precisou ser interrompido, isso impactou muito nos negócios [4]. Aos poucos as padarias tentam se adaptar a esse novo cenário com fluxo baixo de clientes e abertura gradual das áreas internas.

Segundo coordenador de Pesquisas Estruturais e Especiais em Empresas do IBGE Alessandro Pinheiro, por segmento o maior percentual de empresas em que a pandemia tem tido efeito negativo está no setor de Serviços (74,4%). Os dados sinalizam que a Covid-19 impactou mais fortemente em segmentos que para a realização de suas atividades é imprescindível o contato pessoal [1]. A queda no lucro de padarias variou de 40% a 50% durante isolamento social, segundo Sindicato das Indústrias de Alimentação de Brasília (Siab). Para evitar prejuízos, empresários apostaram em novos produtos e no delivery [2]. Com dificuldade de acesso a crédito e medidas de socorro dos governos, panificadores buscam mecanismos para reduzir impactos da crise e a modernização dos negócios.

A proposta de solução para minimizar esta situação levantada é a criação de um aplicativo baseado em Linguagem C, que funcionará como um delivery para pequenas e médias padarias, junto com seus clientes que moram no bairro, nele os clientes que frequentam a padaria se cadastram de forma simples e rápida no sistema conseguindo assim fazer seus pedidos no conforto e na segurança de sua casa

sem interação social (no app o usuário poderá procurar todos os tipos de produtos da padaria).

Pensando na acessibilidade onde todas as padarias mesmo as pequenas, que estão de portas fechadas, possam continuar funcionando e vendendo seus produtos, para que com isso consigam se manter nessa situação atual tão delicada.

Com os clientes cadastrados no sistema, as padarias podem ser mais assertivas nas ofertas e promoções, que seriam enviadas diariamente, assim como também cupons de desconto, incentivando e promovendo o uso do aplicativo. Um aplicativo voltado a ajudar na comunicação entre a padaria e o cliente, ao invés de um sistema administrativo, pois a grande maioria das padarias já possuem sistemas para controle e cobrança. A possibilidade de entrega seria uma opção também pois nesse cenário pandêmico atual o sistema de delivery tem sido essencial. Optar por um aplicativo é a forma mais segura e prática, além de a implantação ser mais acessível e barata para a padaria, inclusive ajudar as pequenas padarias de bairros onde o fluxo de clientes é alta, evitando assim aglomerações.

Diante dessa problemática, justifica-se o desenvolvimento do aplicativo para padarias proposto acima, com o objetivo geral de facilitar a comunicação e a comercialização entre o comerciante e o cliente sem interações sociais físicas.

2. Materiais e Métodos

Para realização desse projeto um brainstorm (chuva de ideias), que serviu para o grupo discutir e debater as ideias principais e as problemáticas chegando em um consenso do tema, para ter um embasamento do que será desenvolvido foram utilizados pesquisas e referências bibliográficas para o respaldo da problemática levantada.

Para desenvolver o fluxograma, o Draw.io foi utilizado, é uma plataforma online que serve para construir diagramas, para a codificação do aplicativo em linguagem C foi usada uma IDE (Ambiente de Desenvolvimento Integrado), chamada Visual Studio Code. Na realização dos wireframes e protótipo do aplicativo, foi utilizado o Figma, um editor gráfico no

navegador web. *Vide em anexo 1 - Protótipo Web Figma - Autoria própria.*

Houve uma implementação de banco de dados utilizando o SSMS, porém não houve ligação entre a linguagem e o banco de dados. *Vide em anexo 3 - Banco de Dados SSMS - Autoria própria.*

3. Desenvolvimento

Para tal problemática, o aplicativo em linguagem C feito, tem o objetivo de estabelecer a comunicação do comércio com seus produtos ao cliente. O programa irá funcionar de duas formas. Um sistema ao comerciante, que constitui em efetuar os cadastros dos produtos para quando o cliente entrar no aplicativo possa ter acesso aos produtos. Outro para o cliente, onde visualiza os produtos cadastrados e efetua suas compras.

Vide em anexo 2 - Fluxograma Aplicativo para Padarias - Autoria própria.

No menu inicial o usuário terá a opção para escolher de qual forma irá se cadastrar, como comercio ou cliente. Para o comércio será da seguinte maneira: no cadastro será necessário o nome do comércio, e-mail, telefone, endereço, CNPJ(opcional), usuário e senha para que o comércio possa se cadastrar, em seguida ele irá se deparar com a tela de login, que por sua vez vai pedir o usuário e a senha que foram cadastrados na tela anterior. Após ter passado por essas duas etapas o comercio conseguirá entrar na tela "Menu Comércio" no aplicativo. Para o cliente, no cadastro serão necessários: nome do cliente, e-mail, telefone, endereço, CPF, usuário e senha para se cadastrar que posteriormente também irá para o login e depois terá acesso ao "Menu Cliente", conseguindo assim visualizar os produtos cadastrados.

Após o comerciante ter acesso ao "Menu Comércio" será possível cadastrar seu produto com nome, código e preço para que possam ser exibidos para os clientes em "CADASTRAR PRODUTO". Poderá exibir seus produtos em estoque na opção "EXIBIR PRODUTOS". Conseguirá fazer a alteração dos produtos que contém em estoque em "ALTERAR PRODUTOS". E verificar se os dados da empresa estão corretos dentro do mesmo Aplicativo na opção "DADOS DO COMÉRCIO".

```
C:\Users\biell\Desktop\VSC\P.\ProjetoPI.exe
----- MENU COMERCIO -----
1)CADASTRAR PRODUTO
2)EXIBIR PRODUTOS
3)ALTERAR PRODUTOS
4)DADOS DO COMERCIO
5)INFORMACOES DO SISTEMA
6)SAIR
QUAL OPCAO DESEJA ACESSAR?
```

Figura 1 – Menu Comércio (Terminal executável em Linguagem C) – Autoria própria.

Já para o usuário cliente, após ter acesso ao "Menu Cliente" será disponibilizado a opção de poder comprar os produtos disponíveis em "COMPRAR", procurar algo em específico que o cliente queira, para que possa ser feito o delivery dentro da área de entrega que o comércio disponibiliza "PROCURAR

PRODUTOS", e ver se os seus dados que foram cadastrados estão corretos na opção "INFORMAÇÕES DO USUARIO".

```
C:\Users\biell\Desktop\VSC\P.\ProjetoPI.exe
----- MENU CLIENTE -----
1)COMPRAR
2)PROCURAR PRODUTOS
3)INFORMACOES DO USUARIO
4)INFORMACOES DO SISTEMA
5)SAIR
QUAL OPCAO DESEJA ACESSAR?
```

Figura 2 – Menu Cliente (Terminal executável em Linguagem C) – Autoria própria.

Na opção "Comprar" do "Menu Cliente", ao selecionar todos os produtos será totalizado o valor total de sua compra e após isso será possível escolher se o pagamento será feito na entrega e qual a forma de pagamento que o cliente deseja, a seguir, o pedido será realizado.

Para o aplicativo gerar a validação no login e conceder o acesso ao usuário no menu do aplicativo para o usuário, é necessário fazer manipulações de cadeias e comparar as informações usadas no login com as cadastradas pelo próprio usuário. Conforme Manzano [5], manipular sequências de caracteres, significa operar conjuntos de caracteres na forma de vetor e as funções que auxiliam as manipulações estão presentes no arquivo de cabeçalho chamado string.h. Com isso, variáveis globais com o vetor "SIZE" foram definidas com o tamanho igual a 1, para armazenarmos apenas um dado de cada:

```
34 //cliente
35 char nomeCliente[SIZE][50];
36 char emailCliente[SIZE][50];
37 char telefoneCliente[SIZE][15];
38 char usuCliente[SIZE][50];
39 char senhaCliente[SIZE][50];
40 char endCliente[SIZE][60];
41 char CPF[SIZE][50];
42 //comercio
43 char nomeComercio[SIZE][50];
44 char emailComercio[SIZE][50];
45 char telefoneComercio[SIZE][15];
46 char usuComercio[SIZE][50];
47 char senhaComercio[SIZE][50];
48 char endComercio[SIZE][60];
49 char CNPJ[SIZE][50];
```

Figura 3 - Variáveis Globais dos dados do Usuário (VScode, Linguagem C) - Autoria própria

A seguir nas funções de "loginCliente" e "loginComercio" iremos ler duas sequências de caracteres (string) usando o método strcmp(). Manzano [5] declara que o método strcmp() tem a finalidade de fazer a comparação das cadeias fornecidas, no caso deste programa, tendo como sintaxe igual a zero (=0), para caso o primeiro string

seja igual ao segundo string, encaminhar o acesso ao Menu do usuário:

```
198 if (strcmp(usu1Cliente, usuCliente) == 0
199 && strcmp(senha1Cliente, senhaCliente) == 0) {
200 system("cls");
201 printf("\n\n LOGADO\n\n SEJA BEM VINDO, %s\n\n", usu1Cliente);
202 printf("\n\n ");
203 system("pause");
204 menuCliente(produto); }
```

Figura 4 - Parte da Função "loginCliente" (VScode, Linguagem C) - Autoria própria.

Na criação do código em Linguagem C, algumas ferramentas na linguagem foram essenciais, como por exemplo a declaração de tipo de dados chamada "Struct". Conforme Mizrahi [6], podemos definir o tipo struct antes da função principal, para permitir o acesso global a todas as funções definidas no programa. Mizrahi [6] declara que por meio da palavra-chave struct definimos um novo tipo de dado. Após ter sido informado, o novo tipo existe e pode ser utilizado para criar variáveis de modo similar a qualquer tipo simples. Com isso é possível armazenar um conjunto de variáveis sob um único nome, neste projeto serão as informações dos produtos do comércio (Código, nome e preço):

```
10 typedef struct{
11     int codigo;
12     char nome[100];
13     float preco;
14 }Produto;
15
16 Produto produto[TAM];
```

Figura 5 - Struct Produto (VScode, Linguagem C) - Autoria própria.

Para a realização dos menus, foram criadas diversas funções, para haver a liberdade de fazer um caminho para cada escolha do usuário, e indicarmos um parâmetro chamado "Produto" em cada função que compõe dados de produtos do comércio. De acordo com Ascencio e Campos [7], o usuário pode criar quantas funções quiser, dependendo do problema que estiver sendo resolvido pelo programa. As funções às vezes precisam receber valores externos, chamados parâmetros, e também podem devolver algum valor produzido para o ambiente externo, denominado retorno. Dessa forma há varias funções ligadas umas as outras no código sem estar na função principal:

```
18 //FUNÇÕES//
19 void cadastroCliente();
20 void cadastroComercio();
21 void loginCliente();
22 void loginComercio();
23 void menuCadastro();
24 void menuComercio(Produto *produto);
25 void menuCliente(Produto *produto);
26 void alterarProdutos(Produto *produto);
27 void exibirProdutos(Produto *produto);
28 void procurarProdutos(Produto *produto);
29 void cadastrarProdutos(Produto *produto);
30 void Comprar(Produto *produto);
```

Figura 6 - Todas as funções (VScode, Linguagem C) - Autoria própria.

Utilizando a estrutura é possível armazenar os produtos cadastrados pelo comércio na função de nome "cadastrarProdutos" e exibi-los para os dois (Comércio e Cliente) nas funções "exibirProdutos" e "procurarProdutos" respectivamente. Para o cliente conforme o código do produto que procurou, ele poderá compra-lo na função de nome "Comprar". De acordo com Mizrahi [6], membros individuais de cada estrutura são acessados aplicando-se o operador ponto seguido do nome da variável, que, nesse caso é o elemento da matriz.

```
490 for(int i=0;i<TAM; i++){
491     if(codCompra == produto[i].codigo) {
```

Figura 7 - Membro individual "produto[i].codigo" da Struct Produto (VScode, Linguagem C) - Autoria própria.

Assim, o usuário cliente irá conseguir ter acesso ao produto escolhido na hora da compra, que foi armazenado na estrutura e realizar o pedido.

Para a realização desse aplicativo, tivemos algumas matérias que foram essenciais para o desenvolvimento do mesmo sendo elas: Paradigmas e Linguagens de Programação, Algoritmos e Lógica de Programação, Fundamentos Matemáticos da Computação e sistemas computacionais.

A matéria de Paradigmas e Linguagens de Programação teve um papel fundamental, pois com essa matéria aprendemos muitos conceitos e fundamentos das linguagens, aprendendo assim as suas semânticas e a sintaxes, também aprendemos qual linguagem podemos escolher para uma melhor performance desse aplicativo, sendo assim, decidindo a linguagem de programação c para a realização de deste aplicativo.

A matéria de Algoritmos e lógica de Programação, nos ajudou a pôr em pratica os conhecimentos que tivemos na matéria de Paradigmas e Linguagens de Programação, e nos ajudou a exercitar a nossa lógica para um melhor desempenho nas linguagens abordadas, sendo assim, aprendemos a fazer nossos primeiros códigos, fluxogramas e pseudocódigos, e com isso nós desenvolvemos para a realização desse projeto.

A matéria de Fundamentos Matemáticos nos ajudou a entender alguns conceitos que foram essenciais para a realização desse projeto, exercitando a nossa lógica e aprimorando os nossos conhecimentos sobre a matemática, desse jeito conseguimos ter uma boa logica para conseguirmos

aplicar alguns conceitos matemáticos sem nenhum problema e isso facilitou a programação do nosso projeto.

A matéria de Sistemas Computacionais, nos mostrou os requisitos e componentes fundamentais para que o nosso aplicativo possa funcionar tanto na parte do comércio, como na parte do cliente, pois com isso vimos que não precisamos de configurações muito avançadas como por exemplo um processador e um HD de última geração para que o nosso aplicativo possa rodar bem, e sim configurações precisas e adequadas.

A matéria de Comunicação e Linguagem, foi essencial para aprendermos a nos comunicar em diferentes formas e em situações diversificadas, e com isso tivemos uma grande base de como nos expressar de um material de fácil entendimento para melhorar a explicação e a realização do nosso projeto.

4. Considerações Finais

Em suma, concluímos que, um aplicativo ajudará muito as pequenas e médias padarias na situação na qual vivemos hoje, pois com esse aplicativo amenizaremos os problemas desse tão importante comércio, sendo assim, facilitando as suas vendas para os seus clientes, mesmo sem a interação social, e com isso esse comércio não terá que fechar as suas portas por conta de problemas financeiros e por falta de recursos para se manter. Conseguimos chegar em resultados importantes em nosso projeto, são eles: A tela de cadastro e login, tanto para o cliente quanto para o comércio. A forma do comércio cadastrar os produtos que o mesmo tem em seu estoque. A parte do cliente comprar os produtos cadastrados pelo comércio. E a efetivação da compra em si.

Para o futuro, estamos pensando na forma de entrega dos produtos comprados pelos clientes, pois com isso o cliente receberá a sua compra sem sair de casa, ajudando assim tanto o lado do comércio quanto o lado do cliente.

5. Referências

[1] Instituto Brasileiro de Geografia e Estatística. Pandemia foi responsável pelo fechamento de 4 em cada 10 empresas com atividades encerradas [Internet]. Brasil: Governo Federal; 16 Jul 2020. [Última Atualização 17/07/2020; Acesso em 27 abril 2021]. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/28295-pandemia-foi-responsavel-pelo-fechamento-de-4-em-cada-10-empresas-com-atividades-encerradas>

[2] Silva A.N. Pandemia foi responsável pelo fechamento de 4 em cada 10 empresas com atividades encerradas [Internet]. Brasil: Correio Braziliense; 21 Jan 2021 [Última Atualização 22 Jan 2021; Acesso em 27 Abr 2021]. Disponível em: <https://www.correiobraziliense.com.br/cidades-df/2021/01/4901630-pandemia-leva-a-reinvencao-das-padarias-lucro-cai-ate-50.html>

[3] Guimarães E. Padarias terão que se reinventar durante a pandemia de coronavírus [Internet]. Brasil: Estado de Minas; 27 Mai 2020 [Última Atualização 27 Mai 2020; Acesso em 27 Abr 2021]. Disponível em: https://www.em.com.br/app/noticia/economia/2020/05/27/internas_economia,1151216/padarias-terao-que-se-reinventar-durante-a-pandemia-de-coronavirus.shtml

[4] Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. Como o ramo de panificação pode melhorar o desempenho na crise?: O quanto a crise pelo novo coronavírus afetou as padarias? [Internet]. Brasil: Governo Federal; 30 Abr 2020 [Última atualização 9 Set 2020; Acesso em 29 Abr 2021]. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/como-o-ramo-de-panificacao-pode-melhorar-o-desempenho-na-crise,d164d29365ac1710VgnVCM1000004c00210aRCRD>

[5] Manzano J.A.N. Linguagem C: acompanhada por uma xícara de café. 1ª ed. São Paulo: Érica; 2015. 480 p. ISBN: 978-85-365-1936-4.

[6] Mizrahi V.V. Treinamento em Linguagem C. 2ª ed. São Paulo: Editora Pearson; 2008. 434 p. ISBN: 9788576051916.

[7] Ascencio A.F. Campos EA. Fundamentos da Programação de Computadores: algoritmos, Pascal, C/C++ e Java. 2ª ed. São Paulo: Editora Pearson; 2007. 448 p. ISBN: 9788576051480.

1. Figma

Editor gráfico de vetor e prototipagem de projetos de design baseado principalmente no navegador web.

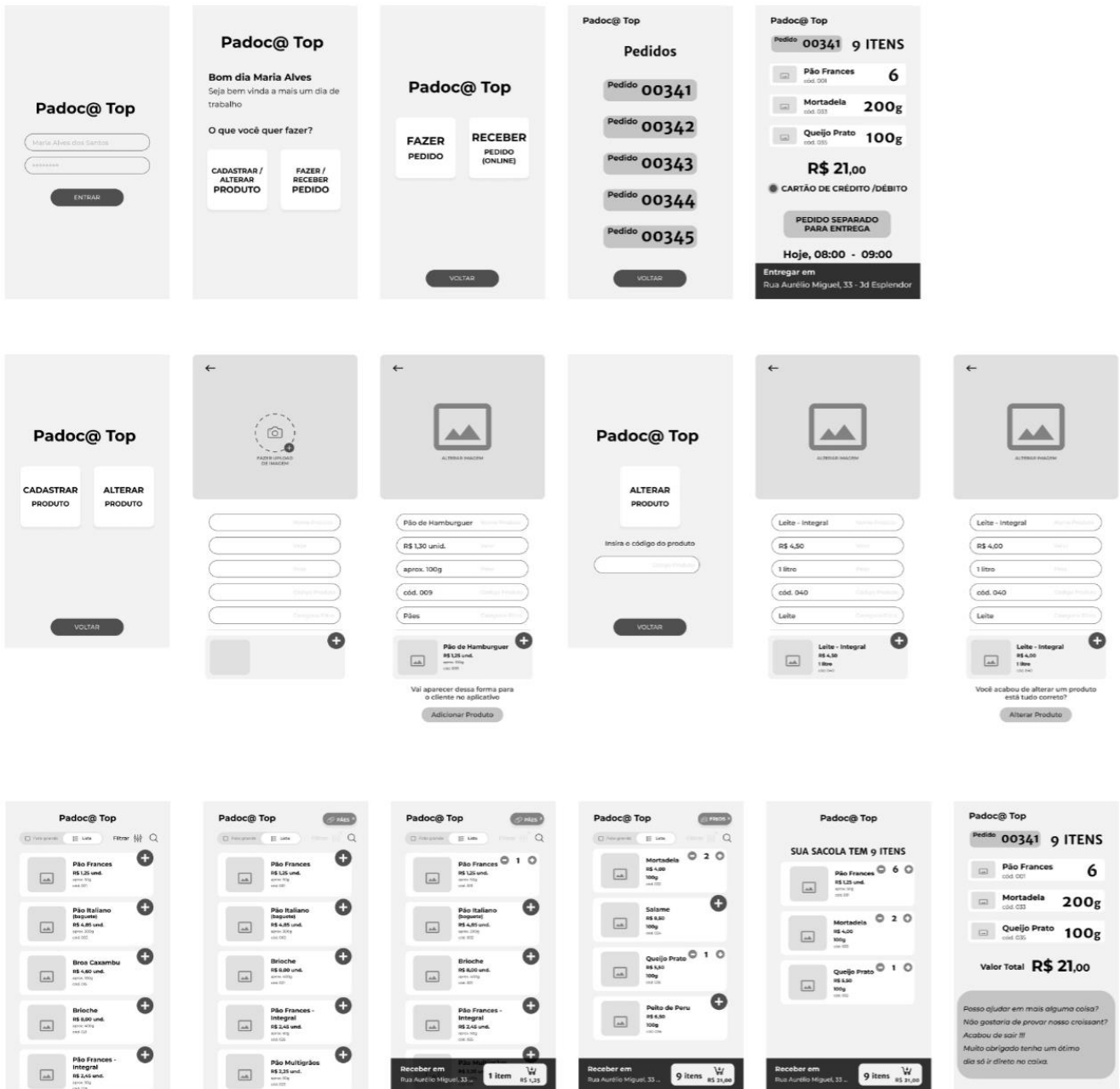
TELAS ÁREA CLIENTE – CADASTRO SIMPLES E COMPRA



Protótipo Navegável

<https://www.figma.com/proto/mWN4i7eqHWmI02T7pFGEC4/PADOC%40-TOP-cliente?node-id=422%3A835&scaling=scale-down&page-id=0%3A1>

TELAS ÁREA FUNCIONÁRIO – CADASTRAR E ALTERAR PRODUTOS PADARIA, RECEBER PEDIDOS ONLINE E REALIZAR COMPRAR PARA O CLIENTE PRESENCIAL

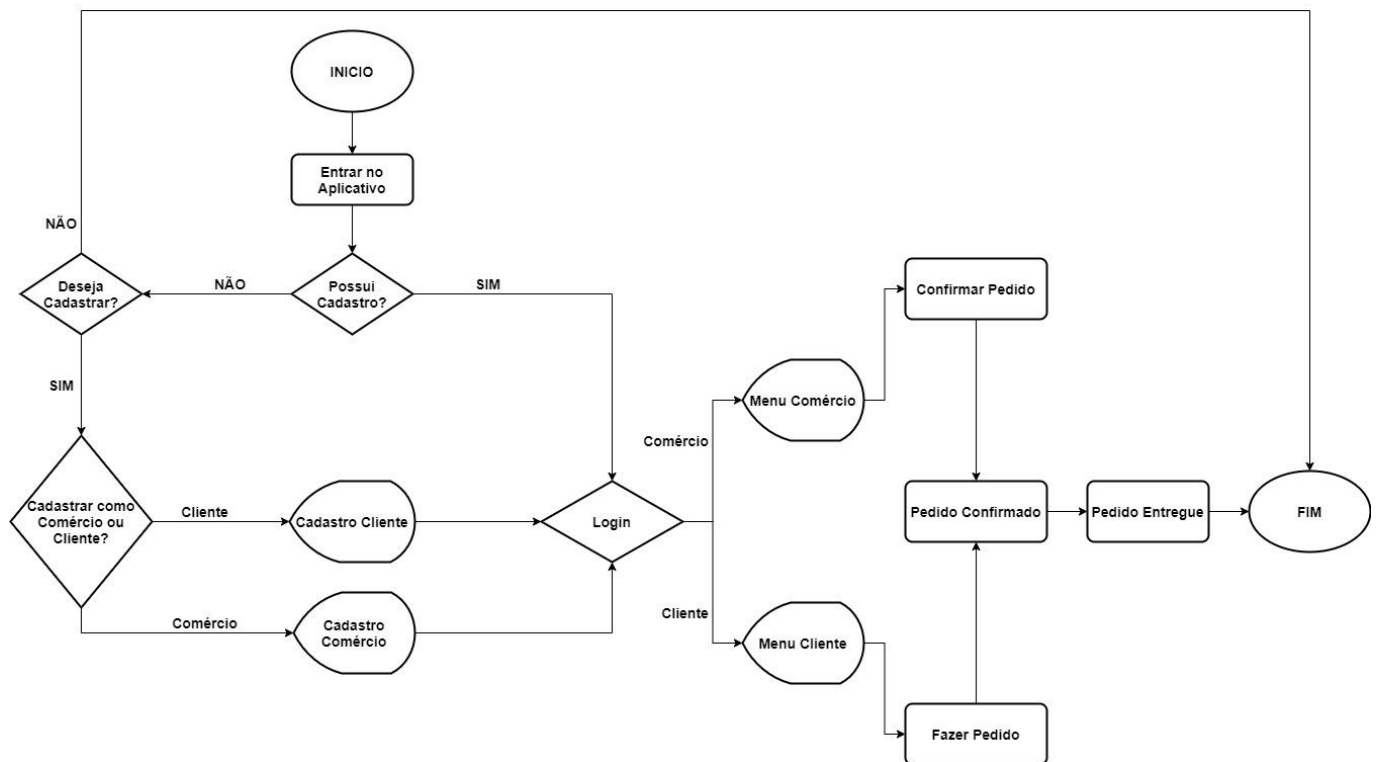


Protótipo Navegável

<https://www.figma.com/proto/joR9dtuyvvFFLP75K7UfW7/PADOC%40-TOP-funcionario?scaling=scale-down&page-id=0%3A1&node-id=1426%3A0>

2. Fluxograma

Feito no Draw.io. Uma visão total do aplicativo em um tipo de diagrama.



3. Banco de Dados

Feito no SQL Server Management Studio. Houve uma implementação de um Banco de dados sobre o aplicativo, porém não houve uma ligação entre a linguagem e o banco de dados.

Informações do cliente:

```

CREATE TABLE cadastroCliente (

    IDcliente int,
    Nome_Cliente varchar(50) not null,
    Email varchar(50) not null,
    Usuario varchar(50) not null,
    Senha varchar(50) not null,
    Endereco varchar(50) not null,
    CPF bigint not null ,

    constraint cadastroCliente_pk primary key (IDcliente),
    constraint cadastroCliente_un unique (CPF)

);
  
```

Informações do comércio:

```
CREATE TABLE cadastroComercio (

    IDcomercio int,
    Nome_Comercio varchar(50) not null,
    Email varchar(50) not null,
    Usuario varchar(50) not null,
    Senha varchar(50) not null,
    Endereco varchar(50) not null,
    CNPJ bigint not null ,

    constraint cadastroComercio_pk primary key (IDcomercio),
    constraint cadastroComercio_un unique (CNPJ)

);
```

Produtos do comércio:

```
CREATE TABLE produtosComercio (

    IDproduto int not null,
    Codigo int not null,
    Nome_Produto varchar(50) not null,
    Preco money not null,

    constraint ProdutosComercio_pk primary key (IDproduto),
    constraint produtosComercio_un unique (Codigo)

);
```

Pedidos:

```
CREATE TABLE Pedidos (

    IDpedido int,
    IDcliente int,
    IDcomercio int,
    IDproduto int,
    Forma_pagamento varchar(50) not null,
    Valor money not null,
    Troco money null,

    constraint Pedidos_pk primary key (IDpedido),
    constraint Pedidos_cliente_fk foreign key (IDcliente) REFERENCES cadastroCliente (IDcliente),
    constraint Pedidos_comercio_fk foreign key (IDcomercio) REFERENCES cadastroComercio (IDcomercio),
    constraint Pedidos_produto_fk foreign key (IDproduto) REFERENCES produtosComercio (IDproduto)

);
```