

Capstone Project

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Cardápio do Dia](#)

[Pedido - Adicionar Item](#)

[Pedidos](#)

[Pedido - Detalhes](#)

[Fidelidade](#)

[Fidelidade - Leitor QR Code](#)

[Widget - Cardápio](#)

[Notificações](#)

[Considerações Chave/Key Considerations](#)

[Como seu app vai tratar a persistência de dados?](#)

[Descreva qualquer caso de uso específico \(“corner case”\) da experiência do Usuário \(UX\).](#)

[Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.](#)

[Descreva como você implementará o Google Play Services.](#)

[Próximos Passos: Tarefas Necessárias](#)

[Tarefa 1: Configuração do Projeto/Project Setup](#)

[Tarefa 2: Implementar a Interface de Usuário \(UI\) para cada Activity e Fragment](#)

[Tarefa 3: Integração Firebase](#)

[Tarefa 4: MenuFragment - Cardápio do Dia](#)

[Tarefa 5: AddPlateToOrderFragment - Pedido - Adicionar Item](#)

[Tarefa 6: OrdersFragment - Pedidos](#)

[Tarefa 7: OrderFragment - Pedido Detalhes](#)

[Tarefa 8: LoyaltyProgramFragment - Fidelidade](#)

[Tarefa 9: QRCodeReaderActivity - Fidelidade - Leitor QR Code](#)

[Tarefa 10: Widget](#)

[Tarefa 11: Notificações](#)

Usuário do GitHub: [ericksprengel](#)

Marmitop

Descrição

Agora o Marmitop mudou a forma como você faz a reserva dos seus pedidos.

As reservas antes feitas pelo Whatsapp ficarão muito mais práticas. Com poucos cliques e sua reserva para o dia seguinte está feita.

O aplicativo oferece cardápio do dia seguinte, reserva de pedido, visualização do histórico de pedidos e controle do programa de fidelidade.

* Marmitop é uma empresa de delivery de almoço que trabalha com reservas diárias.

Atualmente as reservas são feitas por Whatsapp. Os clientes recebem diariamente o cardápio do dia seguinte e fazem as reservas diretamente pelo Whatsapp.

Público-Alvo/Intended User

Feito para clientes da Marmitop.

Funcionalidades/Features

- Visualização do cardápio do dia seguinte
- Fazer reserva de pedido*
- Visualização do histórico de pedidos
- Modificação do pedido*
- Visualização do programa de fidelidade
- Check-in no programa fidelidade (feita na retirada do pedido)
- Notificação de disponibilidade do cardápio do dia seguinte
- Notificação de pedido disponível para retirada
- Notificação de alerta de tempo limite para reserva próximo de expirar


* Reservas só podem ser feitas/alteradas até às 10:00

Protótipo de Interfaces do Usuário

Cardápio do Dia

<div> <div>Cardápio 24/10</div> <div>Reserva de pedidos feitas até às 10:00</div> <div> <div>Prato 1: Feijoada Light</div> <div>arroz, feijão preto, carne seca, pernil, calabresa, couve e farofa</div> </div> <div> <div>Prato 2: Peito de Frango ao forno com catupiry</div> <div>arroz, feijão e filé de frango</div> </div> <div> <div>Prato 3: Carne Louca</div> <div>arroz, feijão e purê de batata</div> </div> <div> <div>Prato 4: Prato Vegetariano</div> <div>arroz, feijão, carne de soja com cenoura e batata</div> </div> <div> <div>Prato 5: Prato Fitness</div> <div>arroz integral, filé de frango grelhado, abobrinha com manjericao e alho</div> </div> <div> <div>Pedidos</div> <div>Cardápio</div> <div>Fidelidade</div> </div> </div>	<p>"Cardápio do dia" exibe todas as opções de pratos com uma descrição da composição do item. Ao clicar em um item o usuário será redirecionado para "Pedido - Adicionar Item".</p>
---	---

Pedido - Adicionar Item

<div> <div>Pedido</div> <div> <div>Opções:</div> <div> <div><input type="radio"/></div> <div>500g</div> <div>R\$14,00</div> </div> <div> <div><input checked="" type="radio"/></div> <div>850g</div> <div>R\$19,00</div> </div> </div> <div> <div>Observações:</div> <div></div> </div> <div>Quantidade</div> <div> <div>Adicionar ao pedido</div> <div></div> </div> </div>	<p>"Pedido - Adicionar Item" é a tela onde o usuário escolhe as opções disponíveis para o item pedido: tamanho, observações especiais e quantidade. Ao clicar em "Adicionar ao pedido" o usuário será direcionado para a tela de pedidos.</p>
---	---

Pedidos


Pedidos		
Reserva de pedidos feitas até às 10:00		
24/01/18 		
Feijoada Light 850g		R\$19,00
Salada Simples		R\$5,00
23/01/18		
Strogonoff de frango 500g		R\$14,00
Virado à Paulista 500g		R\$14,00
19/01/18		
Pernil Assado 500g		R\$14,00
Salada Simples		R\$5,00
18/01/18		
Filé de Frango 500g		R\$14,00
Alterar pedido do dia (24/01) 		
Pedidos	Cardápio	Fidelidade

"Pedidos" exibe o histórico de pedidos do usuário em ordem decrescente, do mais recente para o mais antigo. O usuário pode abrir "Pedido - Detalhes" para visualizar o pedido do dia e editá-lo caso o prazo não tenha expirado.

Pedido - Detalhes

<div><div>Pedido 24/01</div><div><div>Feijoada Light 850g</div><div>R\$19,00</div><div><div>-</div><div>1</div><div>+</div></div></div><div><div>Salada Simples</div><div>R\$5,00</div><div><div>-</div><div>1</div><div>+</div></div></div><div><div>Total</div><div>R\$24,00</div></div></div>	<p>"Pedido - Detalhes" exibe a lista de itens pedidos, valor total e permite que o usuário altere a quantidade de cada item da lista caso o prazo para alteração do pedido não tenha expirado.</p>
--	--

Fidelidade

<div><div>Fidelidade</div><div><div>✓</div><div>✓</div><div>✓</div><div>✓</div><div>✓</div><div>6</div><div>7</div><div>8</div><div>9</div><div>10</div><div>11</div><div>12</div></div><div>O 13º item é por nossa conta!</div><div></div><div><div>Pedidos</div><div>Cardápio</div><div>Fidelidade</div></div></div>	<p>"Fidelidade" exibe a "pontuação" atual do usuário e o um botão para ir para tela "Fidelidade - Leitor QR Code". A cada <i>check-in</i> um item da lista é marcado com um <i>check</i> e ao final de 12 check-ins o usuário ganha um almoço.</p>
---	--

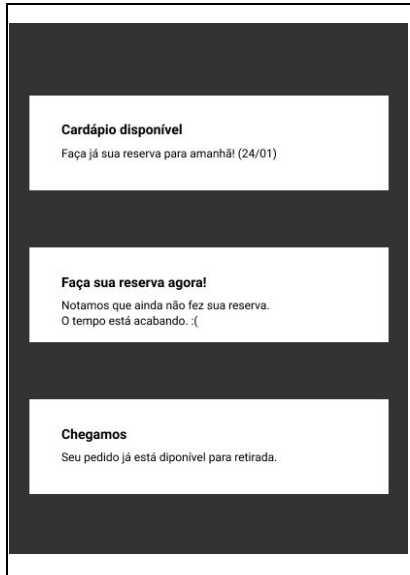
Fidelidade - Leitor QR Code

	<p>"Fidelidade - Leitor QR Code" é a tela de leitura do QR Code para o programa de fidelidade.</p>
--	--

Widget - Cardápio

	<p>"Widget - Cardápio" exibe o cardápio do dia caso já esteja disponível.</p>
--	---

Notificações

	<p>3 tipos de notificações são enviadas ao usuário:</p> <ul style="list-style-type: none"> ● Cardápio disponível: todo final do dia o Cardápio para o próximo dia é inserido no sistema e então é enviada uma notificação para que o cliente possa fazer sua reserva; ● Alerta de expiração de reserva: todo dia às 9:00 o usuário que não fez reserva será notificado ● Pedido disponível para retirada: por volta de 11:30 os pedidos ficam disponíveis para retirada. <p>* O modelo de negócio atual utiliza o Whatsapp para estas notificações. O aplicativo apenas replica esta funcionalidade.</p>
---	---

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Os dados serão persistidos na nuvem utilizando o Firebase Realtime Database. O armazenamento temporário/offline será abstraído pelo Firebase Realtime Database (<https://firebase.google.com/docs/database/android/offline-capabilities>).

O controle de sessão será feito através do FirebaseUI (<https://github.com/firebase/FirebaseUI-Android/blob/master/auth/README.md>).

O banco de dados utilizando o Cloud Firestore irá armazenar:

- Cardápio do dia (lista de pratos)
- Lista de pedidos de cada usuário (cada pedido é composto por uma lista de pratos, sendo cada prato composto pelas opções do usuário)
- Fidelidade: lista de check-ins de cada usuário

* Esta lista acima será modelada na Tarefa 3.

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

- Na tela de exibição de detalhes do pedido do dia, caso a quantidade seja de um item da lista seja reduzida a zero, um popup de confirmação de remoção do item confirmará se o item será removido
- Na tela de Cardápio, uma mensagem de "cardápio não disponível" deverá ser exibida caso o cardápio ainda não esteja disponível
 - O serviço Marmitop não funciona aos sábados, domingos e feriados. Para melhorar a experiência do usuário, ao invés de exibir "cardápio ainda não disponível", em dias que o serviço não funcionará; será utilizada uma api externa para verificar se é um feriado e exibir uma mensagem mais adequada. (A lista de feriados será carregada do serviço calendario.com.br através de uma AsyncTask)
- Ao tentar fazer um checkin no programa de fidelidade com um QR Code inválido, será exibido um popup de erro. Após o popup dispensar o popup, o aplicativo se manterá na tela de leitura de QR Code

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

- Butter Knife: para facilitar o mapeamento das Views e melhorar manutenibilidade (<http://jakewharton.github.io/butterknife/>)

Descreva como você implementará o Google Play Services.

Descreva quais Google Play Services você utilizará e como.

- Google Vision: usado para ler o QR Code para o programa de fidelidade (<https://developers.google.com/vision/android/barcodes-overview>)
- Firebase:
 - Autenticação: autenticação dos usuários através de email, Facebook e Google (<https://firebase.google.com/docs/auth/android/firebaseui>)
 - Firebase Realtime Database: armazenamento de dados: cardápio, pratos, pedidos e programa de fidelidade (<https://firebase.google.com/docs/database/>)
- FCM: envio de notificações: cardápio disponível e pedidos disponíveis (<https://firebase.google.com/docs/cloud-messaging/>)

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto/Project Setup

- Criação do projeto Android
- Criação do projeto no Google Play Services
- Configuração do Firebase

Tarefa 2: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

- Construir a UI da MainActivity com Bottom Navigation composta por 3 fragments:
 - MenuFragment: Cardápio do Dia
 - OrdersFragment: Pedidos
 - LoyaltyProgramFragment: Fidelidade
- Construir a UI da AddPlateToOrderActivity: Pedido - Adicionar Item
- Construir a UI da OrderActivity: Pedido - Detalhes
- Construir a UI da QRCodeReaderActivity: Fidelidade - Leitor QR Code

* Nesta tarefas todas as telas terão apenas informações dummy para completar a navegação proposta no wireframe.

Tarefa 3: Integração Firebase

- Modelar banco para persistir itens abaixo:
 - Cardápio,
 - Pedidos
 - Fidelidade
- Inserir cardápio manualmente no banco
- Testar integração com Firebase para carregar itens do cardápio

Tarefa 4: MenuFragment - Cardápio do Dia

- Implementar layout
- Implementar PlateAdapter
- Carregar cardápio da nuvem (inserido manualmente na Tarefa 3)

Tarefa 5: AddPlateToOrderFragment - Pedido - Adicionar Item

- Implementar layout
- Carregar opções do prato (tamanho ou tipo)
- Salvar prato ao pedido

Tarefa 6: OrdersFragment - Pedidos

- Implementar layout
- Implementar OrderAdapter
- Carregar pedidos da nuvem

Tarefa 7: OrderFragment - Pedido Detalhes

- Implementar layout
- Implementar OrderPlateAdapter
- Carregar pratos do pedido
- Exibir valor total do pedido

Tarefa 8: LoyaltyProgramFragment - Fidelidade

- Implementar layout
- Implementar LoyaltyCheckInAdapter
- Carregar quantidade de check-ins

Tarefa 9: QRCodeReaderActivity - Fidelidade - Leitor QR Code

- Implementar layout
- Implementar integração com Google Vision
- Implementar validação de QR Code

Tarefa 10: Widget

- Implementar layout
- Implementar serviço para lista de RemoteViews
- Implementar ação para abrir cardápio no aplicativo

Tarefa 11: Notificações

- Implementar integração com FCM

* Por estar fora de escopo, as mensagens serão enviadas manualmente pelo console do FCM.

