

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Tela 1](#)

[Tela 2](#)

[Tela 3](#)

[Tela 4](#)

[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: Criação do Banco de Dados](#)

[Tarefa 4: Conexão da Base de Dados e Implantação da lógica do APP](#)

[Tarefa 5: Criação de testes](#)

Usuário do GitHub: FelipeBS86

TelePrompter Mobile

Descrição

TelePrompter Mobile é um aplicativo desenvolvido utilizando a linguagem Java, que permite ao usuário ler sua fala em frente à câmera sem ter que desviar seu olhar. Por possuir auto rolagem de texto, com velocidade configurável, sua leitura poderá ser fluida e natural.

Público-Alvo/Intended User

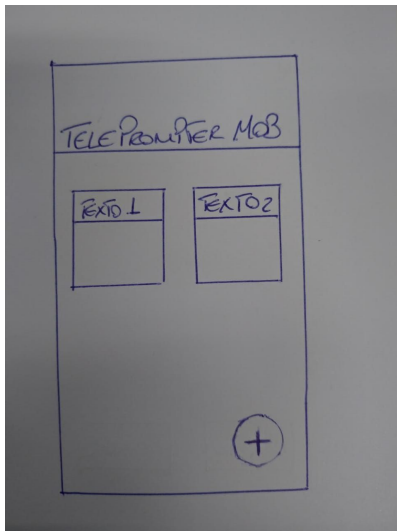
Apresentadores de tv, discursantes, palestrantes e qualquer pessoa que necessite realizar algum discurso.

Funcionalidades/Features

- Tamanho de fonte configurável
- Receba textos pre-escritos a partir de uma api do Firebase
- Inicie um discurso e interrompa a qualquer momento
- Ajuste de velocidade de rolagem
- Acessibilidade via D-Pad
- Todas as Strings utilizadas no projeto no arquivo Strings.xml

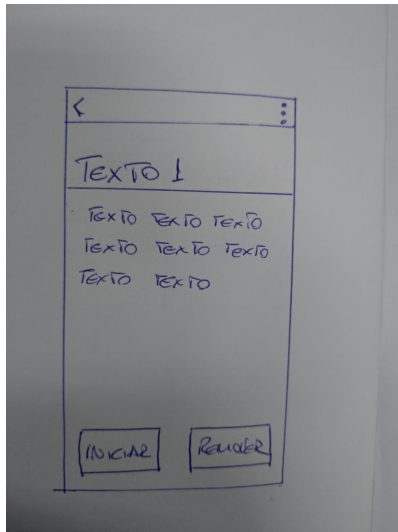
Protótipo de Interfaces do Usuário

Tela 1



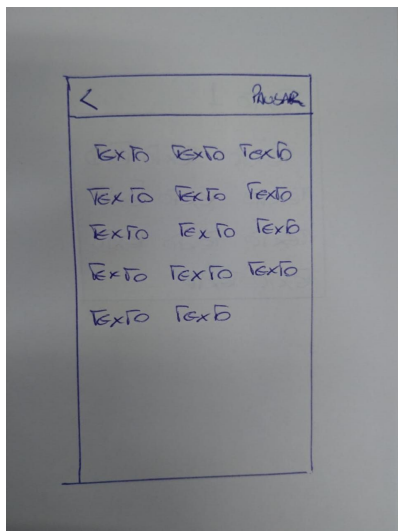
Tela principal do TelePrompter Mobile. Possui uma AppBar, CardViews que com os textos adicionados, ao clicar em um card o usuário é direcionado a tela de detalhes do texto. Além disso o app possui um fab para que possa adicionar um novo texto.

Tela 2



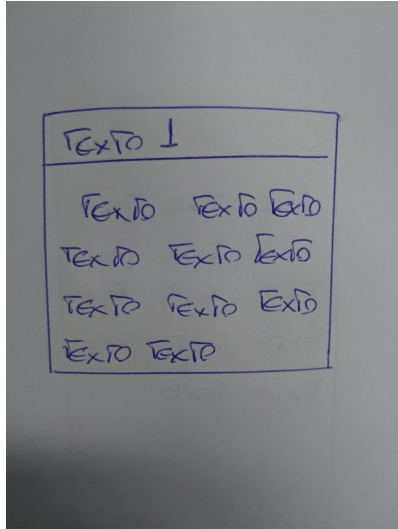
Tela de detalhes do texto selecionado, a partir dela o usuário poderá acessar a tela de configuração e além disso poderá iniciar a rolagem do texto ou deletar o texto corrente.

Tela 3



Tela de visualização da rolagem de texto. Ao clicar no Iniciar da tela 2 essa tela será exibida e a rolagem de texto irá começar. Nessa tela há um botão para o usuário pausar a execução do texto caso seja necessário.

Tela 4



Exemplo do Widget para o app TelePrompter Mobile, nele será exibido o título do texto e uma parte do texto correspondente.

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

A Aplicação irá se conectar irá utilizar o SQLite para armazenar os textos adicionados e um Content Provider para carregar os textos na tela.

Para a busca dos dados no banco será utilizado um AsyncTaskLoader.

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

Os textos salvos não poderão ser editados, o usuário deverá incluir um novo texto caso queira realizar alguma alteração.

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

Serão utilizadas as seguintes bibliotecas:

- Google AppCompat
- ButterKnife
- Timber
- Espresso
- Junit

Descreva como você implementará o Google Play Services.

Os seguintes serviços serão utilizados:

- AdMob
- Analytics API

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto/Project Setup

- Download e instalação do Android Studio
- Criação de um novo projeto com SDK 19 como o SDK mínimo
- Escolha das cores do APP
- Configuração dos flavors gratis e pago

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

- Implementar a UI da Activity Principal
- Implementar as Activitys secundárias
- Implementar os fragments necessários
- Implementar o Shared Preferences para as configurações
- Ajustar a aparência final do APP

Tarefa 3: Criação do Banco de Dados

- Planejamento do Schema
- Implementar do Schema
- Criação da base de dados e as classes de contrato
- Implementar o Content Provider

Tarefa 4: Conexão da Base de Dados e Implantação da lógica do APP

- Implementar as lógicas, classes e métodos necessários
- Implementar métodos para busca dos dados do banco
- Criação de AsyncTask para busca de textos da base remota (Firebase)

Tarefa 5: Criação de testes

- Criação de classes e métodos de testes unitários
- Implementar log para a aplicação

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**”