

Nome: Gustavo Vergara Garcia

RGM: 25061101

Documentação

Revisite a documentação do projeto desenvolvido na PIT I, peça *feedback* aos seus colegas e veja possíveis melhorias. Insira, a seguir, toda a documentação atualizada e melhorada:

Histórias:

| |
|---|
| ID: 1 |
| Título: Cardápio da loja |
| Requerente: Cliente |
| Ação: Como cliente, quero saber o que é vendido na loja |
| Critérios de aceitação: Deve carregar em menos de 1 segundo quando usando uma conexão 4G. Deve exibir todos os produtos disponíveis da loja selecionada. Produtos devem ter imagem, título e preço. Título e preço dos produtos devem ser legíveis. |
| Regras de negócio: Todos os itens devem ser carregados a partir de um único chamado ao backend. |
| Prioridade: Desenvolver o layout da célula de item do cardápio. Desenvolver o layout da tabela que lista o cardápio. Chamar o backend para pegar o cardápio. |
| Pontos de história: 13 |

| |
|---|
| ID: 2 |
| Título: Busca de item do cardápio |
| Requerente: Cliente decidido |
| Ação: Como cliente decidido, quero procurar um item específico no cardápio |
| Critérios de aceitação: <ul style="list-style-type: none">• Termo de busca deve ser comparado com o título, descrição e opções de personalização. |
| Regras de negócio: <ul style="list-style-type: none">• Busca deve ser feita localmente, isto é, sem chamar o backend. |
| Requisito não funcional: <ul style="list-style-type: none">• Cardápio deve ser atualizado assim que o termo de busca mudar. |

| |
|--|
| Prioridade: |
| <ul style="list-style-type: none"> • Adicionar barra de busca na tela do cardápio. • Implementar algoritmo de busca em cima dos itens do cardápio recebidos do backend. • Atualizar lista com os resultados da busca à medida que a busca é digitada. |
| Pontos de História: 3 |

| |
|---|
| ID: 3 |
| Título: Seleção de loja |
| Requerente: Viajante |
| Ação: Como viajante, quero poder fazer pedido em lojas de lugares diferentes |
| Critérios de aceitação: <ul style="list-style-type: none"> • Lista todas as lojas atualmente abertas. |
| Regras de negócio: <ul style="list-style-type: none"> • Lojas mais próximas devem aparecer primeiro na lista. |
| Requisito não funcional: <ul style="list-style-type: none"> • Resposta do backend deve ser guardada localmente durante toda uma sessão. |
| Prioridade: <ul style="list-style-type: none"> • Adicionar botão no topo da tela de cardápio • Desenvolver layout da modal de seleção de loja • Chamar o backend para pegar o cardápio da loja selecionada |
| Pontos de história: 3 |

| |
|--|
| ID: 4 |
| Título: Personalização de item |
| Requerente: Cliente |
| Ação: Como cliente, quero personalizar a minha bebida para o meu gosto |
| Critérios de aceitação: <ul style="list-style-type: none"> • Todos os itens de personalização recebidos do backend devem ser exibidos. • Preço do item deve ser recalculado de acordo com os itens selecionados. • Deve carregar itens em menos de 1 segundo. |
| Regras de negócio: <ul style="list-style-type: none"> • Deve salvar as personalizações ao adicionar ao carrinho. • Todos os itens devem ser carregados a partir de um único chamado ao backend. |
| Requisito não funcional: <ul style="list-style-type: none"> • Preço do produto deve ser calculado de acordo com as personalizações selecionadas. |

| |
|---|
| Prioridade: |
| <ul style="list-style-type: none">• Desenvolver layout da tela de layout do produto• Desenvolver layout das células de personalização• Chamar o backend para pegar os detalhes do produto e seus itens de personalização• Calcular total do item com base nos itens selecionados |
| Pontos de história: 8 |

| |
|---|
| ID: 5 |
| Título: Revisão de pedido |
| Requerente: Grupo de clientes |
| Ação: Como um grupo de clientes, queremos poder revisar todas os itens do pedido antes de terminar a compra |
| Critérios de aceitação: <ul style="list-style-type: none">• Todos os itens adicionados devem aparecer no carrinho.• Ao diminuir a quantidade de um item para zero ele deve ser removido da lista.• Total deve ser atualizado de acordo com os itens e suas quantidades. |
| Regras de negócio: <ul style="list-style-type: none">• Carrinho deve ser mantido localmente |
| Requisito não funcional: <ul style="list-style-type: none">• Carregamento do carrinho deve ser imediato |
| Prioridade: <ul style="list-style-type: none">• Desenvolver layout do carrinho• Desenvolver alteração de quantidade de itens do carrinho• Chamar o backend para pegar o tempo estimado de entrega para esse carrinho |
| Pontos de história: 5 |

| |
|--|
| ID: 6 |
| Título: Realizar pedido |
| Requerente: Cliente regular |
| Ação: Como cliente regular, quero fazer meu pedido pelo app para evitar filas |
| Critérios de aceitação: <ul style="list-style-type: none">• Nome deve vir preenchido com o nome utilizado no último pedido.• Deve mostrar um indicador de carregamento ao clicar no botão de realizar pedido. |
| Regras de negócio: <ul style="list-style-type: none">• Botão de realizar pedido só deve aceitar um toque para evitar pedidos duplicados. |
| Requisito não funcional: <ul style="list-style-type: none">• Chamada ao backend não deve demorar mais do que 2 segundos. |

Prioridade:

- Desenvolver layout do alerta perguntando o nome
- Preencher campo do nome com o nome usado no último pedido
- Chamar o backend com os itens do carrinho e nome do cliente para fazer o pedido

Pontos de história: 3

ID: 7

Título: Previsão de entrega de um carrinho

Requerente: Cliente a caminho

Ação: Como cliente a caminho, quero saber em quanto tempo tenho até meu pedido ficar pronto

Critérios de aceitação:

- Deve ser exibido imediatamente após confirmação pelo backend de que o pedido foi feito.
- Tempo deve estar formatado de maneira legível por um humano.

Regras de negócio:

- Usar previsão recebida do backend ao fazer o chamado de realizar pedido.

Requisito não funcional:

- Não deve ser feito um novo chamado ao backend.

Prioridade:

- Desenvolver layout de pedido feito com sucesso
- Exibir tempo de entrega com base no retorno do backend ao fazer um pedido

Pontos de história: 1

ID: 12

Título: Previsão de entrega de um pedido feito

Requerente: Cliente com pressa

Ação: Como cliente com pressa, quero saber quanto tempo falta para receber meu pedido

Critérios de aceitação:

- Lista todos os pedidos em andamento.

Regras de negócio:

- Previsão de entrega deve ser atualizada automaticamente.

Requisito não funcional:

- Tirando o primeiro acesso, deve mostrar pedidos feitos imediatamente.

Prioridade:

- Desenvolver layout da célula de pedido em andamento.
- Desenvolver layout de tela de pedidos.
- Chamar backend para pegar lista de pedidos.

Pontos de história: 2

| |
|---|
| ID: 13 |
| Título: Pedidos anteriores |
| Requerente: Cliente curioso |
| Ação: Como curioso, quero poder ver quais pedidos eu fiz no passado |
| Critérios de aceitação: <ul style="list-style-type: none"> • Pedidos concluídos são exibidos na mesma lista que pedidos em andamento. • Pedidos concluídos são visualmente distintos de pedidos em andamento. |
| Regras de negócio: <ul style="list-style-type: none"> • Guardar lista de pedidos localmente. • Exibir lista local de pedidos enquanto o chamado para atualizar a lista é feito. |
| Prioridade: <ul style="list-style-type: none"> • Desenvolver layout da célula de pedido concluído. • Inserir célula de pedido concluído na tela de pedidos. |
| Pontos de história: 5 |

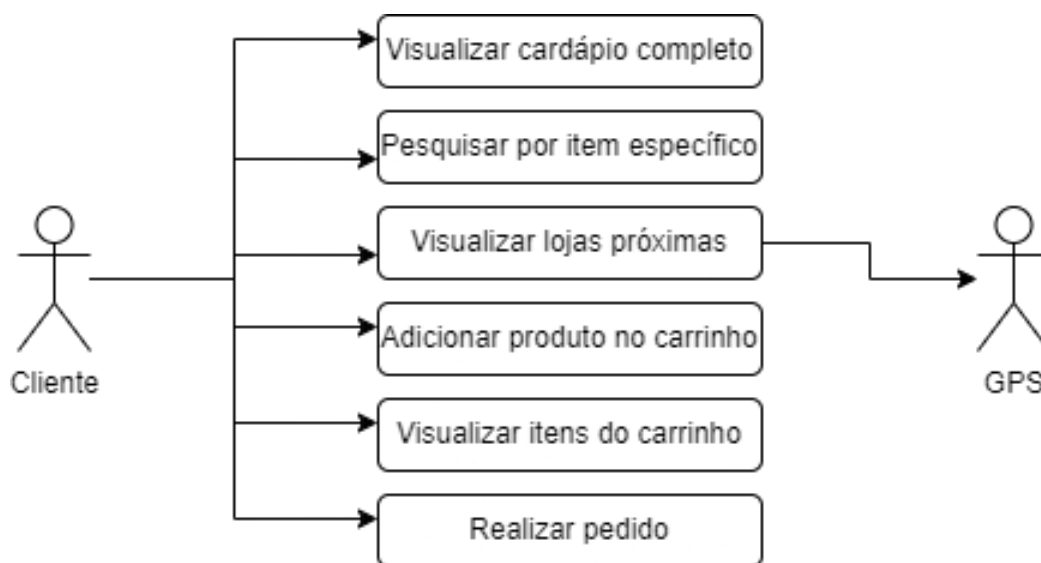
| |
|--|
| ID: 14 |
| Título: Informações nutricionais |
| Requerente: Cliente consciente |
| Ação: Como cliente consciente, quero saber informações nutricionais e alergênicos nas bebidas |
| Critérios de aceitação: <ul style="list-style-type: none"> • Informações devem estar juntas a descrição do produto. |
| Regras de negócio: <ul style="list-style-type: none"> • Informações devem vir do backend. |
| Requisito não funcional: <ul style="list-style-type: none"> • Deve ser carregado ao mesmo tempo que outras informações do produto. |
| Prioridade: <ul style="list-style-type: none"> • Adicionar texto de descrição de produto a tela de detalhe de produto. • Preencher descrição com a informação recebida do backend. |
| Pontos de história: 1 |

Backlog

| ID | História do usuário | Estimativa em pontos | Prioridade |
|----|------------------------|----------------------|------------|
| 1 | Cardápio da loja | 13 | 1 |
| 4 | Personalização de item | 8 | 2 |
| 6 | Realizar pedido | 3 | 3 |
| 5 | Revisão de pedido | 5 | 4 |

| | | | |
|----|--|---|----|
| 3 | Seleção de loja | 3 | 5 |
| 2 | Busca de item do cardápio | 3 | 6 |
| 14 | Informações nutricionais | 1 | 7 |
| 9 | Avaliação do aplicativo | 2 | 8 |
| 7 | Previsão de entrega de um carrinho | 1 | 9 |
| 12 | Previsão de entrega de um pedido feito | 2 | 10 |
| 13 | Pedidos anteriores | 5 | 14 |

Casos de Uso



Nome: Visualizar cardápio completo

Descrição: O cliente deseja visualizar o cardápio completo da loja, com todos os produtos e opções disponíveis.

Fluxo Principal:

1. O cliente abre o aplicativo.
2. O cliente navega até a seção "Cardápio".
3. O sistema exibe o cardápio completo da loja selecionada.
 - a. <Curso Alternativo Alfa>
4. O cliente pode navegar e visualizar as descrições e preços de cada item do cardápio.
5. O cliente pode selecionar um item e visualizar os detalhes ou adicionar ao carrinho de compras.

Fluxo Alternativo Alfa

- a3. O cliente não selecionou uma loja
- a4. O sistema seleciona a loja mais próxima do cliente
- a5. O caso de uso continua e, é desviado para a linha 4 do Fluxo Principal deste mesmo caso de uso.

Nome: Pesquisar por item específico

Descrição: O cliente deseja encontrar rapidamente um item específico no cardápio da loja.

Fluxo Principal:

1. O cliente abre o aplicativo.
2. O cliente navega até a seção "Cardápio".
3. O cliente utiliza a função de pesquisa dentro do cardápio.
4. O sistema exibe os resultados correspondentes à pesquisa realizada.
5. O cliente pode selecionar um item dos resultados e visualizar os detalhes ou adicionar ao carrinho de compras.

Nome: Visualizar lojas próximas

Descrição: O cliente deseja encontrar lojas próximas à sua localização atual.

Pré-condições:

- Cliente deu acesso a sua localização

Fluxo Principal:

1. O cliente abre o aplicativo.
2. O cliente navega até a seção "Cardápio".
3. O cliente toca no nome da loja.
4. O sistema utiliza a localização do dispositivo do viajante para identificar as lojas próximas.
5. O sistema exibe as lojas encontradas em uma lista, com nome e distância.
6. O cliente pode selecionar uma loja específica para visualizar o seu cardápio.

Nome: Adicionar produto no carrinho

Descrição: O cliente deseja pedir um produto.

Fluxo Principal:

1. O cliente abre o aplicativo.
2. O cliente navega até a seção "Cardápio".
3. O cliente seleciona a bebida desejada.
4. O sistema exibe a tela "Detalhe de Produto" com os ingredientes padrão da bebida.
5. O cliente seleciona o tamanho desejado.
6. O cliente seleciona ingredientes extras desejados para adicionar à bebida.
7. O sistema atualiza a personalização da bebida de acordo com os ingredientes selecionados.
8. O cliente toca em "Adicionar".

Nome: Visualizar itens do carrinho

Descrição: O cliente deseja visualizar todos os itens adicionados ao carrinho de compras antes de finalizar a compra.

Pré-condições:

- O cliente adicionou um produto ao carrinho.

Fluxo Principal:

1. O cliente é redirecionado até a tela "Carrinho".
2. O sistema exibe uma lista de todos os itens adicionados ao carrinho.
3. O cliente pode visualizar os detalhes de cada item, incluindo quantidade, preço unitário e opções de personalização.
4. O cliente pode fazer ajustes no carrinho, como remover itens ou alterar quantidades.
5. O cliente pode voltar para a seção de cardápio para adicionar mais itens ao carrinho, se desejar.
6. O cliente pode tocar em "Pedir" para fazer um pedido.

Nome: Realizar pedido

Descrição: O cliente regular deseja revisar os itens do carrinho e confirmar o pedido antes de finalizá-lo.

Pré-condições:

- O cliente já adicionou produtos ao carrinho.

Fluxo Principal:

1. O cliente abre o aplicativo.
2. O cliente acessa o carrinho pronto.
3. O sistema exibe os itens do carrinho.
4. O cliente pode fazer ajustes no carrinho, se desejar.
5. O cliente toca em "Pedir".
6. O sistema pergunta o nome do cliente.
7. O cliente informa seu nome.
8. O sistema exibe a confirmação do pedido feito com a estimativa de entrega.

Diagrama de Classes

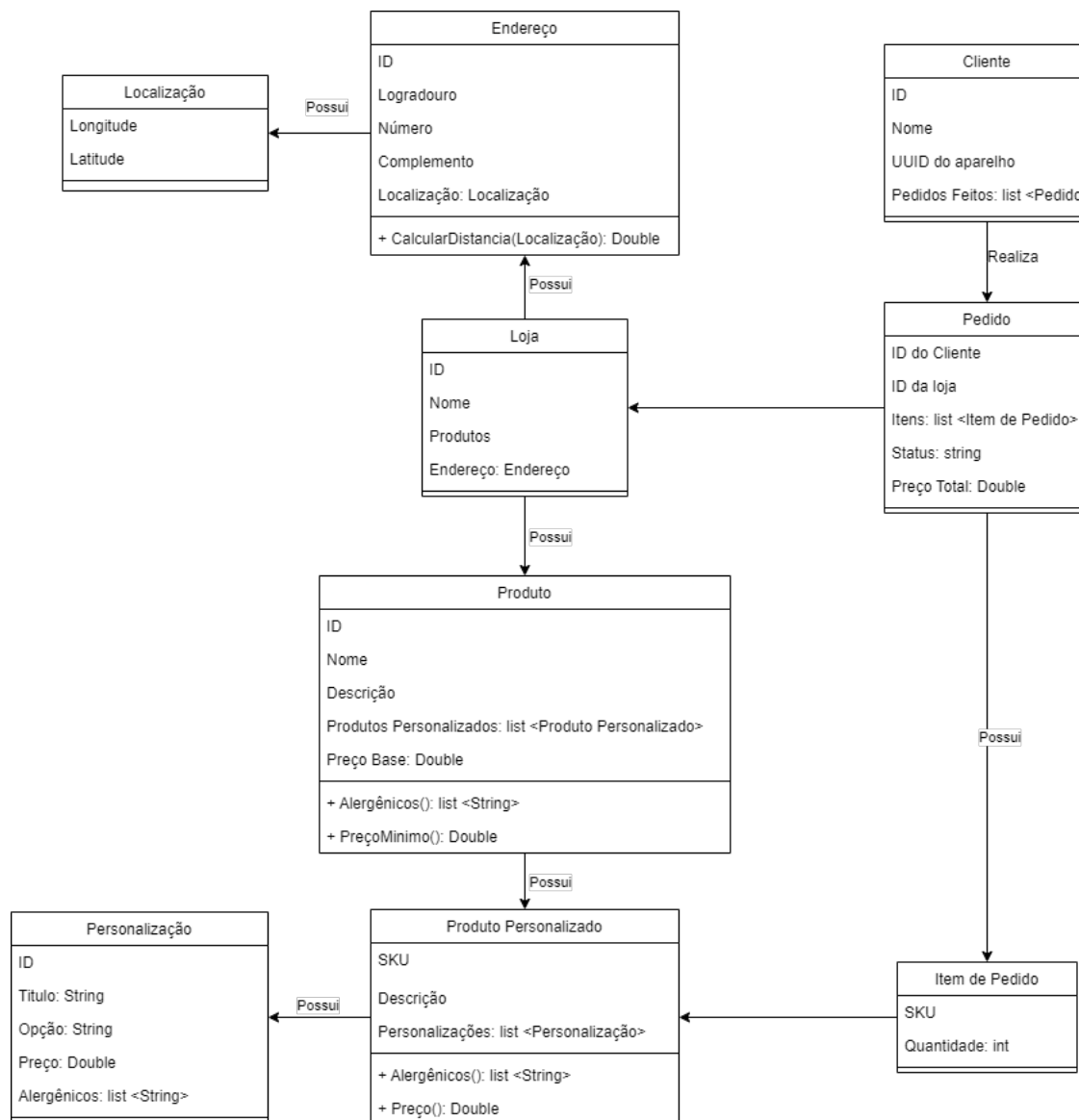
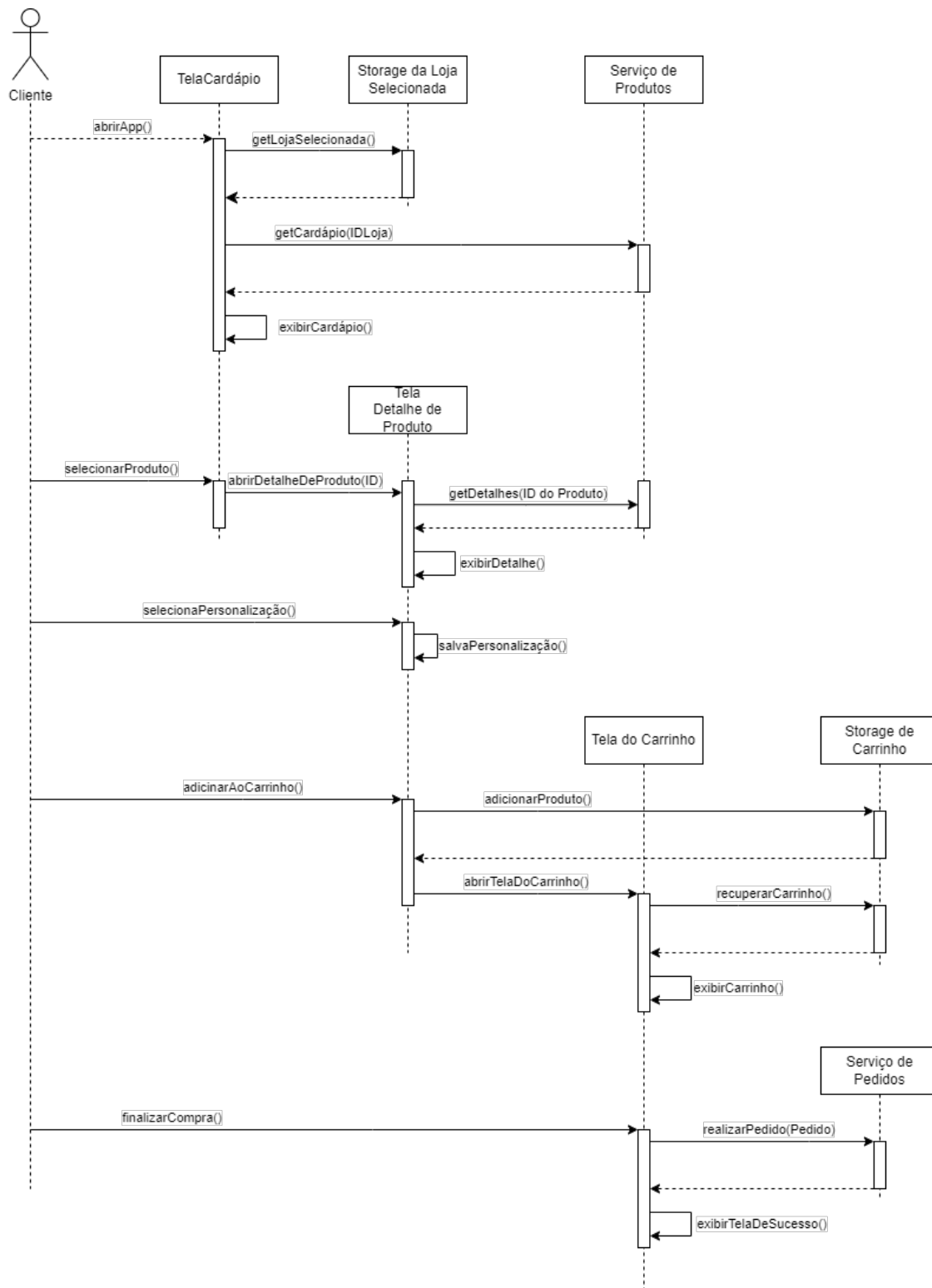


Diagrama de Sequência



Wireframes

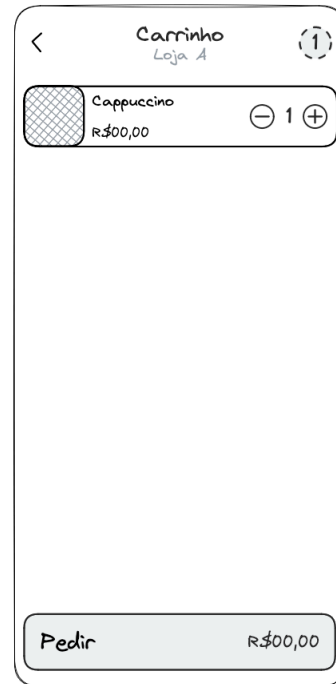
Cardápio



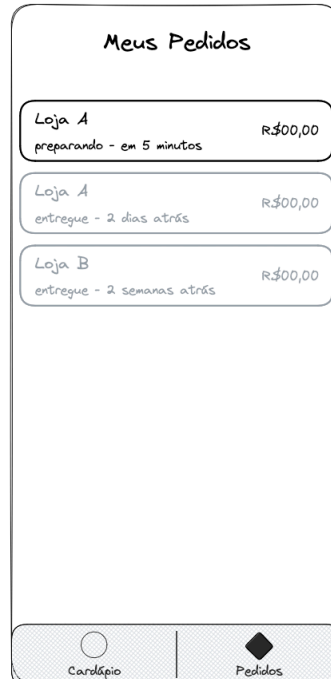
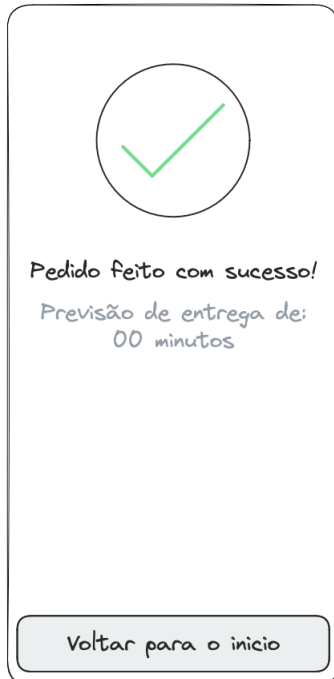
Detalhe de produto



Carrinho



Pedido feito com sucesso Lista de pedidos



Codificação:

Na Tabela a seguir insira as informações referentes ao desenvolvimento do código do processo.

| | |
|---|---|
| Linguagem | Swift e ES6 |
| Banco de Dados | SQLite, UserDefaults e o FileSystem |
| Hospedagem | Nenhuma |
| Plataforma | iOS |
| Modo de Codificação | (x) Tradicional () <i>Low-code</i> |
| Link do repositório no GitHub com os códigos abertos | https://github.com/GustavoVergara/CoffeeShopApp |
| Link da solução em funcionamento | Instruções para execução da solução podem ser encontradas no README.md no repositório no GitHub. |
| Link do vídeo narrado (no mínimo 5 min) | https://www.youtube.com/watch?v=6fnxKhTOhKA |

Testes da Solução

Escolha 5 colegas para testar sua aplicação, disponibilize o *link* de acesso ou os recursos necessários para que testem como usuários. Preencha a Tabela a seguir com as informações obtidas:

| | |
|---|----------------------------------|
| Nome: Lua Almeida | Data do teste: 01/12/2023 |
| O que testou e funcionou: Busca, customização de produto, finalização de pedido. | |
| O que testou e não funcionou – O que deve ser corrigido: Estranhou de não conseguir ver os pedidos feitos. | |
| Funcionalidade não testada (faltou ou não foi implementada): Tela da lista de Pedidos feitos que foi implementada com base nesse feedback. | |

| | |
|---|----------------------------------|
| Nome: Guido Sarmiento | Data do teste: 03/12/2023 |
| O que testou e funcionou: Customização de produto, adição ao carrinho, finalização de pedido e lista de pedidos feitos. | |
| O que testou e não funcionou – O que deve ser corrigido: Acessou o carrinho antes de adicionar algum produto, causando um estado estranho no aplicativo. | |
| Funcionalidade não testada (faltou ou não foi implementada): Novo acesso ao carrinho, implementado com base no feedback. | |

| | |
|---|----------------------------------|
| Nome: Bruno Guimarães | Data do teste: 03/12/2023 |
| O que testou e funcionou: Busca, Customização de produto e adição ao carrinho. | |
| O que testou e não funcionou – O que deve ser corrigido: Nada | |
| Funcionalidade não testada (faltou ou não foi implementada): Novo acesso ao carrinho, implementado com base no feedback. | |

| | |
|---|----------------------------------|
| Nome: Matheus Zicari | Data do teste: 03/12/2023 |
| O que testou e funcionou: Seleção e customização de produto, lista de pedidos. | |
| O que testou e não funcionou – O que deve ser corrigido: Animação na tela de pedido concluído deu. | |
| Funcionalidade não testada (faltou ou não foi implementada): Novo acesso ao carrinho. | |

| | |
|---|----------------------------------|
| Nome: Ana Paula Vergara | Data do teste: 04/12/2023 |
| O que testou e funcionou: Seleção e customização de produtos, adição ao carrinho de múltiplos produtos e finalização de compra | |
| O que testou e não funcionou – O que deve ser corrigido: Não encontrou nenhum problema | |
| Funcionalidade não testada (faltou ou não foi implementada): | |

Vídeo da Solução atualizada

Após levantar os *feedbacks* e executar as correções necessárias e pertinentes, grave um vídeo de **até 5 minutos** apresentando as modificações realizadas no sistema.

| | |
|--------------------------|---|
| Link para o vídeo | https://www.youtube.com/watch?v=dJOeXFgrHsg |
|--------------------------|---|