

Como utilizar este Template

1. Faça uma cópia [File → Make a copy...]
2. Renomeie este arquivo para: **“Capstone_Stage1”**
3. Substitua todos os textos **em verde**

Instruções para Envio

1. Assim que completar todas as partes, faça o download deste documento como um PDF [File → Download as PDF]
 2. Crie um repositório novo no Github para o capstone. Dê o nome **“Capstone Project”**
 3. Adicione este documento no seu repositório. Tenha certeza que está nomeado como **“Capstone_Stage1.pdf”**
-

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Tela 1](#)

[Tela 2](#)

[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: Implement UI for Each Activity and Fragment](#)

[Tarefa 3: Your Next Task](#)

[Tarefa 4: Your Next Task](#)

[Tarefa 5: Your Next Task](#)

Usuário do GitHub: jfrancodardengo

Peça Me

Descrição

O app permite que os atendentes do estabelecimento possam atender os clientes, criar os pedidos, adicionar e remover itens da comanda, conferir o valor de cada mesa e enviar os mesmos através de uma rede ou impressora, para o setor de produção. Dessa forma, isso evita que os pedidos se percam e os clientes fiquem insatisfeitos quando seus pedidos chegam de maneira errada à sua mesa.

Público-Alvo/Intended User

Estabelecimentos alimentícios

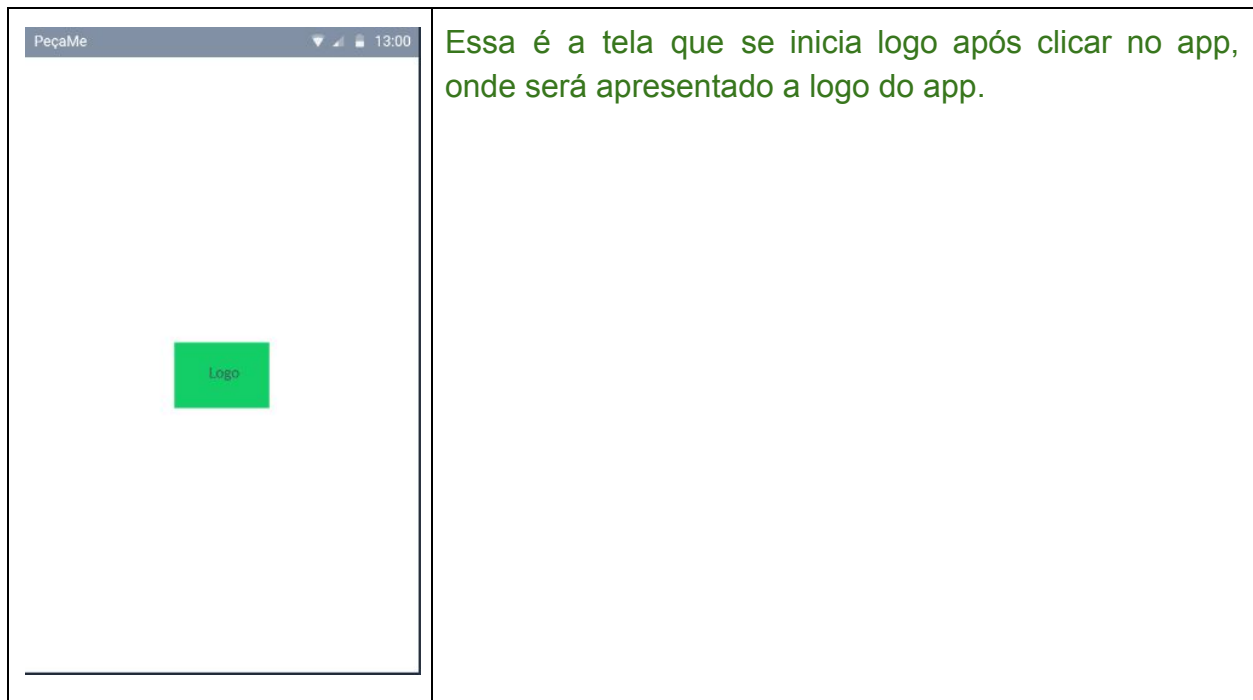
Funcionalidades/Features

Liste as principais funcionalidades do seu app. Por exemplo:

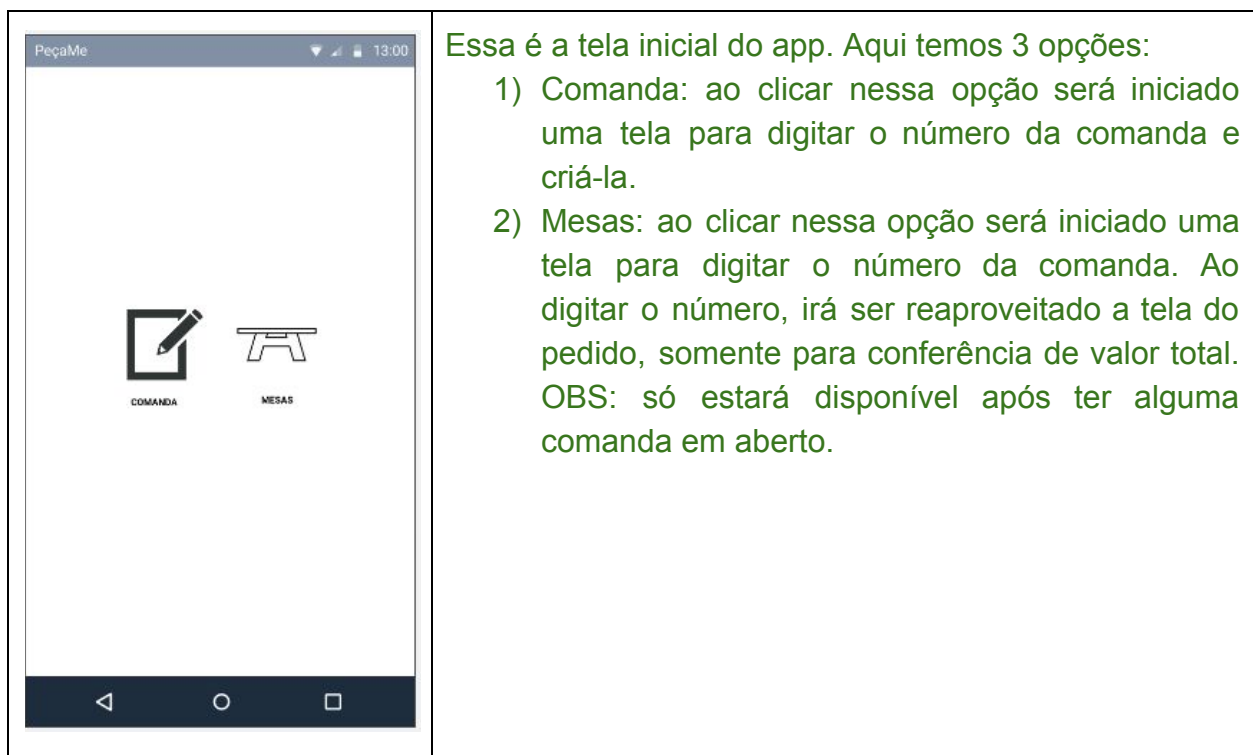
- Salvar informações em um banco de dados
- Criação de pedidos
- Envio de pedidos para setor de produção
- Conferência de consumo

Protótipo de Interfaces do Usuário

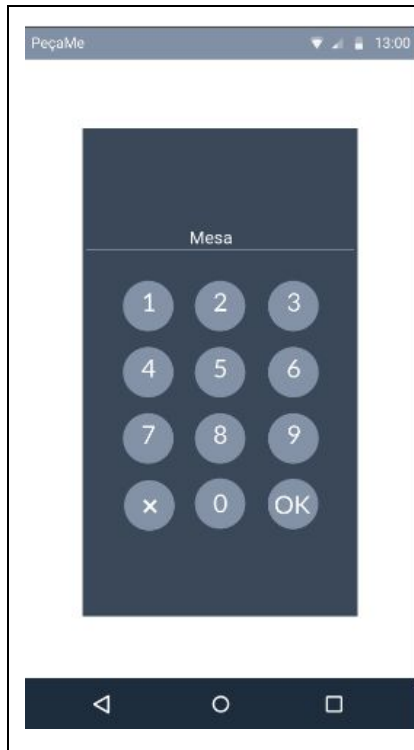
Tela 1



Tela 2



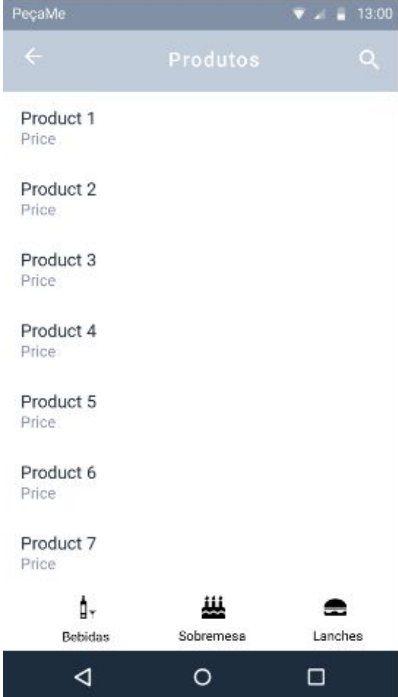
Tela 3



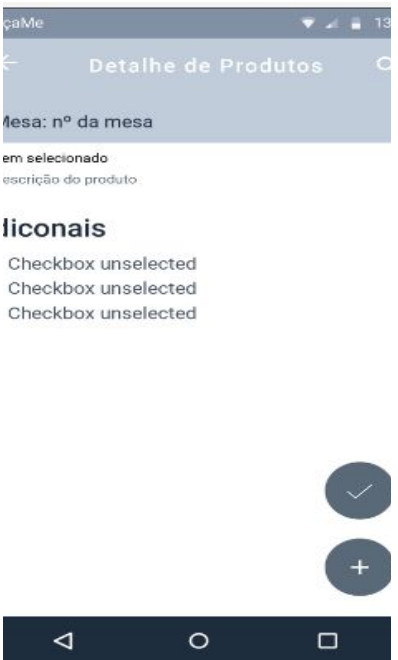
Essa tela aparece em duas situações:

- 1) Quando eu clico na opção de comanda: você digita o número da mesa que vai abrir a conta. Após dar OK, irá aparecer a tela com os produtos.
- 2) Quando clico na opção mesas: você digita o número da mesa que quer conferir os itens do pedido. Após dar OK, irá aparecer a tela do pedido com os seus itens e o valor total.

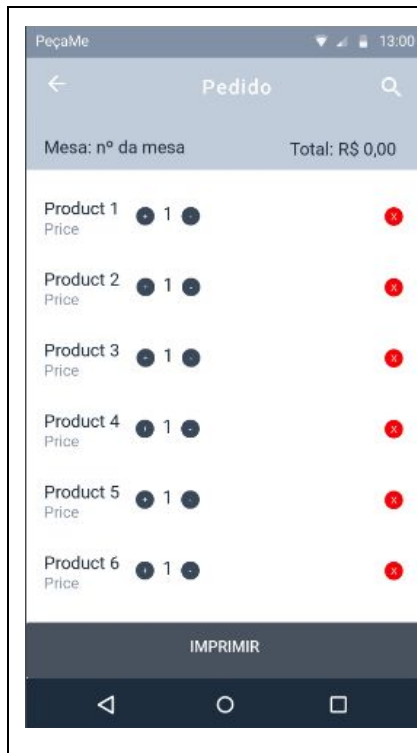
Tela 4

	<p>Essa tela aparece os produtos de acordo com a categoria. Ao selecionar a categoria desejada nos tabs, os produtos são modificados, aparecendo sua descrição e seu valor.</p> <p>Ao clicar em um produto, irá para a tela de detalhes do mesmo.</p>
--	---

Tela 5

	<p>Na tela de detalhes do produto, é possível verificar em qual comanda esse produto vai pertencer, poderá ser adicionado ou removido componente do produto, ou seja, seus adicionais.</p> <p>Ao clicar no botão “ + ”, esse produto vai ser adicionado ao pedido. Ao clicar no botão confirma, vai confirmar o pedido.</p>
---	---

Tela 6



Após confirmar o pedido, é exibido todos os itens do pedido, com suas respectivas descrições, valor de cada item, número da mesa e valor total. É possível aumentar o diminuir a quantidade de cada item, como também de cancelá-lo. Essa tela aparece em duas partes no app. Após confirmar o pedido e quando seleciona a opção de mesas na tela inicial.

No fluxo normal, após confirmar pedido, o botão de imprimir fica visível, para poder enviar o pedido para o setor de produção. Ao clicar em mesas na tela inicial, o botão fica invisível.

Tela 7



Essa tela demonstra como seria o Widget do app. Ele mostraria as mesas que estão disponíveis e as ocupadas.

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

A persistência de dados vai ser com SQLite onde os dados da aplicação, como por exemplo, a comanda da mesa com os pedidos, possam ser retirados no setor de produção.

Será usado um ContentProvider para fazer um intermédio entre o app e o banco de dados, garantindo assim um nível extra de abstração de dados.

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

Ao criar um pedido, o atendente vai até a mesa do cliente e insere no app o número da comanda referente à mesa. Ele consegue inserir e/ou remover os produtos do cliente e tem a opção de imprimir. Quando se clica em imprimir, esse pedido vai para o setor de produção de maneira remota através do Firebase e dentro do setor ele é recuperado e exibido em um computador/tablet ou impresso em papel.

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

ButterKnife para usar annotations e abstrair os binds das views, garantindo que não tenha boilerplate no desenvolvimento;

Firebase Realtime Database para armazenar e sincronizar os dados que serão enviado para o setor de produção.

Google Analytics para poder ter um parâmetro melhor sobre comportamentos dos clientes e ajudar na tomada de decisões.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

Será construído as interfaces de UI e tudo relacionado a ela, como os xmls dos values. Será necessário adicionar a biblioteca design, para ser compatível com o material design.

Subtarefas. Por exemplo:

- Construir a UI da MainActivity
- Construir a UI da ChooseTableActivity
- Construir a UI da ListProductActivity
- Construir a UI da DetailProductActivity
- Construir a UI da OrderActivity
- Construir a UI da caixas de diálogos de confirmação

Tarefa 2: Implementar as bibliotecas

Adicionar as bibliotecas de ButterKnife

Tarefa 3: Criar o bando de dados SQLite

Criar o banco de dados para ter armazenado nele os produtos do app, que vão ser inseridos de forma estática, diretamente no banco de dados e só carregado dentro do aplicativo.

Subtarefas.

- Criar o contrato
- Criar o banco de dados
- Criar o CRUD

Tarefa 4: Criar o Content Provider

Criar o Content Provider para conectar os dados do app com o banco de dados.

Subtarefas:

- Adicionar a permissão do Content Provider
- Adicionar o Content Resolver

Tarefa 5: Criar o Provider

Criar o Provider para acessar o Content Provider

Subtarefas:

- Criar uma classe provider
- Registrar o provider no Manifest
- Definir URIs para identificar os TaskContentProvider
- Atualizar o contract do bando de dados
- Construir a URIMatcher
- Implementar o CRUD no TaskContentProvider

Tarefa 6: Criar o SharedPreferences

Criar o SharedPreferences para poder armazenar status da mesa, por exemplo, aberta ou fechada, número da mesa.

Subtarefas:

- Implementar as lógicas de negócio
- Testar manualmente

Tarefa 7: Criar o Widget

Criar o widget para demonstrar as mesas disponíveis e ocupadas. Essas mesas poderão ser identificadas por cor, sendo verde para disponíveis e vermelhas para ocupadas.

Subtarefas:

- Criar a interface
- Obter o status de cada mesa
- Testar manualmente

Tarefa 8: Criar o serviço Firebase

Criar o serviço Firebase para poder armazenar e recuperar as informações no setor de produção.

Subtarefas:

- Implementar o Realtime Database
- Implementar a ferramenta Analytics

Instruções para Envio

4. Assim que completar todas as partes, faça o download deste documento como um PDF [File → Download as PDF]
5. Crie um repositório novo no Github para o capstone. Dê o nome “**Capstone Project**”
6. Adicione este documento no seu repositório. Tenha certeza que está nomeado como “**Capstone_Stage1.pdf**”