

## 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](#)

[Tela 3](#)

[Tela 4](#)

[Tela 5](#)

[Tela 6](#)

[Tela 7](#)

[Tela 8](#)

## [Considerações Chave/Key Considerations](#)

[Linguagem de programação e versão](#)

[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: Implementação](#)

[Tarefa 4: Validação de dados](#)

## Tarefa 5: Tratar casos de erros

**Usuário do GitHub:** rgcosta

# SocialLinks

## Descrição

SocialLinks permite compartilhar produtos e serviços através de um link único e ser recompensado pelas vendas. Compartilhe produtos de seu interesse com seus amigos e ganhe comissão sob as vendas. Ótimo para compartilhar com seus seguidores nas redes sociais.

## Público-Alvo/Intended User

Usuários do instagram, youtube e outras redes sociais que gostariam de compartilhar produtos/serviços que usam com seus seguidores.

## Funcionalidades/Features

- Permite login do usuário através da conta Facebook ou Google
- Exibe uma lista de produtos por categoria
- Gera um link único de afiliado para divulgação do produto com terceiros
- Simula a compra de um produto
- Monitora quantos produtos foram vendidos a partir do link gerado
- Calcula o valor total de comissão recebido baseado nos produtos divulgados
- Salva os dados de monitoramento/desempenho do link localmente e sincroniza na nuvem

## 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**

	<p>Tela de login onde os usuários podem se cadastrar através das contas de Facebook ou Google. Para autenticação do usuário, será implementado o Firebase Autenticação.</p>
--	---

## Tela 2



Tela onde lista os produtos disponíveis para divulgação. A lista de produtos será obtida a partir de uma API simples localizada em <https://www.placenpepper.com/android/api/{categoria}/lista.json>. Para recuperação dos dados da API, será utilizado a biblioteca Retrofit.

OBS: A json da lista de produtos ainda será criado e disponibilizado no link por mim.

### Tela 3



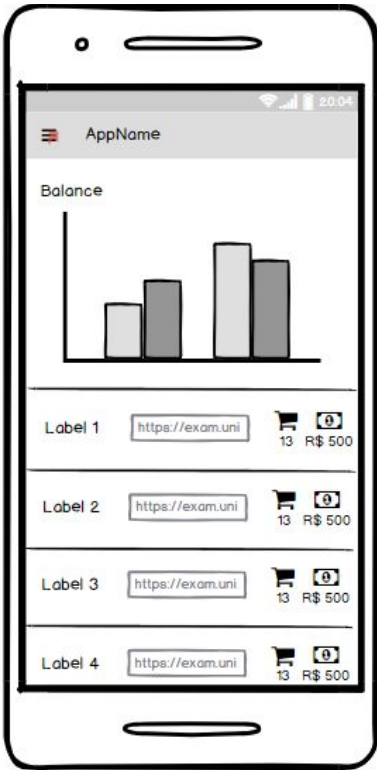
## Tela 4

 <p>The image shows a smartphone screen displaying a product page for 'Whey Protein Standard Gold'. The product is a black tub with a gold label. The label features the 'ON' logo and the text 'GOLD STANDARD WHEY'. Below the product image, the text 'Whey Protein Standard Gold' is displayed. Underneath, there are three icons: a factory icon labeled 'Optimum', a price tag icon labeled 'R\$ 199', and a diamond icon labeled '12%'. To the right of these icons is a circular share icon. Below the icons, the text 'Full description of the product.' is followed by 'Specification XZY' and 'Qtd 999'. At the bottom, under the heading 'Rules', there is a list of four bullet points: 'Available to share on Instagram only', 'Available to share on Youtube only', 'As many as you would like', '12% commission per sale', and 'Respect all the rules of the program'.</p>	<p>Tela de detalhes do produto ao clicar em um dos produtos da Tela 2.</p>
---	--

## Tela 5

	<p>Ação do botão de compartilhar ao ser clicado. O botão flutuante ao ser clicado, gera um link único para o usuário compartilhar o produto nas redes sociais.</p> <p>O link único é implementado através do Firebase Dynamic Links.</p>
--	--

## Tela 6



The mockup shows a mobile app interface. At the top, there's a status bar with a signal icon, a battery icon, and the time 20:04. Below that is a header bar with a hamburger menu icon and the text 'AppName'. The main content area is divided into two sections. The first section, titled 'Balance', contains a bar chart with four bars of increasing height. The second section contains a list of four items, each labeled 'Label 1' through 'Label 4'. Each item has a text input field containing 'https://exam.uni', a shopping cart icon, and a currency icon with the value 'R\$ 500'.

Tela de monitoramento de desempenho dos links divulgados ao clicar no menu Links da Tela 3.

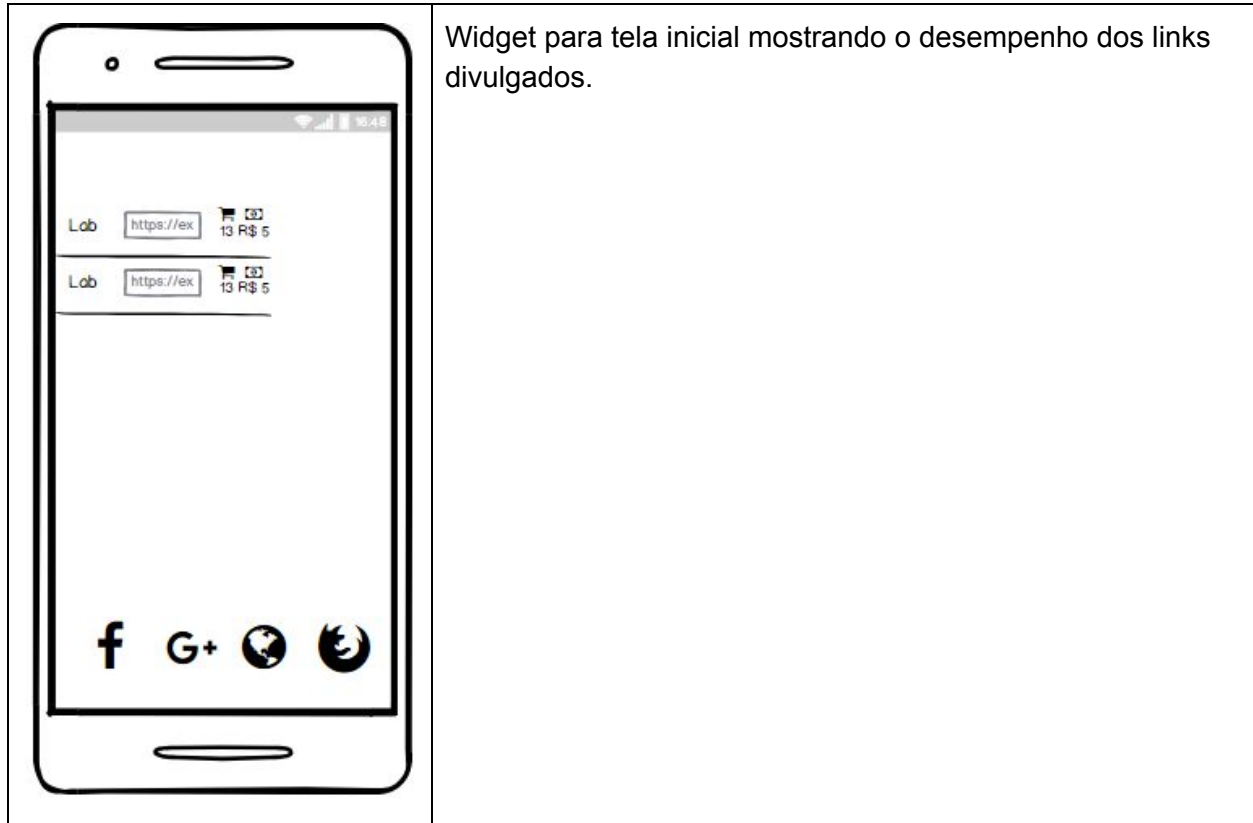
Os dados de desempenho dos links serão armazenados localmente e sincronizados utilizando o Firebase Realtime Database.



## Tela 7

	<p>Quando uma pessoa clicar em um dos links divulgados, seja essa pessoa usuária do app ou não, ela será direcionada (após baixar o app) para essa tela de simulação de compra do produto.</p> <p>Uma vez que o produto é comprado, o usuário afiliado que divulgou o link recebe sua comissão.</p>
--	---

## Tela 8



## Considerações Chave/Key Considerations

### Linguagem de programação e versão

O aplicativo será escrito apenas na linguagem de programação Java e fará uso de bibliotecas de terceiro estáveis bem como Gradle e Android Studio em suas última versão.

### Como seu app vai tratar a persistência de dados?

Os dados serão salvos na web através do Firebase Realtime Database e localmente através do recurso de permanência local dos dados off-line.

A lista de produtos será carregada da API especificada no Tela 2, fazendo uso de uma AsyncTask em pedidos de curta duração. Para tanto, o uso da biblioteca Retrofit será feito para simplificação do processo.

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

Para cada produto que o usuário desejar compartilhar, deve ser selecionado a partir da tela principal e gerado o link único pelo botão flutuante. Se durante o processo o usuário sair do app ou pressionar o botão Home, ele poderá retornar de onde parou sem perder os dados. Ao gerar o link do produto desejado, os mesmos serão salvos localmente e online utilizando o Firebase. Quando o link for clicado, o usuário será direcionado imediatamente para a tela específica do produto podendo simular uma compra. Para que tal ação seja realizada, o Firebase Dynamic Links será utilizado.

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

- Picasso para tratar o carregamento e cache das imagens.
- Retrofit para consumir API de produtos
- Butter Knife para aumento de produtividade
- Firebase Autenticação para controle de acesso de usuários
- Firebase Realtime Database para persistência de dados
- Firebase Dynamic Links para implementar a principal feature do app

### **Descreva como você implementará o Google Play Services.**

Serão implementados os seguintes serviços do Firebase:

- Firebase Dynamic Links para gerar links únicos de divulgação de produtos permitindo que ao link ser clicado o usuário é direcionado direto para a tela específica do produto dentro do aplicativo podendo simular uma compra do mesmo. Caso a pessoa que clicou no link não tenha o app instalado, será direcionado a Play Store para primeiramente baixar o app e então para a tela do produto dentro do app.
- Firebase Realtime Database para sincronizar os dados referente a performance dos links bem como para persistência dos dados localmente.
- Firebase Autenticação para autenticar e cadastrar o usuário no app através dos provedores Facebook e Google.

## 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

- Configurar as bibliotecas a serem utilizados
- Modelar a estruturar os dados
- Criar o projeto no Firebase Console

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

- Construir a UI da MainActivity que conterà a lista de produtos disponíveis
- Construir a UI da ProductDetailActivity que mostrará os detalhes do produto
- Criar a UI de simulação de compra do produto quando clicarem no link
- Criar a UI de monitoramento de desempenho dos links
- Construir a UI de login

### Tarefa 3: Implementação

1. Implementar a tela de login e criar usuários no Firebase Autenticação
2. Implementar a MainActivity recuperando a lista de produto da API
3. Gerar um link único através do Firebase Dynamic Links
4. Implementar a Activity para receber o link dinâmico e mostrar o produto específico
5. Implementar a tela de monitoramento de desempenho dos links bem como salvar os dados usando o Firebase Realtime Database

### Tarefa 4: Validação de dados

- Realizar double check se app está tratando e validando todos os dados corretamente da API e eventuais inputs

## Tarefa 5: Implementar widget

- Implementar o widget da tela inicial conforme ilustração da Tela 8.

## Tarefa 6: Tratar casos de erros

- Identificar e tratar casos de erros isolados.
- Executar todos os testes necessários e garantir o funcionamento do sistema.

## Tarefa 7: Organização e acessibilidade

- Organizar recursos de strings em strings.xml
- Organizar recursos de dimensionamento em dimens.xml
- Disponibilizar recurso de acessibilidade conforme prática padrão e fazendo uso de atributo `contentDescription` para imagens e “sp” para fontes e habilita a alternância de layout RTL

Adicione quantas tarefas que achar necessárias para concluir seu app.

---

### 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**”