

### 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 - Registro de viajante

Tela 2 - Login

Tela 3 - Lista de viagens / Adicionar viagens

Tela 4 - Detalhes da viagem

Tela 5 - Lista de gastos/ Adicionar gastos

Tela 6 - Detalhes de gastos

Tela 7 - Widget

### 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: Construir classes de modelo

Tarefa 4: Construir as activities e fragments das funcionalidades.

Tarefa 5: Construir camada de persistencia de dados

Tarefa 6: Construir Widget.

**Usuário do GitHub:** ManuJs

# Boa Viagem

## Descrição

Esse é um aplicativo para controle de gastos em viagem comerciais e a lazer. Ele ajuda as pessoas a registrarem seus gastos em suas viagens para auxiliar no planejamento de futuras viagens e agilizar o reembolso durante elas.

## Público-Alvo/Intended User


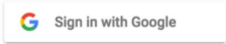
Estudantes em intercâmbio, profissionais que viajam a trabalho e turistas de carteirinha. Viajantes em geral.

## Funcionalidades/Features



- Adicionar e ver suas últimas viagens
- Adicionar e ver gastos por viagem
- Visualizar o total de gastos por viagem
- Widget para facilitar a visualização de gastos da viagem atual.

## Protótipo de Interfaces do Usuário

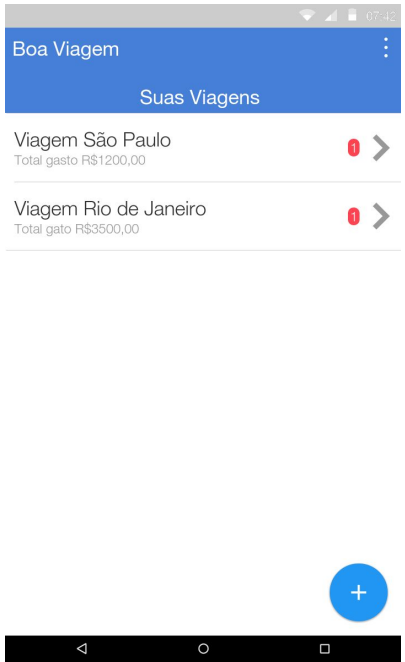
## Tela 1 - Registro de viajante

  <input type="text" value="Login"/> <input type="password" value="Senha"/> <input type="button" value="Registrar"/> <input type="button" value="Já possuo cadastro"/>	<p>O aplicativo permitirá que o viajante se cadastre de duas formas, usando sua conta do google ou pelo cadastro de login e senha. Caso ele já possua cadastro, ele pode ser direcionado direto para tela de login.</p>
---	---

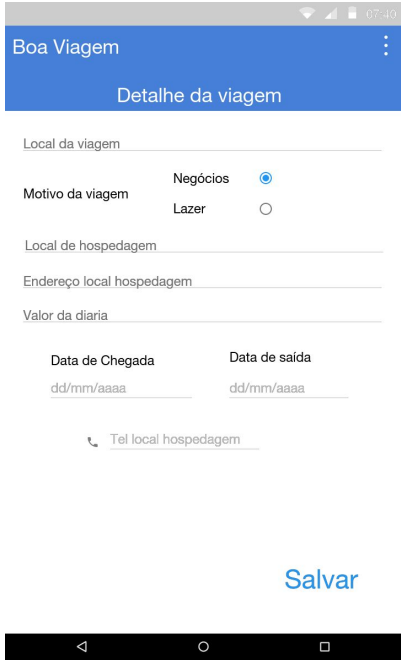
## Tela 2 - Login

  <input type="text" value="Login"/> <input type="password" value="Senha"/> <input type="button" value="Entrar"/> <a href="#">Recupere seu login</a> <input type="button" value="Não possuo cadastro"/>	<p>Tela de login. É necessária a autenticação para que os dados do viajante estejam seguros. É possível que o viajante faça login por dois jeitos: pela conta do Google ou pelo login e senha que ele cadastrou manualmente. Ele será direcionado para a tela principal do app.</p>
---	---


### Tela 3 - Lista de viagens / Adicionar viagens

	<p>Essa é a tela principal pós login. Nela é possível ver as viagens já salvas e seus valores até o momento, além de possibilitar que o viajante adicione novas.</p>
---	--

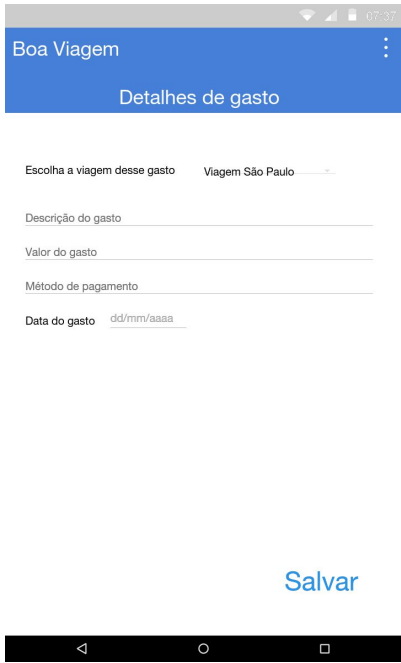
### Tela 4 - Detalhes da viagem

	<p>Quando o usuário deseja adicionar uma nova viagem, é nessa tela que ele informa os dados da viagem.</p>
---	--

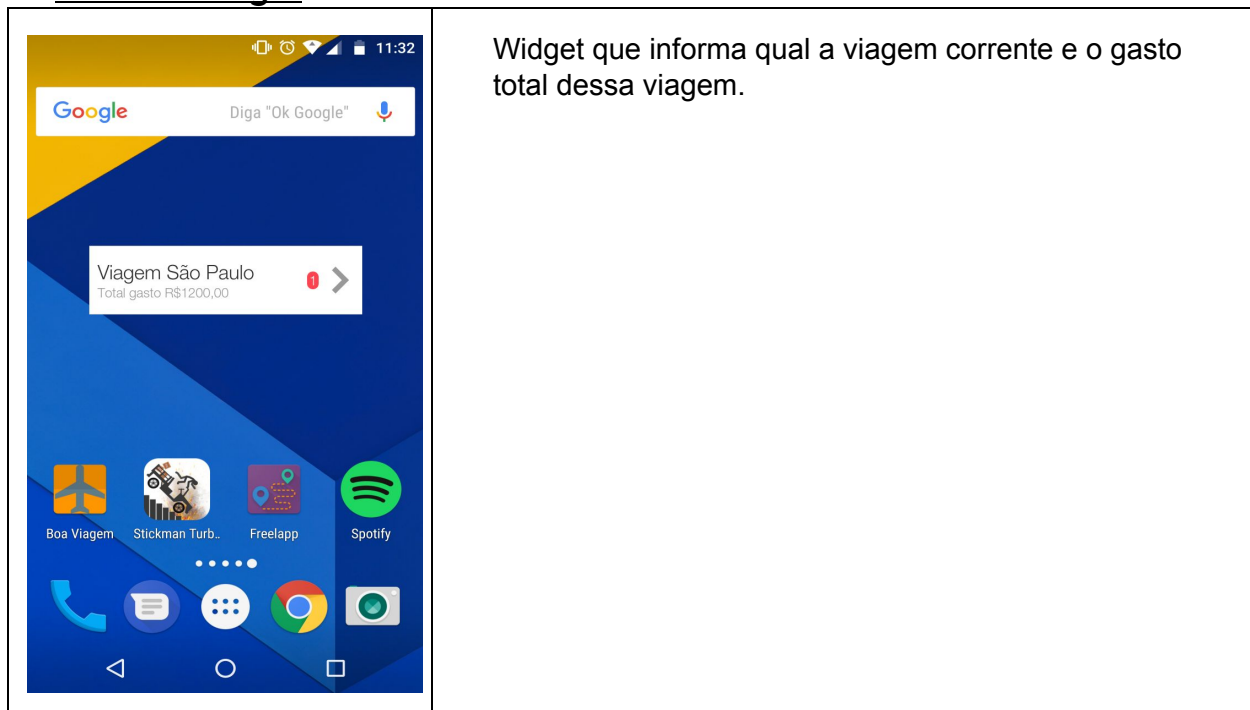
## Tela 5 - Lista de gastos/ Adicionar gastos

	<p>Nessa tela o viajante pode ver os gastos separados por viagem.</p>
---	---

## Tela 6 - Detalhes de gastos

	<p>Após solicitar a adição de um novo gasto o usuário é direcionado para esta tela.</p>
---	---

## **Tela 7 - Widget**



## **Considerações Chave/Key Considerations**

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

Para este app, utilizarei um banco de dados local usando SQLite.

Implementarei um classe que estende ContentProvider para manter a implementação do banco de dados padronizada e compartilhá-las quando necessário.

E também utilizarei duas classes que estendem SimpleCursorAdapter para que os dados salvos no banco de dados possam ser exibidos na listagem de viagens e na listagem de gastos.

Além disso, irei utilizar o AsyncTask para que os dados da viagem a ser editada possam ser exibidos na tela, para que o viajante faça as alterações desejadas.

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

Não acredito que o viajante tenha muitos problemas para ir de uma tela a outra no aplicativo, a única diferença é que após o login o viajante não poderá mais retornar às telas

de login e de cadastro clicando no backbutton do aparelho, isso só será possível se ele fizer o logoff pelo menu da actionBar.

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

Usarei as bibliotecas do firebase auth, para permitir a segurança dos dados do viajante. Usarei também o glide para cacheamento das imagens que poderão ser adicionadas ao aplicativo, bem como o avatar do viajante.

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

O aplicativo usará a biblioteca de AdMob e a api de localização de lugares Google Places para facilitar a adição do local onde o viajante está hospedado ou onde ele realiza seus gastos.

## **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**

- Adicionar ao arquivo build.gradle do aplicativo as dependência:
  - Firebase Auth
  - Glide
- Adicionar o projeto no console do Firebase e adicionar ao projeto o arquivo google-services.json
- Adicionar o plugin do Google Services.

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

- Construir a UI de Registro do viajante
- Construir a UI de Login do viajante
- Construir a UI de Lista de viagens/ Adicionar viagens
- Construir a UI de Detalhes da viagem
- Construir a UI de Lista de gastos/ Adicionar gastos
- Construir a UI de Detalhes de gastos
- Definir os esquemas de cores no arquivo colors.xml dentro da pasta res/values.
- Definir as strings iniciais do aplicativo no arquivo de strings.xml dentro da pasta values
- Adicionar as imagens básicas necessárias na pasta de drawable.

### **Tarefa 3: Construir classes de modelo**

- Adicionar os atributos da classe de modelo correspondente a cada objeto.

### **Tarefa 4: Construir as activities e fragments das funcionalidades.**

- Implementar a activity e o fragment das funcionalidades listadas.
- Construir uma classe AsyncTask para preencher os campos da tela de Dados da Viagem e Dados de Gasto com as informações do banco de dados.



### **Tarefa 5: Construir camada de persistencia de dados**

- Implementar camada do app que é responsável por manter os dados do viajante armazenados na memória do dispositivo.

### **Tarefa 6: Construir Widget.**

- Implementar widget do aplicativo para que o usuário acompanhe o valor de total de gastos da viagem atual.

---

### **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**”