

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: mSchneider86

FitTracker

Descrição

O aplicativo FitTracker proporciona àqueles que buscam uma melhora em sua saúde um acompanhamento de seu progresso, pois como diria o pai da administração moderna Peter Drucker “Se você não pode medir, não pode gerenciar.” Sendo assim, o FitTracker permite que seus usuários monitorem a evolução de seus pesos, percentuais de gordura e frequência cardíaca, e que visualizem seus progressos através de gráficos para que permaneçam motivados e atinjam seus objetivos.

Público-Alvo/Intended User

Usuários de ambos os sexos que estejam em busca de melhorarem sua saúde e condicionamento físico, ou apenas queiram monitorar seu progresso nestas áreas.

Funcionalidades/Features

- Cadastro do perfil do usuário
- Cadastro de peso
- Lista dos pesos cadastrados
- Emissão de gráfico com os pesos cadastrados
- Medição do percentual de gordura através de diferentes métodos
- Cadastro do percentual de gordura
- Lista dos percentuais de gordura cadastrados
- Emissão de gráfico com os percentuais de gordura cadastrados
- Aferição da frequência cardíaca com auxílio do flash da câmera
- Lista das frequência cardíaca cadastrados
- Emissão de gráfico com as frequências cardíacas cadastrados

Protótipo de Interfaces do Usuário

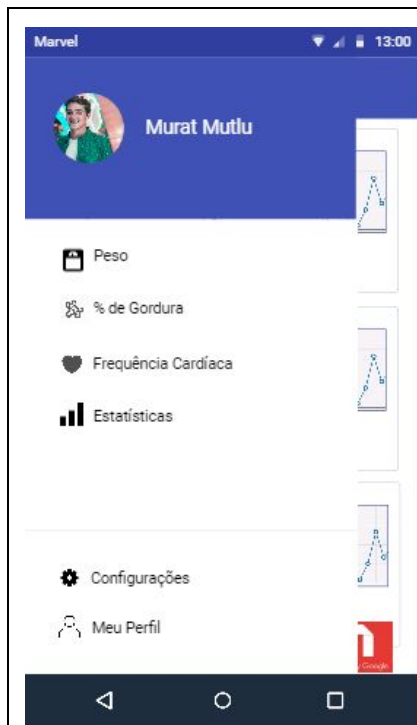
Elas podem ser feitas a mão (tire uma foto dos seus desenhos e os insira neste fluxo), ou usando um programa como o Photoshop ou Balsamiq.

Tela 1



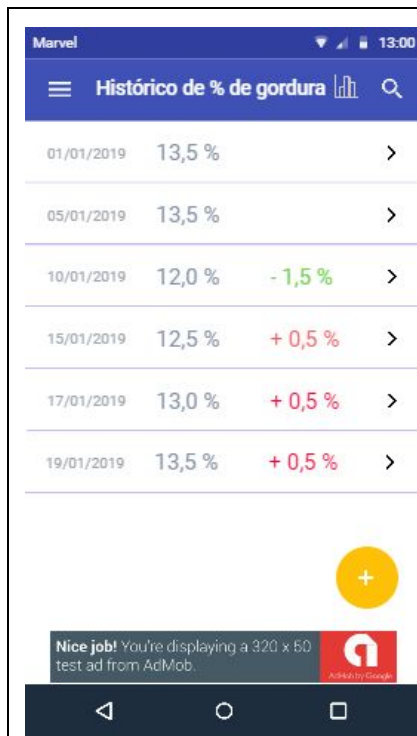
Tela Principal do aplicativo, onde o usuário pode visualizar rapidamente um breve resumo do seu estado atual em cada uma das categorias (controle de Peso, percentual de gordura e frequência cardíaca) além de pré-visualizar um gráfico que mostra o histórico destas medições. Uma vez selecionado um dos cards, ele é direcionado para a tela com a lista destas medições, onde pode obter mais detalhes das mesmas.

Tela 2



Menu principal da aplicação, que dá acesso às telas de controle de peso, percentual de gordura, frequência cardíaca, estatísticas de progresso, além das configurações do aplicativo (que incluem as unidades de medida que serão usadas na aplicação: sistema métrico ou imperial), e as configurações de perfil, onde o usuário informa seus dados, como nome, altura, idade, foto.

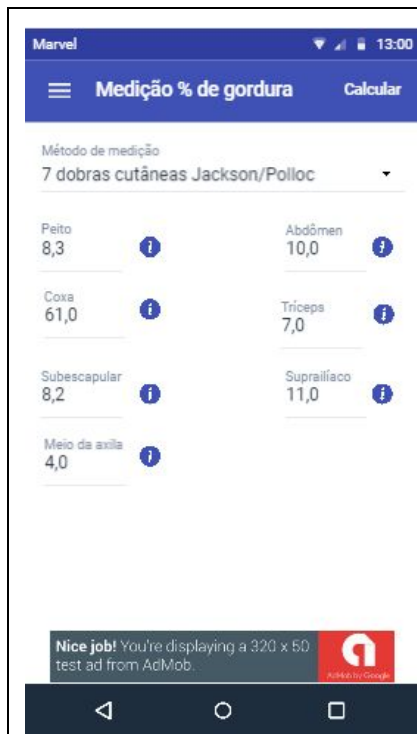
Tela 3



Lista que detalha as medições de percentual de gordura salvas pelo usuário, exibindo além do valor registrado, a data da medição e a diferença para a medição anterior, onde este pode rapidamente verificar se houve um ganho ou perda desde sua última aferição.

Além disso a tela conta com um botão para que uma nova medição seja feita, e um botão que exibe graficamente o histórico de medições.

Tela 4



Tela onde o usuário realiza uma nova medição de seu percentual de gordura. O primeiro passo é selecionar no spinner qual será o método de medição utilizado, onde estarão disponíveis as seguintes opções:

-Informar manualmente (O usuário apenas insere um valor que foi aferido externamente);

- Fita métrica/marinha americana;
- 3 dobras cutâneas Jackson/Polloc;
- 4 dobras cutâneas Jackson/Polloc;
- 7 dobras cutâneas Jackson/Polloc;
- 9 dobras cutâneas Parrillo.

Onde exceto na medição manual, são apresentados os campos necessários para cada método e o usuário deve informar as medições para que o aplicativo realize o cálculo. Ao lado de cada medida a ser informada há um ícone de ajuda onde o local e como a medida deve ser feita é exibida

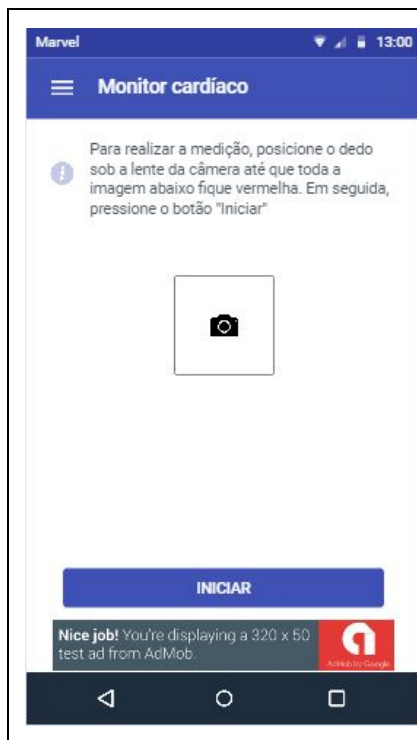
Tela 5



Lista que detalha as medições de peso salvas pelo usuário, exibindo além do valor registrado, a data da medição e a diferença para a medição anterior, onde este pode rapidamente verificar se houve um ganho ou perda desde sua última aferição.

Além disso a tela conta com um botão para que uma nova medição seja feita, e um botão que exibe graficamente o histórico de medições.

Tela 6




Tela onde o usuário pode realizar sua medição cardíaca. A tela fornece ao usuário informações de como proceder e posicionar seu dedo sobre a câmera e ao pressionar o botão iniciar a câmera e o flash são iniciados e a imagem de câmera na tela é substituída pela imagem da câmera para que o usuário se oriente e cubra totalmente a lente, o que então inicia a medição.

Tela 7

	<p>Uma vez iniciada a medição o usuário observa o progresso da mesma através do círculo amarelo que ao concluir, informa o resultado de sua frequência cardíaca e permite que o mesmo salve a aferição realizada..</p>
--	--

Tela 8

	<p>O Widget do aplicativo permite que o usuário rapidamente verifique em sua home screen seu estado atual de progresso, visualizando as últimas medições de gordura, peso e frequências cardíacas, visando manter o mesmo motivado e em busca de seus objetivos.</p>
---	--

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Os dados do aplicativo serão armazenados localmente com Room para serem exibidos quando o usuário não possuir conexão, e salvos no Firebase Firestore.

Além disso, outros Architecture Components serão utilizados para compor uma arquitetura mais robusta e mais facilmente testável, da seguinte forma:

ViewModel: Serão utilizadas ViewModels para realizar o encapsulamento dos dados para um controller de UI e assim fazer com que estes dados sobrevivam a mudanças de orientação.

LiveData: Para facilitar o processo de manter a sincronia dos dados com as telas, uma vez que se trata de uma classe *observable*, que irá atualizar apenas observadores que estiverem ativos em seus ciclos de vida.

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

O usuário irá acessar diretamente os fragmentos com as funcionalidades do aplicativo através da Navigation Drawer, podendo também acessá-las a partir do dashboard da tela inicial, que exibe um resumo dos dados armazenados.

O aplicativo irá conter ainda animações de views e elementos de transição de tela para enriquecer a experiência do usuário. Além disso, o uso de um widget com um resumo da situação atual do usuário, visa melhorar ainda mais tal experiência.

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

- Picasso - para tratar o carregamento e cache das imagens.
- Firebase Database - para armazenar os dados do usuário
- Admob - para exibir anúncios para o usuário
- Butterknife - para reduzir a quantidade de código para inicialização das views
- Android Support Library - para fazer uso de recursos como RecyclerView, CardView, Snackbar, FloatingActionButton, entre outros.
- Android Architecture Components - para a construção de uma arquitetura mais robusta

Descreva como você implementará o Google Play Services.

O aplicativo usará os seguintes Services:

- Google Mobile Ads: Banners de propaganda serão exibidos na parte inferior das telas
- Firebase Analytics: O analytics será inicializado e usado para armazenar e acompanhar as ações do usuário dentro do aplicativo.

Próximos Passos: Tarefas Necessárias

Esta é a parte onde você falará sobre as principais funcionalidades do seu app (mencionadas acima) e as dividirá em tarefas técnicas tangíveis que você pode completar de forma incremental até finalizar o app.

Tarefa 1: Configuração do Projeto/Project Setup

- Criar um novo projeto e adicionar uma Navigation Drawer Activity. Adicionar as dependências em seu respectivo bloco do build.gradle.
- Criar uma navigation drawer
- Garantir que o app utiliza versões estáveis de todas as dependências que serão utilizadas
- Configurar as bibliotecas
- Garantir que o aplicativo será desenvolvido apenas na linguagem Java

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

- Criar cada Activity e Fragment que serão utilizados. Criar layouts para cada uma destes arquivos
- Criar UI para a MainActivity
- Criar UI para Cadastro de Pesos
- Criar UI para Nova Medição de Gordura
- Criar UI para Medição Cardíaca
- Criar UI para Lista de Pesos
- Criar UI para Lista de percentuais de gordura

- Criar UI para Lista de Medições Cardíacas
- Criar UI dos gráficos
- Criar UI do Widget

Tarefa 3: Widget

- Criação de um widget Provider e de um AppWidgetLayout para que o widget que exiba um resumo do progresso atual do usuário:
 - Seu último peso registrado;
 - Seu último percentual de gordura registrado;
 - Sua última frequência cardíaca registrada.

Tarefa 4: Implementar Google Play Services

- Exibir ads de teste na parte inferior de cada tela
- Conectar o app ao Firebase
- Garantir que cada serviço importado no build.gradle está sendo utilizado no aplicativo
- Garantir que o app utiliza apenas uma instância do Firebase Analytics
- Adicionar eventos de log para o Firebase Analytics.

Tarefa 5: Tratamento de Erros

- Exibição de mensagens informativas no lugar dos gráficos quando o usuário ainda não possuir pelo menos 2 marcações salvas em cada categoria.
- Verificação de conectividade de rede. Caso não exista conexão com a internet, notificar o usuário através de SnackBar
- O aplicativo valida todas as informações digitadas pelo usuário, e caso estas não estejam no formato adequado, exibe alerta para o usuário, e em casos inesperados, salva um log de erro e o trata de acordo, visando não interromper o funcionamento normal do aplicativo.

Tarefa 6: Inclusão de suporte de Acessibilidade e Localização

- Inclusão de descrição de todas as ImageViews e ImageButtons usando o atributo ContentDescription
- Realocação de todas as strings fixas no código para o arquivo strings.xml
- Criação de suporte para o idioma inglês através da tradução do arquivo de strings

- Implementação de suporte para layouts RTL
- Adição de navegação usando D-pad

Tarefa 7: Adicionar polimento às interfaces e aplicação do Material Design

- Alteração das cores e fontes
- Adição dos padrões do Material Design para a criação de um app mais agradável visualmente
- Adicionar transições simples entre as activities
- Garantir que o tema do app estenda de AppCompatActivity

Tarefa 8: Adicionar Configuração de Assinatura

- Criação de uma key e uma keystore
- Criação de uma configuração de assinatura no build.gradle

Tarefa 9: Build do App

- O App compila a través de um checkout do repositório, sem a necessidade de nenhuma configuração adicional
- A compilação e deploy do app é feita a partir da tarefa *installRelease* do Gradle
- Todas as dependências são gerenciadas pelo Gradle

Tarefa 10: Implementação de Testes

- Criação de testes unitários e de UI que visam garantir o bom funcionamento das interfaces, assim como a implementação correta das funcionalidades propostas.

Tarefa 11: Conformidade com as Guidelines

- O App está em conformidade com todas as guidelines contidas na descrição do projeto

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**”