

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Tela 1](#)

[Tela 2](#)

[Tela 3](#)

[Tela 4](#)

[Tela 5](#)

[Tela 5.1](#)

[Tela 5.2](#)

[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: Adicionar as classes base](#)

[Tarefa 3: Criar camada de domínio da aplicação](#)

[Tarefa 4: Criar camada de dados da aplicação](#)

[Tarefa 5: Criar camada de apresentação da aplicação](#)

[Tarefa 6: Juntar as três camadas](#)

[Tarefa 7: Testes unitários e de integração](#)

[Tarefa 8: Adicionar a configuração de assinatura](#)

**Usuário do GitHub:** [candalo](#)

# Reading Diary

## Descrição

Um aplicativo que auxilia leitores a manter um histórico de todas as suas leituras. Salve comentários em texto e em áudio dos livros que está lendo, fazendo assim um diário das suas leituras!

## Público-Alvo/Intended User

Leitores de qualquer gênero literário


## Funcionalidades/Features

- Salva comentários em texto de determinado livro
- Salva comentários em áudio de determinado livro
- Pesquisa livros dado um termo de pesquisa
- Exibe informações de um livro
- Permite criação de categorias
- Permite salvar um livro em uma categoria

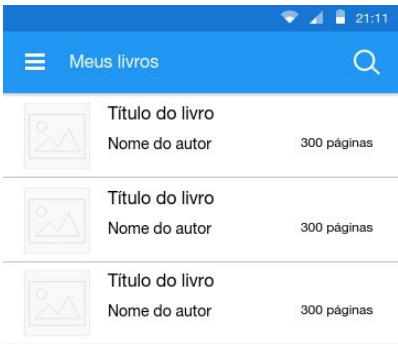
## Protótipo de Interfaces do Usuário

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>Nessa tela o usuário poderá fornecer um termo de pesquisa, e de acordo com esse termo serão listados os livros, exibindo uma imagem da capa do livro, o título, o nome do autor e a quantidade de páginas.</p>
---	---

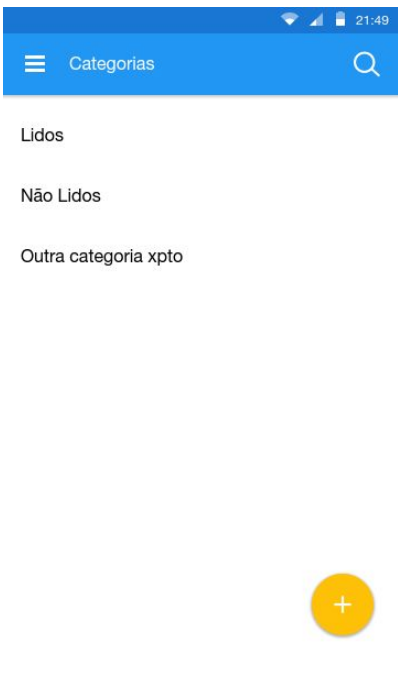
## Tela 2

	<p>Nessa tela serão listados todos os livros selecionados pelo usuário, exibindo uma imagem da capa do livro, o título, o nome do autor e a quantidade de páginas.</p>
---	--

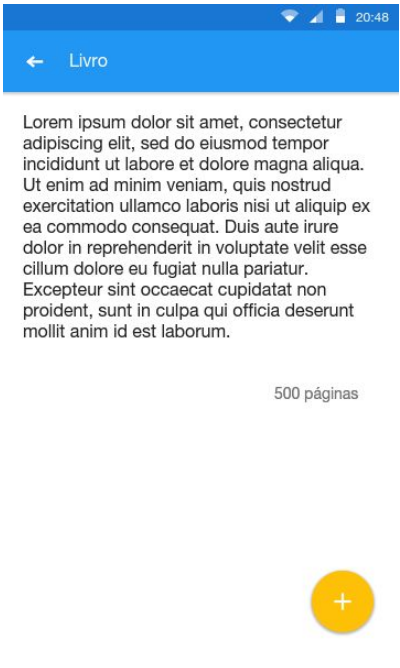
### Tela 3

	<p>Essa tela é semelhante à Tela 2. A diferença é que nessa tela serão exibidos os últimos livros vistos pelo usuário. O número de livros será limitado.</p>
--	--


### Tela 4

	<p>Nessa tela serão listadas as categorias criadas pelo usuário. Há também a possibilidade de criar uma nova categoria, clicando no floating action button.</p>
---	---

## Tela 5

	<p>Nessa tela serão exibidos alguns detalhes do livro, como sua sinopse e a quantidade de páginas. Além disso, há a possibilidade de realizar algumas ações clicando no floating action button.</p>
---	---

## Tela 5.1

	<p>Tela 5 com as informações que o usuário poderá adicionar em cada livro. As informações são registros de texto e áudio.</p>
---	---

## Tela 5.2

<p>The screenshot shows a mobile app interface for a book. At the top is a blue header bar with a back arrow and the word 'Livro'. Below the header is a large text area containing Lorem Ipsum placeholder text. To the right of the text, it says '500 páginas'. At the bottom, there are three horizontal sections, each with a floating action button (FAB) on the right. The first section has a pencil icon and the text 'Lorem ipsum dolor sit amet, consectetur...'. The second section has a microphone icon. The third section has a yellow FAB with a white 'X' icon.</p>	<p>Tela 5 com a exibição dos itens que aparecerão ao clicar no floating action button. Esses botões permitem criar um comentário de texto, gravar um áudio e adicionar o livro em uma categoria.</p>
--	--

## Widget

<p>The screenshot shows a mobile app interface with a widget. The widget is a gray box with the title 'Último livro visto' and two lines of text: 'Aqui vai o título do livro' and 'Aqui a sinopse do livro'. The widget is positioned in the center of the screen. At the bottom, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.</p>	<p>O widget do app irá exibir algumas informações do último livro visto pelo usuário. Essas informações são: título e sinopse do livro. Se o usuário efetuar um clique, será aberto o app no livro em questão.</p>
--	--

## Considerações Chave/Key Considerations

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

A persistência de dados se dará por meio de um ContentProvider que eu irei criar. O app terá duas tabelas: uma para as coleções e outra para os livros.

Quando o usuário criar um novo item na coleção, será armazenado localmente.

Quando o usuário selecionar um livro na Tela 1, esse livro será armazenado localmente.

Também é importante notar que haverá uma relação de 1 - N entre livros e coleções. Sendo assim, um livro pode estar salvo em mais de uma coleção.

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

O fluxo de telas do app é bem simples. Os casos específicos são os seguintes:

- Na Tela 1, ao realizar a pesquisa de livros e selecionar um, o usuário será redirecionado para a Tela 3;
- Na Tela 5, se o usuário pressionar o botão voltar presente na Toolbar ou o botão voltar do device, ele voltará para a Tela 2 ou Tela 3. A tela que irá voltar vai depender de onde ele estava anteriormente. Se o usuário entrar na Tela 5 a partir da Tela 2, se ele voltar, vai para a Tela 2. Se o usuário entrar na Tela 5 a partir da Tela 3, se ele voltar, vai para a tela 3;
- Ao clicar no widget, entrará na Tela 5 do livro em questão.

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

- Picasso para carregamento de imagens
- Butterknife para injeção de views
- Android Floating Action Button para facilitar o trabalho com floating action buttons customizados
- Retrofit para comunicação com a api do Google Books
- Material Search View para facilitar o trabalho com a pesquisa de itens de forma consistente
- Dagger para injeção de dependências
- RxJava e RxAndroid para realização de tarefas assíncronas
- Ultimate Recycler View para facilitar ações em itens da lista

- Schematic para facilitar criação do Content Provider

## Descreva como você implementará o Google Play Services.

Eu irei utilizar dois serviços da biblioteca Google Play Services: Google Analytics e Google Mobile Ads. O Google Analytics eu irei usar para mapear as ações mais importantes do usuário no app e o Google Mobile Ads eu irei usar para fornecer uma versão gratuita (com anúncios) e uma versão paga (sem anúncios)

## 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 novo projeto no Android Studio
- Selecionar apenas a opção **phone and tablet**
- Selecionar a API 19 como a api mínima
- Adicionar as bibliotecas necessárias no arquivo build.gradle do módulo app

### Tarefa 2: Adicionar as classes base

- Adicionar a classe base para os casos de uso
- Adicionar a classe base para os presenters

### Tarefa 3: Criar camada de domínio da aplicação

- Adicionar os models de domínio
- Adicionar os casos de uso (salvar livro, criar categoria, etc.)
- Adicionar interfaces de fronteira com a camada de dados (interface para o repositório, p.ex.)
- Verificar e validar os erros de domínio



#### **Tarefa 4: Criar camada de dados da aplicação**

- Adicionar configuração do Content Provider
- Verificar e validar os erros ao salvar algum dado no banco de dados local
- Adicionar configuração da api
- Verificar e validar os erros ao realizar uma requisição na api

#### **Tarefa 5: Criar camada de apresentação da aplicação**

- Criar todos os layouts necessários
- Adicionar as activities
- Adicionar os fragments
- Verificar e validar todas as entradas de dados
- Adicionar suporte para acessibilidade em todas as telas

#### **Tarefa 6: Juntar as três camadas**

- Adicionar configuração do Dagger para injetar as dependências entre as camadas

#### **Tarefa 7: Testes unitários e de integração**

- Adicionar uma ampla cobertura de testes no app, utilizando testes unitários e testes de integração

#### **Tarefa 8: Adicionar a configuração de assinatura**

- Gerar e adicionar no repositório a keystore e senha para a assinatura do aplicativo