

ENGENHARIA DE SOFTWARE

3º Série – Noturno

André Luiz Souza Volpato – RA: 21011458-2

Gustavo da Silva Bezerra – RA: 21173515-2

José Ricardo Miessi Gomes – RA: 21105683-2

Leonardo Losso Szycha – RA: 21088003-2

Ai Qui Diliça
MARINGÁ-PR

ATIVIDADE DE ESTUDO PROGRAMADA DO 1º BIMESTRE

MARINGÁ
2022

Desenvolvimento de aplicativo de delivery que solucione os problemas dos aplicativos atuais

Introdução

É de conhecimento geral que os aplicativos delivery estão crescendo de formas astronômicas, só no ano de 2020, o Brasil foi responsável por 48,77% do uso de delivery em toda a América Latina, com um giro trilionário no ano de 2021. Só no começo da pandemia no ano de 2020, dados mostram que as receitas de compras pelo delivery do segmento alimentício aumentaram em até 77% entre os dias 1º e 18 de março de 2020, número esse 46% maior que março de 2019.

Estudos apontam que até 47% dos estabelecimentos aderiram a nova forma de venda, só sendo ultrapassado pelo WhatsApp. Entretanto, nem todos os aplicativos são tão bons quanto deveriam ser, com pontos a melhorar e defeitos a serem corrigidos e essa é a razão pela criação desse aplicativo.

Objetivo

Melhorar o relacionamento do cliente e dos restaurantes para com o aplicativo, através de taxas menos abusivas. Aumentar a comodidade ao cliente tal como, avisos de quando seu pedido irá chegar em seu endereço, melhores taxas para os restaurantes em comparação aos concorrentes.

Otimizar o aplicativo para dispositivos móveis (iOs e Android), e fazer uso de uma API de GPS para fazer o monitoramento do trânsito de seu pedido. Aumentar a rentabilidade do entregador e melhoria da economia local.

Justificativa

Com base nos demais aplicativos já existentes de delivery e após más experiências com os demais aplicativos, consideramos que é de alta importância a existência de um aplicativo que não cobre taxas abusivas dos restaurantes, o que

consequentemente acaba prejudicando toda a cadeia de pagamentos, dos entregadores aos clientes.

A falta de recursos para o cliente saber se seu pedido está chegando ou não, bem como a falta de informações acerca de seu pedido, se foi confirmado ou não, se está em produção ou não, se saiu para a entrega ou não.

Desenvolvimento

Vantagens e desvantagens de vender no ifood, Aiqfome e Rappi

O presente estudo tem como objetivo elencar vantagens e desvantagens do uso dos aplicativos de Delivery mais usados no Brasil (iFood, Aiqfome e Rappi)

iFood - Vantagens:

Facilidade de implantação.

Cadastro e aprovação do estabelecimento em alguns minutos, precisando informar.

Segurança no pagamento, os pagamentos são feitos diretamente no aplicativo, o que faz com que o estabelecimento não deixe de receber nenhum pagamento.

Aumento nas vendas, seu estabelecimento estará visível para um grande número de clientes, fazer delivery aumentará os seus clientes, consequentemente, aumentará suas vendas.

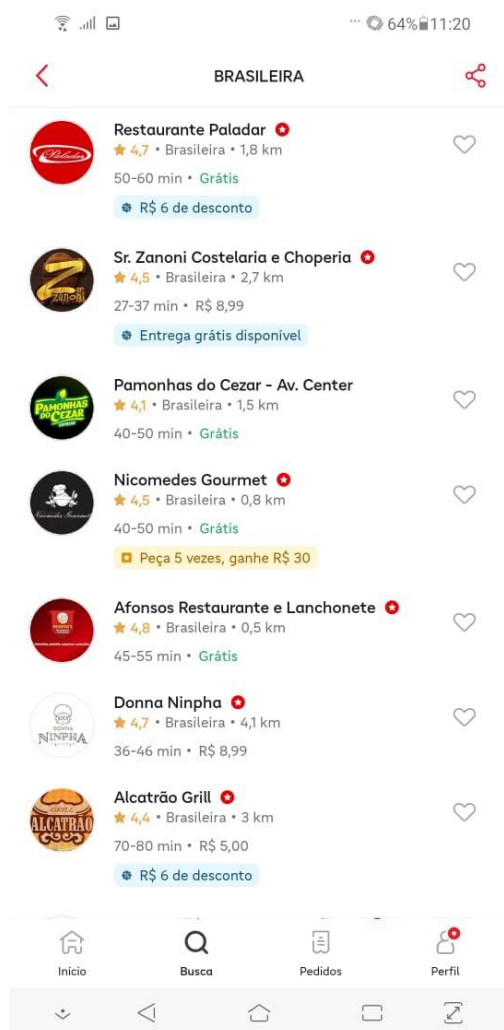
Serviço de entrega próprio, o iFood conta com um serviço de entrega próprio caso seu estabelecimento não possua equipe para tal.

Desvantagens:

Pagamento de comissão, as famosas taxas que o iFood cobra sobre cada pedido, podendo chegar em até 20% do valor do pedido.

Dependência total da internet, caso seu estabelecimento fique sem energia ou sem internet as vendas cairão, então se seu estabelecimento é somente delivery é bom tomar precauções para evitar esse tipo de problema.

Vasta concorrência, justamente pela facilidade de se cadastrar no iFood e pelo amplo alcance dos clientes, a concorrência no iFood é alta.



Aiqfome - Vantagens:

Excelente oportunidade para atrair novos clientes, por ser um app menor que o iFood a possibilidade de atrair novos clientes é maior.

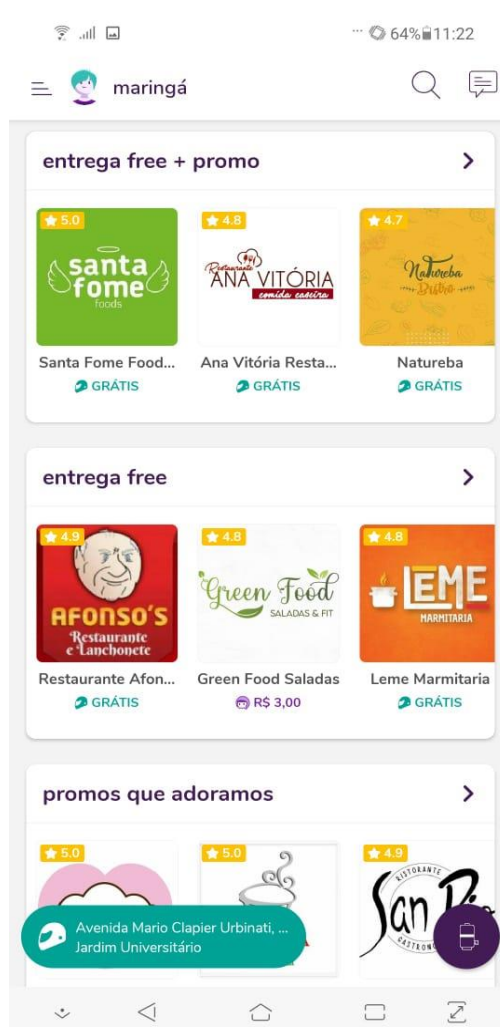
Taxa de comissão menor que a do iFood, o aplicativo cobra 12% de taxa de comissão pelas vendas e uma taxa adicional de 2,99% para os pedidos pagos pelo app, fora que não possui mensalidade e nem carência.

Por ser um negócio menor que o iFood o atendimento é facilitado, sendo pessoas e não bots de chat fazendo o seu atendimento.

Desvantagens:

Por se tratar de um negócio voltado às cidades de pequeno e médio porte, as capitais estão fora do seu escopo de atuação, ficando limitado a somente esses dois tipos de cidades.

Não possui serviço de entrega como no iFood, então o estabelecimento irá precisar ter seu próprio serviço de entrega.



Rappi - Vantagens:

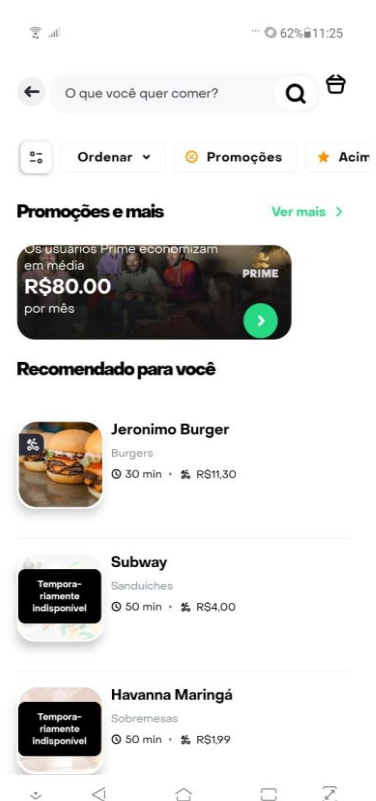
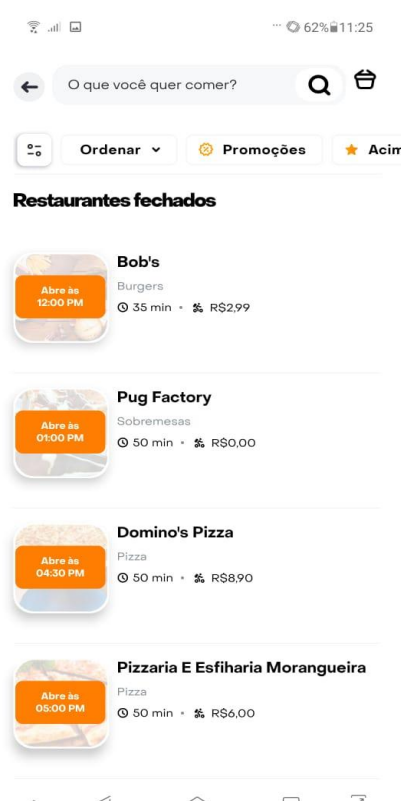
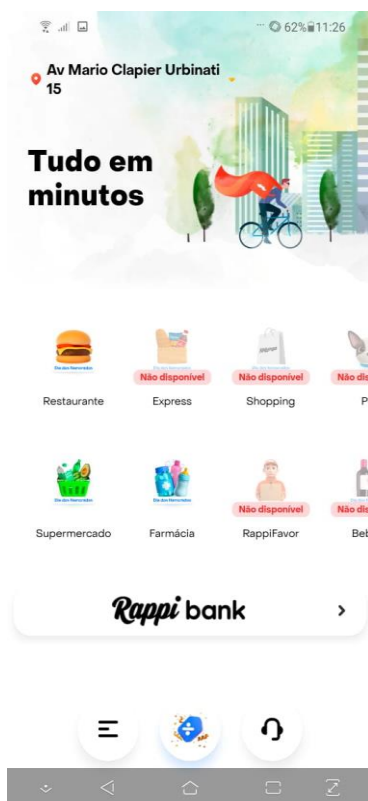
Rappi entra na mesma situação do Aiqfome, sendo este um aplicativo não tão grande como o iFood, mas ainda assim com a possibilidade de captar novos clientes.

Não há cobrança de mensalidades.

Desvantagens:

Taxas de adesão e comissão, o Rappi tem dois planos sendo eles com taxas diferentes. “Entrega do Seu Jeito” - Este plano tem taxa de adesão de R\$40,00 e cobra 12% de comissão a cada venda, além de uma taxa de 3,5% para os pagamentos feitos diretamente pelo aplicativo, não cobrindo uma equipe de entrega.

E o plano “Full Service”, que conta com uma equipe de entrega do aplicativo, tem taxa de adesão de R\$150,00 e cobra uma taxa de venda de 27%.



Entrevista - Lupes Lanches

Fizemos também uma entrevista com a Lupes Lanches para saber a viabilidade do nosso projeto.

1- Qual a sua maior dificuldade em utilizar aplicativos voltados a delivery?

Taxas cobradas com valor muito alto no valor do produto final, Ifood por exemplo cobra uma taxa de 27%, essa taxa é com a utilização dos motoboys deles, caso eu tenha meus próprios motoboys essa taxa cai para 12%, mas neste caso entra os custos de ter um entregador particular, que acaba gerando um custo maior dependendo do movimento.

2- Em caso de a taxa de serviço ser 0 ou próximo a isso, você aumentaria o valor de seu serviço?

Não, meu cardápio no ifood é mais caro que o presencial devido a taxa de 27%, como a maioria dos restaurantes lá cadastrados, acredito que se essa taxa fosse menor, o valor dos produtos também seria menor.

3- De que maneira você se comunica com os seus clientes?

Maioria pelas redes sociais, pedidos para retirar geralmente são feitos pelo whatsapp, ou via telefone mesmo, e claro o atendimento presencial.

4- Qual a maior dificuldade de entrar em contato com os clientes?

No meu caso é meu tempo, digo isso pois falta tempo para criar promoções e publicações nas redes sociais, como o próprio instagram.

5- Você pagaria um valor específico para ter seu comércio destacado dentro de um aplicativo de delivery? Se sim, qual o valor você pagaria?

Não pagaria.

6- Você acredita que utilizando um aplicativo delivery, a concorrência em cima é amenizada?

Não, no meu caso que é alimentação a concorrência é altíssima, acredito que cada um tem seu espaço no comércio, a diferença é se os clientes irão voltar ou não, aí depende somente do restaurante em si.

7- Você sente falta de alguma funcionalidade dentro de um aplicativo de delivery?

Eu utilizo apenas os ifood e aiqfome, então vou responder esta pergunta com base nessas plataformas. Sinto falta de criar promoções em produtos específicos que sejam destacadas no aplicativo, e também um contato mais direto com o cliente, no ifood existe o chat deles, mais muito mal aplicado, muito difícil conversar com o cliente.

The image displays two screenshots of the iFood mobile application interface for the restaurant 'LUPE'S LANCHES'. The left screenshot shows the restaurant's profile page, including a header image of food, the restaurant name with a verified badge, distance (3.3 km), price range (\$\$\$\$), delivery status (available), estimated delivery time (33-43 min), price (R\$ 6,99), a 4.7-star rating from 33 reviews, a minimum order of R\$ 20,00, and a promotional banner for 'Economize até 70% com o Clube iFood'. Below this, three menu categories are listed: 'Combos', 'Lanches de Carne', and 'Vegar'. The 'Combos' section is expanded, showing three items: 'Combo Lupe's X-Burguer' (R\$ 31,00), 'Combo Lupe's Simples' (R\$ 26,00), and 'Combo Lupe's X-Salada' (R\$ 26,00). Each item includes a description and a coupon code 'R\$ 1,99 no Clube'. The right screenshot shows the 'Lanchão do Lupe's' menu category, which is selected. It lists five items: 'Lupe's Simples' (R\$ 15,00), 'Lupe's Duplo' (R\$ 17,00), 'Lupe's Purê' (R\$ 19,00), 'Lupe's Pizza' (R\$ 21,00), and 'Lupe's Calabresa' (R\$ 21,00). Each item includes a description and the same coupon code 'R\$ 1,99 no Clube'. The top of the app shows standard mobile status bar icons and the time 20:29.

Lupe's Lanches

Lanches • 3,3 km • \$\$\$\$

Entrega Hoje, 33-43 min • R\$ 6,99

4,7 33 avaliações

Pedido mínimo R\$ 20,00

Economize até 70% com o Clube iFood

Combos

Combo Lupe's X-Burguer
Pão fofinho, 100g bland especial de hamburger Lupe's, queijo mussarela, maionese temperada da casa, catchup.
Serve 1 pessoa

R\$ 31,00

Combo Lupe's Simples
Pão fofinho, salsicha, batata palha crocante, maionese temperada à moda Lupe's, catchup, alface fresquinha e tomate em rodela....
Serve 1 pessoa

R\$ 26,00

Combo Lupe's X-Salada
Pão fofinho, 100g bland especial de hamburques Lupe's.

Cupons por R\$ 1,99 no Clube

LUPE'S LANCHES
Entregar em 33-43 min

Combos **Lanchão do Lupe's** Lanches de Carne: Vegar

Lanchão do Lupe's

Lupe's Simples
Uma Salsicha, Batata Palha, Maionese Temperada, Catchup, Alface E Tomate.

R\$ 15,00

Lupe's Duplo
Duas Salsicha, Batata Palha, Maionese Temperada, Catchup, Alface E Tomate.

R\$ 17,00

Lupe's Purê
Uma Salsicha, 100g de Purê de Batata, Batata Palha, Maionese Temperada, Catchup, Alface E Tomate.

R\$ 19,00

Lupe's Pizza
Uma Salsicha, Queijo Mussarela, Presunto, Oregano, Batata Palha, Maionese Temperada, Catchup, Alface E Tomate.

R\$ 21,00

Lupe's Calabresa
Uma Salsicha, 150g de Calabresa, Batata Palha, Maionese Temperada, Catchup, Alface e Tomate.

Cupons por R\$ 1,99 no Clube

Levantamento de requisitos funcionais

Tela do Cliente

RF-C001 - Cadastro de clientes: Eu como usuário devo poder realizar o meu cadastro na plataforma de delivery, para poder acessar o aplicativo.

RF-C002 - Listar estabelecimentos: Eu como usuário gostaria de listasse todos os estabelecimentos, para eu ter mais opções de escolha.

RF-C003 - Busca de pratos e restaurantes: Eu como usuário gostaria de procurar por restaurantes e por pratos, para agilizar a escolha.

RF-C004 - Disponibilidade de Restaurantes: Eu como usuário gostaria se o restaurante está disponível para efetuar o pedido.

RF-C005 - Filtro de buscas: Eu como usuário gostaria de utilizar filtros para refinar a pesquisa.

RF-C006 - Histórico de pedidos: Eu como usuário gostaria de ver o histórico de todos os pedidos realizados, para ver o que eu já pedi.

RF-C007 - Informações do Restaurante: Como usuário gostaria de ver as informações que aquele restaurante possui, para ver se o restaurante tem as qualidades que procuro.

RF-C008 - Tipo de Entrega: Eu como usuário quero escolher entre as opções relacionadas ao tipo de entrega:

RF-C008.1 - Entrega agendada para um determinado horário;

RF-C008.2 - Retirada do pedido;

RF-C008.3 - Entrega convencional;

RF-C009 - Localização das entregas: Eu como usuário gostaria de poder definir o local da entrega e deixar salvo, porque posso não estar na minha casa na hora do pedido.

RF-C010 - Sugestões de restaurantes: Eu como usuário gostaria de receber sugestões de restaurantes baseado na sua avaliação por padrão e pela localização mais próxima do usuário, para agilizar minha escolha.

RF-C011 - Avaliação de restaurantes: Eu como usuário quero poder avaliar o meu prato e o restaurante onde pedi, para dar a minha opinião sobre o pedido.

RF-C012 - Avaliação dos pedidos: Eu como usuário gostaria de poder avaliar os pedidos, podendo ainda adicionar comentários, porque quero demonstrar minha satisfação com o pedido.

RF-C013 - Edição de dados: Eu como usuário gostaria de editar meus dados pessoais, pois se mudar de e-mail ou telefone gostaria de atualiza-los.

RF-C014 - Notificações: Eu como usuário gostaria de ser avisado por notificações sobre o status do pedido, porque quero ficar ciente sobre o pedido.

RF-C015 - Pagamentos: Eu como usuário gostaria de escolher entre as formas de pagamento disponíveis, porque quero ter essa praticidade na hora da escolha.

RF-C016 - Forma de entrega: Eu como usuário quero ter opções na escolha da forma de entrega, se é para deixar na porta/portaria ou se encontrar com o entregador, porque quero ter essas opções de entrega.

RF-C017 - Chat: Como usuário eu preciso que tenha a possibilidade de abrir um chat direto com os restaurantes, para eventuais problemas que possam ocorrer.

RF-C018 – Notificação da Entrega: Eu como usuário gostaria de ser avisado quando o pedido estiver a 5 minutos do local de entrega para agilizar o pagamento.

RF-C019 – Carrinho dos pedidos: Eu como usuário gostaria que tivesse um carrinho de pedidos para ir adicionando os itens, para que eu possa pagar mais de um produto apenas uma vez.

RF-C020 – Personalização de pedidos: Eu como cliente final gostaria que tivesse uma tela de personalização de pedidos, para que eu possa ter a opção de acrescentar ou retirar ingredientes ao meu pedido.

RF-C021 – Ícone de atalho para o carrinho de pedidos: Eu como usuário gostaria que tivesse um ícone de atalho para que pudesse acessar o carrinho de pedidos durante a minha navegação pelo aplicativo e não precise voltar ao início para acessá-lo.

Tela dos estabelecimentos

RF-R001 - Cadastro de produtos: Eu como proprietário do restaurante preciso efetuar o cadastro dos itens do restaurante, com uma breve descrição do que é composto e com o seu preço, porque quero que o cliente saiba o que está comprando e quanto pagará.

RF-R002 - Gerenciamento dos pedidos: Eu como proprietário do restaurante preciso que o sistema possibilite a escolha entre aceitar, recusar e modificar um pedido, de acordo com meus próprios critérios, caso haja eventuais problemas, consultando sempre o cliente.

RF-R003 - Cadastro do restaurante: Eu como proprietário do restaurante preciso que seja possível de realizar o cadastro do meu restaurante, para conseguir vender no aplicativo.

RF-R004 - Tipo de entrega: Eu como proprietário do restaurante preciso escolher qual tipo de entrega ofertarei, se será entrega grátis ou paga, porque caso eu tenha meus próprios entregadores não usarei os da plataforma.

RF-R005 - Valor mínimo dos pedidos: Eu como proprietário do restaurante, gostaria que o aplicativo possibilitasse eu colocar um valor mínimo para os pedidos pois, poderia acarretar em prejuízos para o meu estabelecimento.

RF-R006 - Funcionamento do restaurante: Eu como proprietário do restaurante preciso colocar horário de funcionamento do meu estabelecimento, para demonstrar aos clientes o horário de fechamento e abertura do restaurante.

RF-R007 - Promoções: Eu como proprietário do restaurante, gostaria que o aplicativo possibilite que eu faça promoções da forma que eu desejar, pois posso abrir promoções em dias diferentes.

RF-R008 - Distância de entregas: Eu como proprietário do restaurante preciso que o aplicativo permita que eu defina o raio de entrega que gostaria de fazer, pois entregas muito distantes poderiam demorar muito, esfriar o pedido e prejuízo.

RF-R009 - Opção de endereço: Eu como proprietário do restaurante preciso definir o endereço do meu restaurante pois, se tiver um pedido para retirar o cliente deve saber onde encontrar o restaurante.

Tela do Entregador

RF-E001 - Confirmação de entrega: Eu como entregador preciso que o aplicativo possibilite a opção de aceitar ou recusar a entrega do pedido, porque pode ser inviável para fazer a entrega.

RF-E002 – Endereços: Eu como entregador preciso que o aplicativo mostre o endereço do cliente, caso seja alguma área perigosa terei a opção de aceitar ou não o pedido.

RF-E003 - Rotas: Eu como entregador preciso que o aplicativo forneça a melhor rota possível para mim, para agilizar na entrega.

RF-E004 - Imprevistos: Eu como entregador preciso que o aplicativo tenha um chat direto e número de telefone do restaurante, pois se houver algum imprevisto eu conseguiria avisar ao restaurante.

RF-E005 – Pagamento do Entregador: Eu como entregador preciso que o aplicativo mostre quanto será pago a mim, pois pode ser inviável realizar a entrega a depender do valor pago.

RF-E006 - Propostas de pedidos oferecidos ao entregador: Eu como entregador gostaria de receber propostas de corrida próximo a minha rota, para que eu possa expandir meus lucros.

Requisitos de Sistema

RS-001 – Notificação da Entrega:

- O sistema deverá mapear e comparar o local da entrega com o local do entregador;
- Quando o entregador estiver a 5 minutos do local da entrega deverá avisar através de notificação com um som diferente;
- Deverá acessar a API do Waze para verificar a estimativa de tempo entre a distância do entregador e o local de entrega para que quando estiver uma estimativa de 5 minutos a notificação será mandada;
- A notificação deverá emitir um som;
- A notificação deverá ser flutuante, ou seja, irá aparecer na tela do celular do usuário mesmo que estiver fora do aplicativo;
- A notificação deverá apresentar o tempo de 5 minutos, uma mensagem, o nome do aplicativo e o ícone.

RS-002 – Avaliação do Pedido:

- O aplicativo possibilitará ao usuário a avaliação de cada prato;
- As avaliações apareceram ao lado do respectivo prato na tela de cardápio;
- Para que o usuário consiga realizar uma avaliação ele precisará estar logado no aplicativo;
- Será mostrada 5 estrelas translúcidas e o preenchimento delas se dará pela escolha da respectiva estrela;
- Será exibido também a nota média e a nota total, nota essa que será baseada de 1 a 5 estrelas;

- Será exibido também um campo de mensagem “Digite sua avaliação (opcional)” com limite máximo de 0 a 255 caracteres;
- Terá um botão de “cancelar” onde toda a avaliação já feita não será mandada para o sistema;
- Terá também um botão de “Enviar” que enviará a avaliação do prato;

RS-003 – Propostas de pedidos oferecidos ao entregador:

- O aplicativo irá mapear os restaurantes próximos à rota de entrega;
- Os restaurantes mapeados estarão a no máximo 1 km de distância;
- Após o entregador aceitar uma entrega o sistema irá mostrar somente outras sugestões que estão próximos ao local de entrega ou próximo a rota estabelecida;
- Com o sistema de sugestões disponível ao entregador, espera-se maximizar sua rentabilidade;

RS-004 – Adicionar pedido ao carrinho de compras:

- Para escolher um pedido é preciso deslizar o item para a esquerda;
- Ao arrastar para a esquerda abrirá uma tela de personalização de pedidos – RS-C023;
- Ao efetuar a personalização ou optar por não modificar nada, o pedido será adicionado ao carrinho (RF-C022) e retornará ao catálogo de produtos;
- Ao retornar ao catálogo de produtos será mostrado um ícone de carrinho com o número de pedidos

Requisitos Não Funcionais

1.0 Requisitos Não Funcionais de Produto:

1.0.1 - De Usabilidade:

- 1.0.1.1 – Modo Contraste: O aplicativo deverá ter uma opção para ativar o modo de contraste, o qual deverá atender as necessidades de pessoas com algum tipo de dificuldade/deficiência visual. Exemplo: Daltonismo.
- 1.0.1.2 – Leitor de Tela: O aplicativo deverá ter leitor de tela para deficientes visuais conseguirem usar.

- 1.0.1.3 – Aumentar tamanho das letras: O aplicativo deverá ter a opção de aumentar e de diminuir o tamanho das letras.
- 1.0.1.4 – Interface Gráfica com o usuário: A interface deverá ser simples e intuitiva devido ao fato de um possível usuário que não tenho muita experiência com uso do celular. Sendo assim, deverá ter mensagens de erros explicativas que demonstrem claramente como o usuário deve agir.
- 1.0.1.5 – Tutorial: Os usuários deverão operar o sistema após passarem pelo tutorial.
- 1.0.1.6 – Padronização de Telas: O aplicativo deverá ter uma padronização de telas, seguindo a mesma paleta de cores, tamanho e estilos de fontes.

1.0.2 - De Performance e Eficiência:

- 1.0.2.1 – Tempo de Respostas: O tempo de resposta de uma consulta feita pelo usuário deverá ser de no máximo 5 segundos.
- 1.0.2.2 – Requisições: O sistema deverá atender no mínimo 50 requisições ativas.
- 1.0.2.3 – Erros por dia: O sistema deverá ser capaz de funcionar sem que ocorra 3 erros por dia, para evitar que o sistema entre em colapso com erros sucessivos.

1.0.3 – De Segurança e Proteção:

- 1.0.3.1 – Criptografia: Deverá proteger os dados durante a comunicação entre sistemas utilizando criptografia.
- 1.0.3.2 – Sessões Ativas: O sistema deverá ter apenas uma sessão ativa.
- 1.0.3.3 – Acesso: Os dados de usuário deverão ser acessados apenas pelo usuário.

1.0.4 – De Disponibilidade:

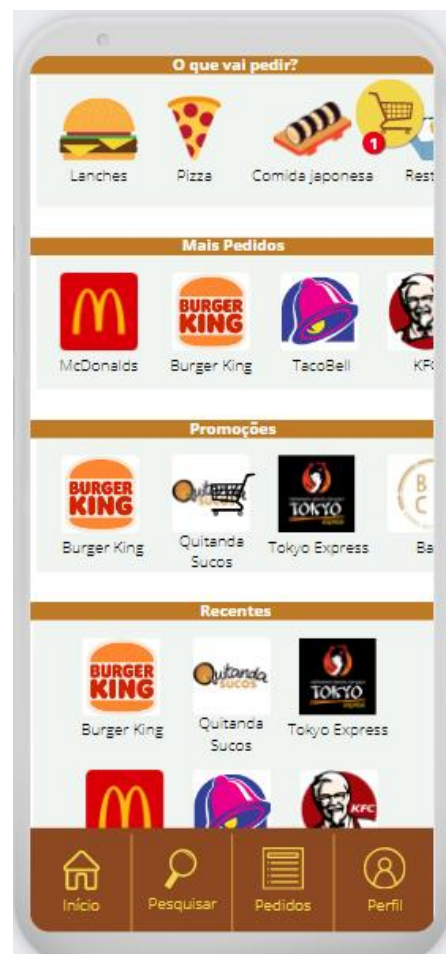
- 1.0.4.1 – O sistema deverá ser acessível 24 horas por dia e 7 dias por semana.
- 1.0.4.2 – O sistema deverá ter espelhamento de um banco de dados para caso um cair terá o outro para manter o sistema disponível aos usuários.

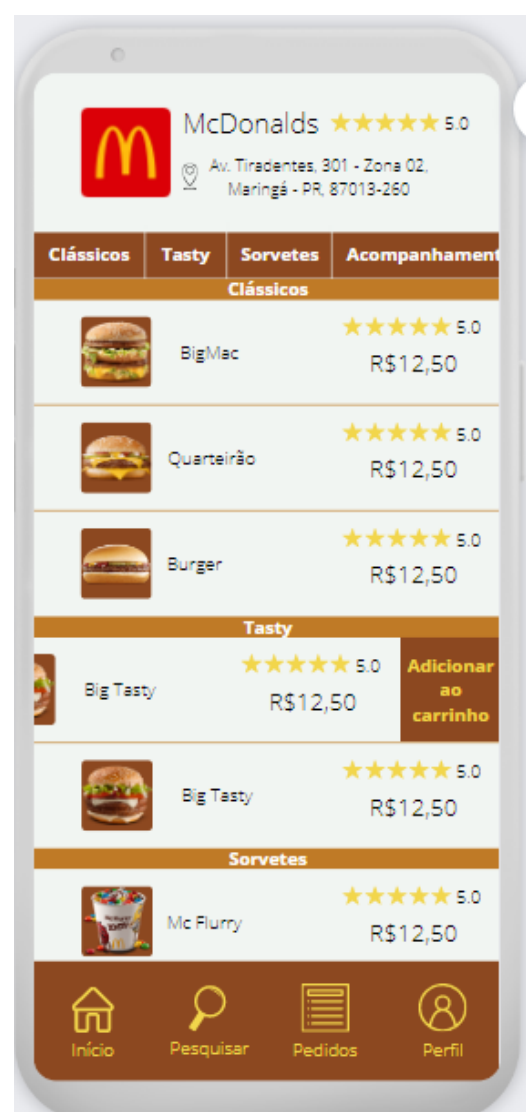
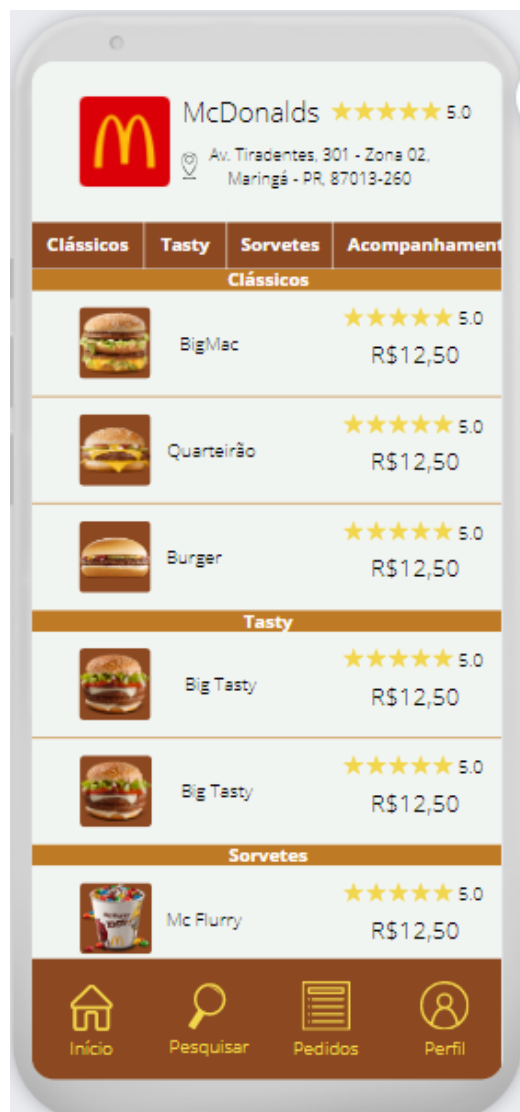
1.0.5 – Organizacionais (Operacionais e Desenvolvimento):

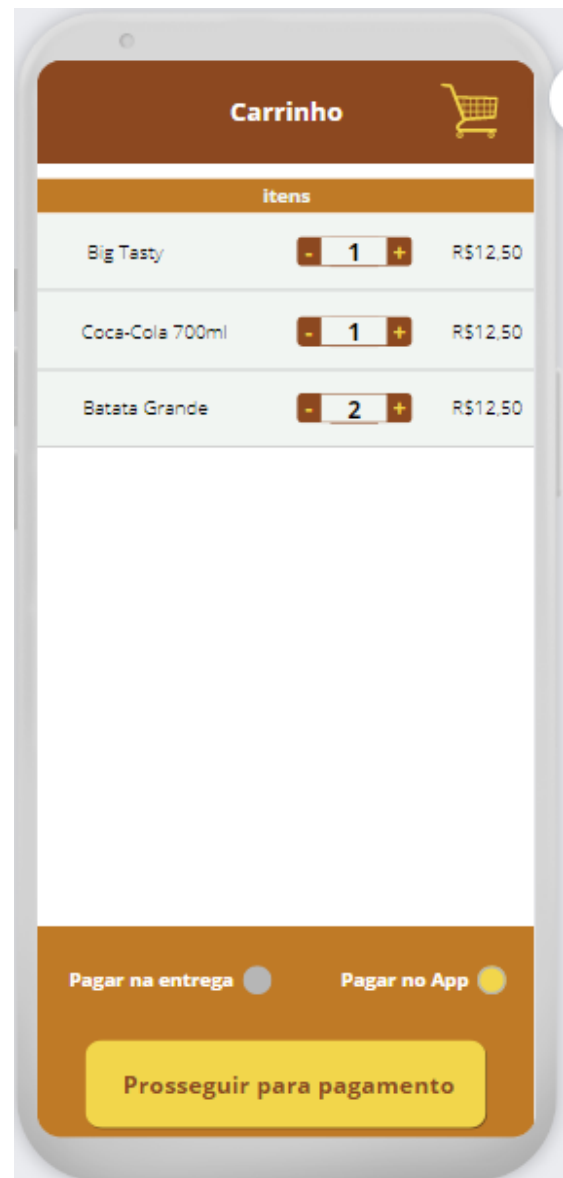
- 1.0.5.1 – Sistema Operacional: O aplicativo deverá funcionar na plataforma Android a partir da versão 5.0 ou superior.
- 1.0.5.2 – Linguagem a ser desenvolvido: O sistema será desenvolvido em Linguagem C usando banco de dados Oracle

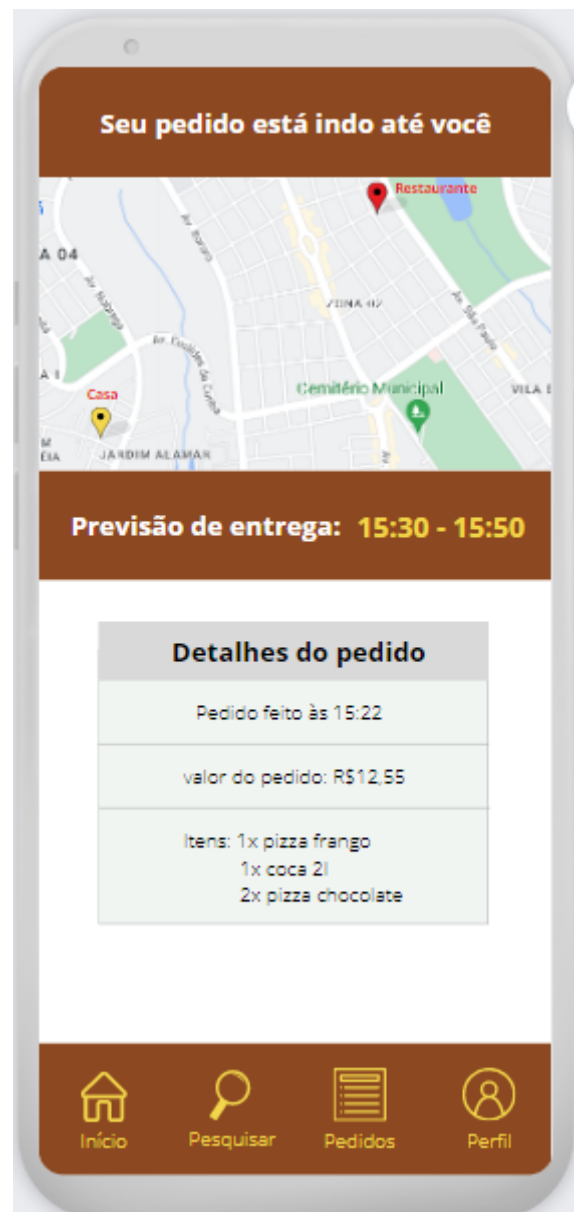
1.0.6 - Externos:

- 1.0.6.1 – Autorizações: O sistema deverá pedir autorização ao usuário para fazer o uso do geolocalização, galeria de fotos e uso da câmera, conforme estabelecido pelo Art 5º, XII da Lei Geral de Proteção de Dados Pessoais (LGPD).
- 1.0.6.2 – Maioridade: O sistema deverá aceitar somente usuários maiores de 18 anos.











+ Novo Produto

Insira a categoria:

Nome do produto (máximo 30 caracteres):

0/30 caracteres

Descrição do produto (Máximo 255 caracteres):

0/255 caracteres

Valor:

R\$

Alergênicos presentes no produto:

Nenhum

Glúten

Nozes

Molusco

+

Adicionar da galeria

Câmera

Entregas disponíveis: 5

McDonalds

Rua 11 de setembro
valor: R\$10,00
Destino: Av Paranaguá,145 ,Zona 7

Aceitar

McDonalds

Rua 11 de setembro
valor: R\$10,00
Destino: Av Paranaguá,145 ,Zona 7

Aceitar

McDonalds

Rua 11 de setembro
valor: R\$10,00
Destino: Av Paranaguá,145 ,Zona 7

Aceitar

McDonalds

Rua 11 de setembro
valor: R\$10,00
Destino: Av Paranaguá,145 ,Zona 7

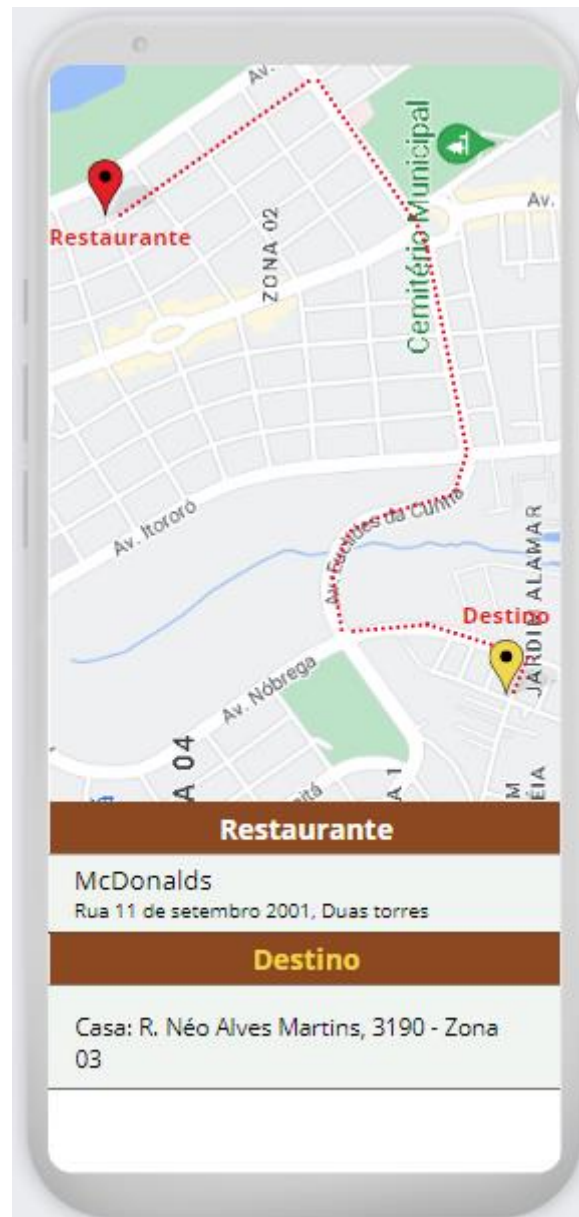
Aceitar

McDonalds

Rua 11 de setembro
valor: R\$10,00
Destino: Av Paranaguá,145 ,Zona 7

Aceitar

Perfil



Canvas

Parceiros Chave

Entregadores locais

Restaurantes locais.

Atividades Chave

Fazer uma lista de restaurantes locais (parceiros do aplicativo), dos mais próximos do local do cliente até os mais distantes;

Plataforma de Delivery voltada a alimentos prontos. (Ex.: Pizzas).

- Trazer cardápios de cada restaurante, de modo que o restaurante que ira inserir os itens e as fotos dos itens;

Quando o entregador do app ou do restaurante estiver perto do local de entrega, o aplicativo irá enviar uma notificação para o usuário que seu pedido está a 5 minutos de distância;

O aplicativo mostrará em tempo real a localização do entregador (para os motoristas que tiver o gps ativado);

O aplicativo terá um espaço para avaliar o restaurante e outro para avaliar a comida;

Recurso Chave

Computadores;

Entregadores que são parceiros do aplicativo;

Banco de Dados na nuvem;

Proposta de Valor

Mais opções para o consumidor final de benefícios e preços.

Comodidade, aplicativo intuitivo.

Taxas mais justas;

Relação com o cliente

Rede Social (Instagram, Twitter, Facebook, Youtube, TikTok).

E-mail para possíveis parcerias e eventuais dúvidas;

Canais

Play store (Google)

Aplicativo e site. Conta compartilhada.

Apple Store

Segmentos de Mercado

Todas as pessoas que utilizam smartphones e utilizam delivery.

Mercado Alimentício.

Estrutura de Custos

Publicidade/Marketing (Média de 15.000,00 por mês) ;

Primeira compra do usuário final (Quando o cliente for utilizar o app para pedir algum alimento pela primeira vez, o app irá ajudar o cliente com R\$18,00) e entrega grátis;

Custo de Infraestrutura: - Internet; (R\$1000,00) - Desenvolvimento do App (R\$150.000,00); - Água, Luz, Aluguel (R\$1000,00; R\$6.000,00; 3500,00); - Salários dos funcionários (Média de: R\$42.000,00); - Banco de dados na nuvem (USD\$600,00 a USD\$1000,00);

Convidar mais pessoas para o app por meio de convite de usuários (Ganha cupom de 10,00 reais por pessoa que convidar (limite de 10 pessoas));

Pagamento dos entregadores (entre R\$1.500,00 a R\$2.000,00);

Avaliação do app ganha prêmio (Cupom de desconto de 5,00 reais);

Fontes de Renda
Básico: Taxas de 5% por pedido, sem benefícios.
Plano de assinatura para clientes;
Premium: Taxa de 18% com entrega do app e rastreo em tempo real.
Mensalidade de R\$100 para o restaurante para ter vários outros benefícios. Ex.: Recomendação, promoções do app, consultoria. A mensalidade depende do faturamento do restaurante. Existem três tipos: Pequena, média e grande empresa.

Conclusão

Após baixarmos os aplicativos mais comuns de delivery, percebemos que faltam alguns detalhes para deixar os usuários com a maior comodidade possível. Os aplicativos atuais não avisam quando o seu pedido está chegando e mais de 90% dos casos, fazendo assim com que o usuário não fique pronto para receber seu pedido, pelos mais variados fatores, seja por que perdeu o cartão, não achou o dinheiro ou qualquer outro tipo de imprevisto.

Decidimos fazer um aplicativo de delivery de comida que suprisse as necessidades que temos após as nossas prévias experiências com os demais aplicativos, e também percebemos que os aplicativos atuais não traçam as melhores rotas para os entregadores fazerem as entregas da melhor forma possível no menor tempo possível.

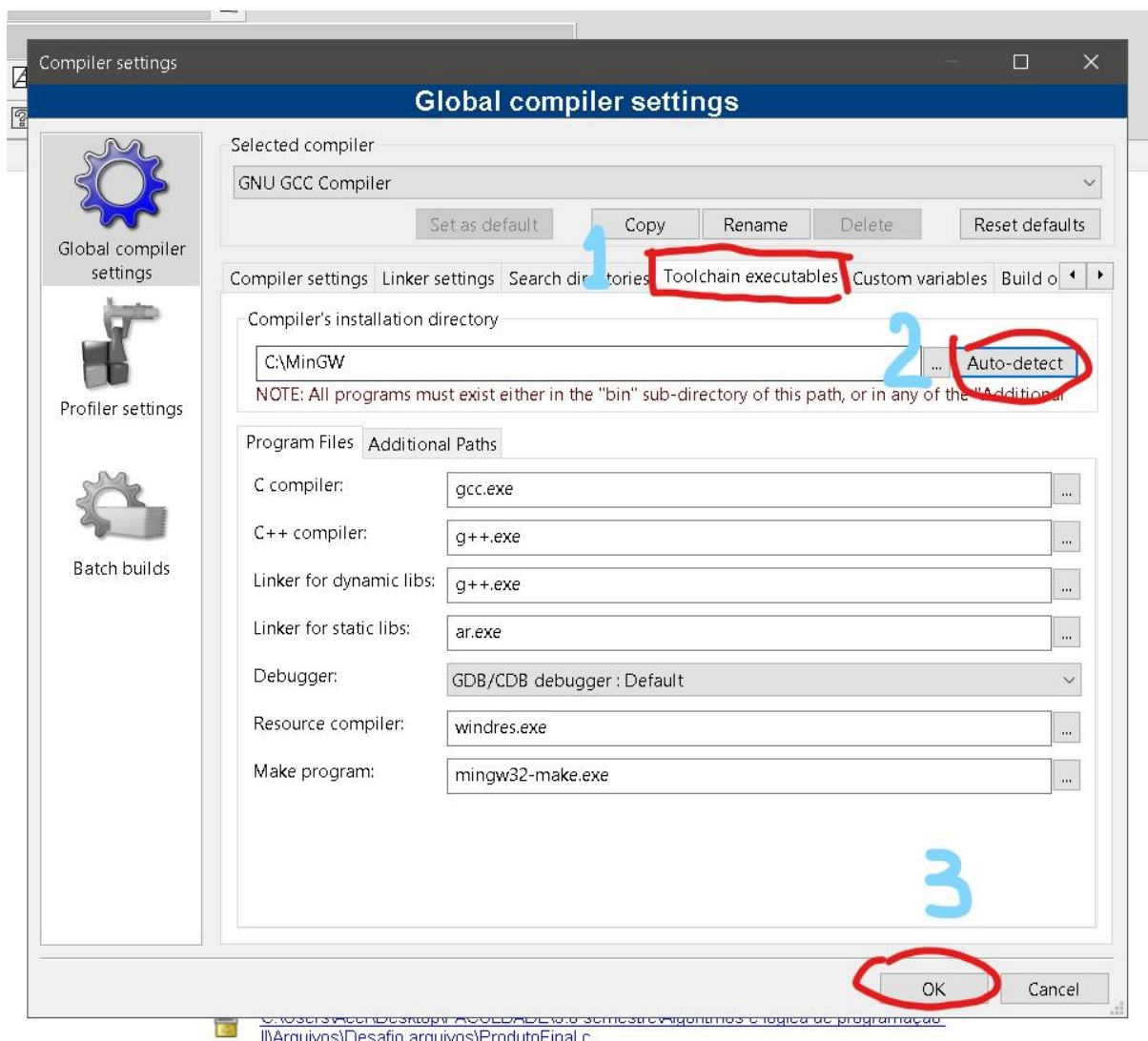
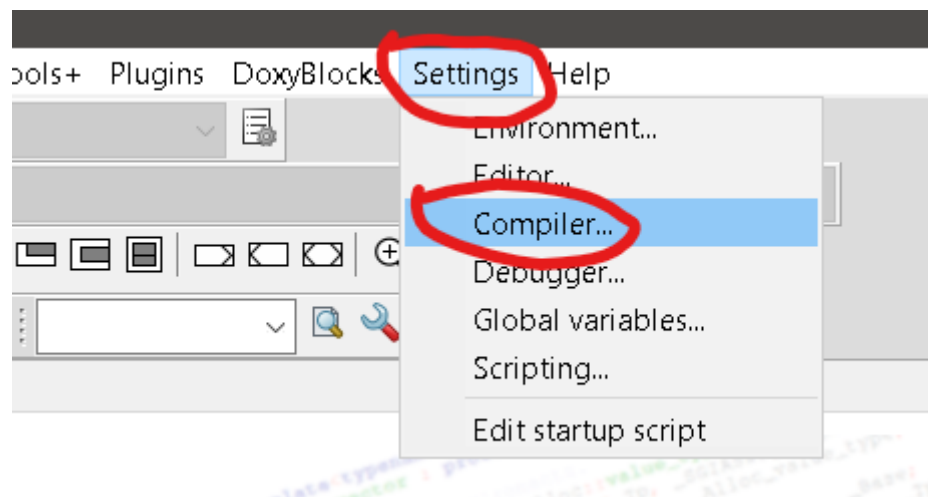
Links Úteis:

[Drive com Código em C, Assembly e Apresentação](#)

[Código em C feito no Code::Blocks, clique para fazer download da versão usada \(Pode abrir o arquivo executável para ver o programa funcionando "AEP 3.exe"\)](#)

[Vídeo da Apresentação](#)

Caso não consiga compilar e Executar o arquivo em C, siga os passos a passos conforme indicado nas imagens abaixo:



Código em C:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

```
///=====GLOBAIS=====
```

```
int n; //cliente logado
```

```
char cpf[20][20];
char nome[20][150];
char telefone[20][20];
char email[20][150];
char senha[20][30];
char endereco[20][5][200];
char cep[20][5][30];
char cartao[20][5][100];
```

```
char restaurante[3][150]={"Lanchonete do Ney", "Pizzaria do Mario", "Tabacaria
sem Cancer"};
```

```
char produto[3][5][100];
char desc[3][5][100];
float valor[3][5];
```

```
char item[10][50];
float total[10];
```

```
///=====
```

```
void copiaClientes();
void copiaClientes();
void copiaProdutos();
void menuHead(char titulo[30]);
void opInvalida();
void opcao(char op[30], int num);
```

```
int validaEmail(char aux[150]);
void info();
void enderecos();
void addAddress();
void cartoes();
void addCard();
void perfilCliente();
void pagar(int t);
void carrinho();
void printaCardapio(int x);
void restaurantes();
void menuCliente();
void cadastro();
void login();
void tela1();
```

```
///===== [MAIN] =====
```

```
int main(){
    copiaClientes();
    copiaProdutos();
    tela1();

    return 0;
}
```

```
///----- CLIENTES -----
```

```
void copiaClientes(){
    FILE*arq;
    arq = fopen("clientes/cadastros.txt", "r");

    char aux[250];
    int linha=0, j=0, i=0, c;

    if(arq == NULL){
```

```
printf("erro ao abrir o arquivo\n");
}else{
```

```
while(fgets(aux, 250, arq)!=NULL){
    linha++;
    c=0;
```

```
if(linha>2){
```

```
    //-----CPF-----
```

```
    j=0;
```

```
    while(aux[c]!=' '){
```

```
        cpf[i][j]=aux[c];
```

```
        c++;
```

```
        j++;
```

```
    }
```

```
    cpf[i][j]='\0';
```

```
    //-----NOME-----
```

```
    c=20;
```

```
    j=0;
```

```
    while(aux[c]!=' ' ||aux[c+1]!='.'){
```

```
        nome[i][j]=aux[c];
```

```
        c++;
```

```
        j++;
```

```
    }
```

```
    nome[i][j]='\0';
```

```
    //-----TELEFONE-----
```

```
    j=0;
```

```
    for(c=43; c<58; c++){
```

```
        telefone[i][j]=aux[c];
```

```
        j++;
```

```
    }
```

```
telefone[i][j]='\0';
```

```
//-----EMAIL-----
```

```
j=0;
```

```
c=64;
```

```
while(aux[c]!=' ' ||aux[c+1]!='.'){
```

```
    email[i][j]=aux[c];
```

```
    c++;
```

```
    j++;
```

```
}
```

```
email[i][j]='\0';
```

```
//-----SENHA-----
```

```
j=0;
```

```
c=103;
```

```
while(aux[c]!=' ' ||aux[c+1]!='.'){
```

```
    senha[i][j]=aux[c];
```

```
    c++;
```

```
    j++;
```

```
}
```

```
senha[i][j]='\0';
```

```
//-----ENDEREÇO-----
```

```
j=0;
```

```
c=122;
```

```
while(aux[c]!=' ' ||aux[c+1]!='.'){
```

```
    endereco[i][0][j]=aux[c];
```

```
    c++;
```

```
    j++;
```

```
}
```

```
endereco[i][0][j]='\0';
```

```
//-----CEP-----
```

```

        j=0;
        for(c=223;c<232; c++){
            cep[i][0][j]=aux[c];
            j++;
        }
        cep[i][0][j]='\0';

        i++;
    }
}
fclose(arq);
}
}

///-----RESTAURANTE E PRODUTOS-----

void copiaProdutos(){
    FILE*arq;
    char tmp[15][50], tmp2[15][100];
    arq = fopen("restaurantes/produtos.txt", "r");

    char aux[200], aux2[15][20];
    int linha =0, j=0, i=0, c;

    if(arq==NULL){
        printf("erro ao abrir o arquivo\n");
    }else{
        while(fgets(aux, 200, arq)!=NULL){
            linha ++;
            c=0;

            if(linha >2){
                j=0;
                while(aux[c]!=' ' ||aux[c+1] != '.'){
                    tmp[i][j] = aux[c];
                    c++;
                }
            }
        }
    }
}

```

```

        j++;
    }
    tmp[i][j]='\0';

    c=26;
    j=0;
    if(aux[28]==''){
        tmp2[i][0]='\0';
    }else{
        while(aux[c]!=' ' ||aux[c+1] != '.'){
            tmp2[i][j] = aux[c];
            c++;
            j++;
        }
        tmp2[i][j]='\0';
    }

    j=0;
    for(c=99; c<105; c++){
        if(aux[c]!='\n'){
            aux2[i][j]=aux[c];
            j++;
        }
    }
    aux2[i][j]='\0';
    i++;
}

}

fclose(arq);
}

for(int i=0; i<5; i++){
    strcpy(produto[0][i], tmp[i]);
    strcpy(produto[1][i], tmp[i+5]);
}

```



```
strcpy(produto[2][i], tmp[i+10]);
```

```
strcpy(desc[0][i], tmp2[i]);
```

```
strcpy(desc[1][i], tmp2[i+5]);
```

```
strcpy(desc[2][i], tmp2[i+10]);
```

```
valor[0][i] = atof(aux2[i]);
```

```
valor[1][i] = atof(aux2[i+5]);
```

```
valor[2][i] = atof(aux2[i+10]);
```

```
}
```

```
}
```

```
///-----AUXIIARES-----
```

```
void menuHead(char titulo[30]){
```

```
    int e = strlen(titulo);
```

```
    char espaco[30], under[90];
```

```
    for(int i = 0; i<e; i++){
```

```
        espaco[i]=' ';
```

```
    }
```

```
    espaco[e]='\0';
```

```
    for(int i = 0; i<e*3; i++){
```

```
        under[i] = '_';
```

```
    }
```

```
    under[e*3] = '\0';
```

```
    printf("\n %s\n", under);
```

```
    printf("|%s%s%s|\n", espaco, espaco, espaco);
```

```
    printf("|%s%s%s|\n", espaco, titulo, espaco);
```

```
    printf("|%s|\n\n\n", under);
```

```
}
```

```
void opInvalida(){
```

```
    system("cls");
```

```

printf("OPCAO INVALIDA\n\n");
system("pause");
}
void opcao(char op[30], int num){
    printf("[%d] - %s\n\n", num, op);
}
int validaEmail(char aux[150]){
    int c=0;
    do{
        if(strcmp(aux, email[c])==0 && email[c][0]!='\0'){
            return c;
        }
        c++;
    }while(c<20);
    return -1;
}
///-----METODOS DO PROGRAMA-----
void info(){
    system("cls");
    int res;
    menuHead("    Informacoes    ");

    printf("Nome:    %s\n", nome[n]);
    printf("E-mail:  %s\n", email[n]);
    printf("Telefone: %s\n", telefone[n]);
    printf("\n\n_____ \n\n");
    opcao("Voltar", 0);
    scanf("%d", &res);

    switch(res){
    case 0:
        break;
    default:
        opInvalida();
    }
}

```

```

        info();
    }
}

void enderecos(){
    system("cls");
    int res;
    menuHead("Enderecos");
    for(int i = 0; i<5; i++){
        if(endereco[n][i][0]!='\0'){
            printf("[%d] - %s\nCEP:%s", i+1, endereco[n][i], cep[n][i]);
            printf("\n_____ \n");
        }
    }
    printf("\n\n");
    opcao("Excluir", 1);
    opcao("Adicionar novo endereco\n", 2);
    opcao("Voltar", 0);
    scanf("%d", &res);
    int i;

    switch(res){
        case 0:
            break;
        case 1:
            printf("Digite o numero do endereco que deseja EXCLUIR: ");
            scanf("%d", &res);
            system ("cls");
            printf("TEM CERTEZA QUE DESEJA EXCLUIR\n");
            printf("%s\nCEP:%s\n\n", endereco[n][res-1], cep[n][res-1]);
            printf("[1] - SIM || [2] - NAO\n");
            do{
                scanf("%d", &i);
                if(i>2 || i<1){
                    opInvalida();
                }
            } while(1);
        }
    }
}

```

```

    }
}while(i>2 || i<1);

switch(i){
case 1:
    endereco[n][res-1][0]='\0';
    cep[n][res-1][0]='\0';
    printf("\nEndereco apagado com sucesso!\n");
    system("pause");
    for(int i=0; i<9; i++){
        for(int j = i; j<10; j++){
            if(endereco[n][i][0]=='\0' && endereco[n][j][0]!='\0'){
                strcpy(endereco[n][i], endereco[n][j]);
                endereco[n][j][0]='\0';
            }
        }
    }

    enderecos();
    break;
case 2:
    enderecos();
    break;
}
break;

case 2:
    addAdress();
    enderecos();
    break;
default:
    opInvalida();
    enderecos();
}

```

```

}
void addAddress(){
    int i=0;
    while(endereco[n][i][0]!='\0'){
        i++;
    }
    printf("Insira o novo endereco: ");
    fflush(stdin);
    gets(endereco[n][i]);

    printf("Insira o CEP:");
    fflush(stdin);
    gets(cep[n][i]);
    printf("\nEndereco cadastrado com sucesso!\n");
    system("pause");
}
void cartoes(){
    system("cls");
    int res;
    menuHead("Cartoes");
    for(int i = 0; i<5; i++){
        if(cartao[n][i][0]!='\0'){
            printf("[%d] - %s", i+1, cartao[n][i]);
            printf("\n_____ \n");
        }
    }
    printf("\n\n");
    opcao("Excluir", 1);
    opcao("Adicionar novo cartao\n", 2);
    opcao("Voltar", 0);
    scanf("%d", &res);
    int i;

    switch(res){

```

case 0:

break;

case 1:

printf("Digite o numero do cartao que deseja EXCLUIR: ");

scanf("%d", &res);

system("cls");

printf("TEM CERTEZA QUE DESEJA EXCLUIR: ");

printf("%s\n\n", cartao[n][res-1]);

printf("[1] - SIM || [2] - NAO\n");

do{

scanf("%d", &i);

if(i>2 || i<1){

opInvalida();

}

}while(i>2 || i<1);

switch(i){

case 1:

cartao[n][res-1][0]='\0';

printf("\nCartao apagado com sucesso!\n");

system("pause");

for(int i=0; i<9; i++){

for(int j = i; j<10; j++){

if(cartao[n][i][0]=='\0' && cartao[n][j][0]!='\0'){

strcpy(cartao[n][i], cartao[n][j]);

cartao[n][j][0]='\0';

}

}

}

cartoes();

break;

case 2:

cartoes();

```

        break;
    }
    break;

case 2:
    addCard();
    cartoes();
    break;
default:
    opInvalida();
    cartoes();
}
}

void addCard(){
    int i=0;
    while(cartao[n][i][0]!='\0'){
        i++;
    }
    printf("Insira o novo cartao: ");
    fflush(stdin);
    gets(cartao[n][i]);
    printf("\nCartao cadastrado com sucesso!\n");
    system("pause");
}

void perfilCliente(){
    system("cls");
    int res;
    menuHead("    Perfil    ");

    opcao("Informacoes pessoais", 1);
    opcao("Enderecos", 2);
    opcao("Cartoes", 3);
    opcao("Logout\n", 4);
    opcao("Voltar", 0);

```

```

scanf("%d", &res);

switch(res){
case 0:
    break;
case 1:
    info();
    perfilCliente();
    break;
case 2:
    enderecos();
    perfilCliente();
    break;
case 3:
    cartoes();
    perfilCliente();
    break;
case 4:
    tela1();
    break;
default:
    opInvalida();
    perfilCliente();
}
}

void pagar(int t){
    system("cls");
    int check=0, check2=0;
    menuHead("Pagamento");

    for(int i = 0; i<5; i++){
        if(endereco[n][i][0]!='\0'){
            printf("[%d] - %s", i+1, endereco[n][i]);
            printf("\n_____ \n");

```



```

        check2++;
    }
}

int res, k;

if(check2==0){
    printf("\nNENHUM ENDERECO CADASTRADO\n");
    addAddress();
    k=0;
}else{
    printf("\nQual endereco deseja usar?\n");
    scanf("%d", &k);
}

for(int i = 0; i<5; i++){
    if(cartao[n][i][0]!='\0'){
        printf("[%d] - %s", i+1, cartao[n][i]);
        printf("\n_____ \n");
        check++;
    }
}

if(check==0){
    printf("\nNENHUM CARTAO CADASTRADO\n");
    addCard();
    res=0;
}else{
    printf("\nQual cartao deseja usar?\n");
    scanf("%d", &res);
}

printf("\n\nPAGAMENTO REALIZADO!\nSeu pedido chegará entre 40 minutos e 1
hora\nNo endereco: %s\n\n", endereco[n][k]);
for(int i=0; i<10; i++){
    item[i][0] = '\0';
}

```

```

        total[i]=0.0;
    }
    system("pause");
}

void carrinho(){
    system("cls");
    menuHead("Pagamento");
    int res, i, op=0;
    float tot=0;

    printf("Sua compra foi:\n");
    printf("_____ \n");
    for(int i=0; i<10; i++){
        if(item[i][0]!='\0'){
            printf("\n[%d] - %-30s R$%-6.2f \n", i+1, item[i], total[i]);
            tot+=total[i];
        }
    }
    printf("_____ TOTAL: R$%.2f _____ \n", tot);

    opcao("Pagar", 1);
    opcao("Excluir item", 2);
    opcao("Voltar", 0);
    scanf("%d", &res);
    switch(res){
        case 0:
            break;
        case 1:
            pagar(tot);
            menuCliente();
            break;
        case 2: //EXCLUIR ITEM
            if(item[0][0] == '\0'){
                printf("\nNao tem itens para serem excluidos\n");
            }
    }
}

```

```

    system("pause");
    carrinho();
}
else{
    printf("Digite o numero do item que deseja EXCLUIR: ");
    scanf("%d", &op);
    system ("cls");
    printf("TEM CERTEZA QUE DESEJA EXCLUIR\n");
    printf("%s\n\n", item[op-1]);
    printf("[1] - SIM || [2] - NAO\n");
    do{
        scanf("%d", &i);
        if(i>2 || i<1){
            opInvalida();
        }
    }while(i>2 || i<1);

    switch(i){
    case 1:
        item[op-1][0]='\0';
        total[op-1]=0.0;
        printf("\nItem apagado com sucesso!\n");
        system("pause");

        for(int i=0; i<9; i++){
            for(int j = i; j<10; j++){
                if(item[i][0]=='\0' && item[j][0]!='\0'){
                    strcpy(item[i], item[j]);
                    item[j][0]='\0';
                    total[i]=total[j];
                    total[j]=0.0;
                }
            }
        }
    }
    carrinho();
}

```

```

        break;
    case 2:
        carrinho();
        break;
    }
}
break;
default:
    opInvalida();
    carrinho();
}
}

void printaCardapio(int x){
    system("cls");
    menuHead(restaurante[x]);

    printf("\nO que deseja adicionar ao carrinho?\n");
    for(int i=0; i<5; i++){
        printf("\n[%d] - %-30s R$%.2f\n%s", i+1, produto[x][i], valor[x][i], desc[x][i]);
        printf("\n_____ \n");
    }
    opcao("Voltar", 0);
    int res, c=0;
    scanf("%d", &res);
    while(res>5 || res<0){
        opInvalida();
        printaCardapio(x);
    }
    res-=1;
    switch(res){
    case -1:
        break;
    default:
        while(item[c][0]!='\0'){

```

```

        c++;
    }
    strcpy(item[c], produto[x][res]);
    total[c]=valor[x][res];
    system("cls");
    printf("Voce adicionou: %s a sacola\nDeseja adicionar mais itens?\n [1] - SIM ||
[2] - NAO\n", produto[x][res]);
    scanf("%d", &res);

    switch(res){
    case 1:
        printaCardapio(x);
        break;
    case 2:
        carrinho();
        printaCardapio(x);
        break;
    default:
        opInvalida();
        printaCardapio(x);
    }
}
}

void restaurantes(){
    system("cls");
    menuHead("  Restaurantes  ");
    int res;

    for(int i=0; i<3; i++){
        opcao(restaurante[i], i+1);
        printf("_____ \n");
    }
    printf("\n[0] - Voltar\n\n");
    scanf("%d", &res);

```

```

while(res>3 || res<=-1){
    opInvalida();
    restaurantes();
}
res-=1;

switch(res){
case -1:
    break;
default:
    printaCardapio(res);
    restaurantes();
}
}
void menuCliente(){

    system("cls");
    int res;
    menuHead("    Ai Qui Dilica    ");

    opcao("Restaurantes", 1);
    opcao("Carrinho", 2);
    opcao("Perfil", 3);
    scanf("%d", &res);

    switch(res){
case 1:
    restaurantes();
    menuCliente();
    break;
case 2:
    carrinho();
    menuCliente();
    break;

```

```

case 3:
    perfilCliente();
    menuCliente();
    break;
default:
    opInvalida();
    menuCliente();
}
}

void cadastro(){
    system("cls");
    char aux[150];
    int i=0;

    while(email[i][0]!='\0'){
        i++;
    }

    do{
        system("cls");
        menuHead("  CADASTRO  ");
        printf("Insira seu email: ");
        fflush(stdin);
        gets(aux);
        if(validaEmail(aux)!=-1){
            printf("\nEmail já cadastrado\n\n");
        }
    }while(validaEmail(aux)!=-1);
    strcpy(email[i], aux);
    do{
        printf("Confirme seu email: ");
        fflush(stdin);
        gets(aux);
        if(strcmp(aux, email[i])!=0){

```

```

        printf("\n\nOs emails nao sao iguais!\n\n");
        system("pause");
    }
}while(strcmp(aux, email[i])!=0);

do{
    printf("Insira uma senha: ");
    fflush(stdin);
    gets(senha[i]);

    system("cls");
    menuHead("  CADASTRO  ");
    printf("Confirme sua senha: ");
    fflush(stdin);
    gets(aux);
    if(strcmp(aux, senha[i])!=0){
        printf("\n\nAs senhas nao sao iguais!\n\n");
        system("pause");
    }
}while(strcmp(aux, senha[i])!=0);
printf("Um E-mail de confirmação foi enviado para: %s", email[i]);

tela1();
}

void login(){
    char aux[150];
    char aux2[50];
    int em, res=-1;

    system("cls");
    menuHead("  LOGIN  ");

    fflush(stdin);
    printf("EMAIL: ");

```



```

gets(aux);
em=validaEmail(aux);
if(em==-1){
    system("cls");
    printf("Email nao cadastrado\n\n");
    system("pause");
}else{
    do{
        fflush(stdin);
        printf("SENHA: ");
        gets(aux2);
        if(strcmp(aux2, senha[em])!=0){
            printf("Senha incorreta!\n\n");
            system("pause");
        }
    }while(strcmp(aux2, senha[em])!=0);
    n=em;
    menuCliente();
}

}

void tela1(){

    for(int i=0; i<10; i++){
        item[i][0] = '\0';
        total[i]=0.0;
    }

    system("cls");
    int res;
    menuHead("    Ai Qui Dilica    ");

    opcao("fazer login", 1);
    opcao("criar conta", 2);

```

```

scanf("%d", &res);
switch(res){
case 1:
    login();
    tela1();
    break;
case 2:
    cadastro();
    break;
default:
    opInvalida();
    tela1();
}
}

```

Código em Assembly:

```

.data
    #MENSAGENS
    bemvindo:          .asciiz "      Bem-vindo ao App do AI QUI DILIÇA"
    pula:              .asciiz "\n"
    telacadastral:     .asciiz "      Tela de Cadastro"
    pnome:             .asciiz "Insira o seu nome completo: "
    pcidade:          .asciiz "Insira a sua Cidade: "
    puf:               .asciiz "UF: "
    pdatadenascimento: .asciiz "Data de Nascimento(ex.: 10/02/1988): "
    pemail:            .asciiz "Insira o seu e-mail: "
    psenha:            .asciiz "Insira a sua Senha: "
    cadastroef:        .asciiz " * Cadastro efetuado com sucesso! *"

    imprimenome:       .asciiz "      Nome: "
    imprimecidade:     .asciiz "      Cidade: "
    imprimeuf:         .asciiz "      UF: "
    imprimdatadenascimento: .asciiz "Data de Nascimento: "
    imprimeemail:      .asciiz "      E-mail: "

```

imprimesenha: .asciiz " Senha: "

#Variaveis

nome: .space 30

cidade: .space 30

uf: .space 3

datanascimento: .space 12

email: .space 50

senha: .space 16

.text

li \$v0,4

la \$a0, bemvindo

syscall

li \$v0,4

la \$a0,pula

syscall

jal cadastro

jal printinfo

jal saida

cadastro:

li \$v0,4

la \$a0, telacadastral

syscall

li \$v0,4

la \$a0,pula

syscall

li \$v0,4

la \$a0,pula

syscall

#----- NOME -----

li \$v0,4

la \$a0, pnome

syscall

li \$v0,8

la \$a0, nome

la \$a1, 30

syscall

move \$s0,\$a0

#----- CIDADE -----

li \$v0,4

la \$a0, pcidade

syscall

li \$v0,8

la \$a0, cidade

la \$a1, 30

syscall

move \$s1,\$a0

#----- ESTADO -----

li \$v0,4

la \$a0, puf

syscall

```
li $v0,8
la $a0, uf
la $a1, 3
syscall
```

```
move $s2,$a0
```

```
li $v0,4
la $a0,pula
syscall
```

```
#----- DATA DE NASCIMENTO -----
```

```
li $v0,4
la $a0, pdatadenascimento
syscall
```

```
li $v0,8
la $a0, datanascimento
la $a1, 12
syscall
```

```
move $s3,$a0
```

```
#----- EMAIL -----
```

```
li $v0,4
la $a0, pemail
syscall
```

```
li $v0,8
la $a0, email
la $a1, 50
```

syscall

move \$s4,\$a0

#----- SENHA -----

li \$v0,4

la \$a0, senha

syscall

li \$v0,8

la \$a0, senha

la \$a1, 10

syscall

move \$s5,\$a0

li \$v0,4

la \$a0,pula

syscall

li \$v0,4

la \$a0,cadastroef

syscall

li \$v0,4

la \$a0,pula

syscall

li \$v0,4

la \$a0,pula

syscall

jr \$ra

printinfo:

#----- IMPRIME NOME -----

li \$v0,4

la \$a0,imprimenome

syscall

li \$v0,4

la \$a0, (\$s0)

syscall

#----- IMPRIME CIDADE -----

li \$v0,4

la \$a0,imprimecidade

syscall

li \$v0,4

la \$a0, (\$s1)

syscall

#----- IMPRIME ESTADO -----

li \$v0,4

la \$a0,imprimeuf

syscall

li \$v0,4

la \$a0, (\$s2)

syscall

li \$v0,4

la \$a0,pula

syscall

#----- IMPRIME DATA DE NASCIMENTO -----

li \$v0,4

la \$a0,imprimedatadenascimento

syscall

li \$v0,4

la \$a0, (\$s3)

syscall

#----- IMPRIME E-MAIL -----

li \$v0,4

la \$a0,imprimeemail

syscall

li \$v0,4

la \$a0, (\$s4)

syscall

#----- IMPRIME SENHA -----

li \$v0,4

la \$a0,imprimesenha

syscall

li \$v0,4

la \$a0, (\$s5)

syscall

jr \$ra

saida:

li \$v0,10

syscall

REFERÊNCIAS:

Disponível em: <https://blog.neemo.com.br/aplicativos-de-delivery-marketplace/>. Acesso em 14 abr. 2022.

Disponível em: <https://saipos.com/ifood/taxa-ifood>. Acesso em 14 abr. 2022.

Disponível em: <https://help.grandchef.com.br/o-crescimento-do-delivery-no-brasil-em-2021/#:~:text=O%20Brasil%20foi%20respons%C3%A1vel%20por,seguido%20dos%20aplicativos%20de%20entregas>. Acesso em 14 abr. 2022.

Disponível em: <https://valorinveste.globo.com/objetivo/empreenda-se/noticia/2020/04/02/com-quarentena-apps-de-entregas-sao-oportunidade-para-trabalhadores-e-comercios.ghtml>
<https://abrasel.com.br/noticias/noticias/ifood-ficou-tao-grande-que-afeta-concorrencia-e-clientes-dizem-analistas/>. Acesso em 14 abr. 2022.

Disponível em: <https://www.novanews.com.br/noticias/variedades/mercado-de-delivery-cresce-no-brasil-e-preve-aumento-nos-proximos-anos>. Acesso em 14 abr. 2022.

PERSEGUINE, R, V. Engenharia de Requisitos. Maringá: UniCesumar, 2016.

Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 10 jun. 2022.