

## 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: **lucaslprimo**

# Classes Manager

## Descrição

Esse app facilita a interação entre professores particulares e seus alunos, provendo para o aluno informações sobre o conteúdo das aulas, posteriores e passadas, tarefas sugeridas em cada aula, links úteis de material de apoio e datas das aulas todas informadas pelo professor via app. O professor pode adicionar os alunos no app, liberando então acesso para eles. O aplicativo conta também com notas do professor, sobre a aula, que ficam visíveis apenas para ao professor.

## Público-Alvo/Intended User

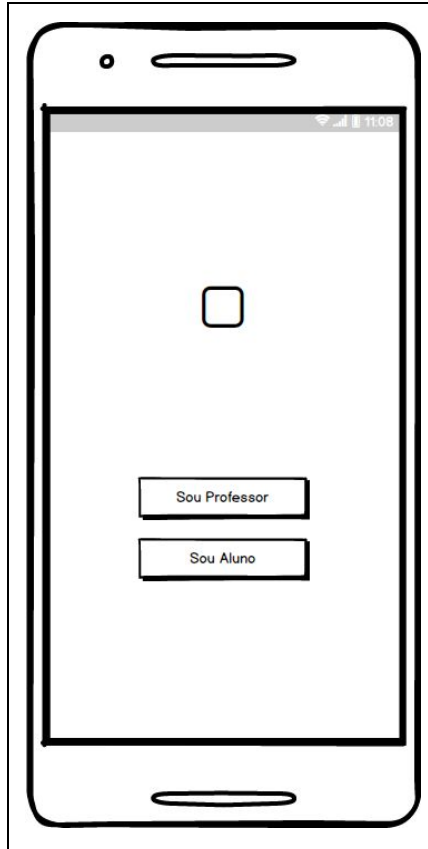
Professores ou pessoas que dão aulas particulares.

## Funcionalidades/Features

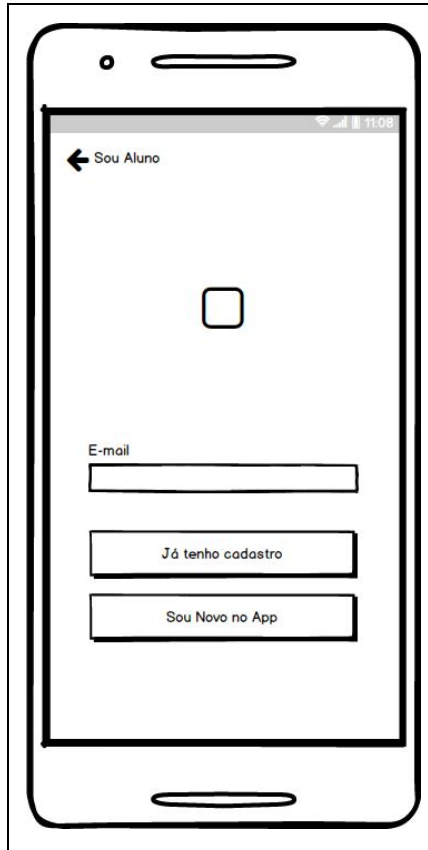
- Salva Informações
- Notifica sobre horários de aulas
- Sincroniza dados via internet
- Listagem de Alunos, Tarefas, Aulas e Links
- Adição e edição de Alunos, Tarefas, Aulas e Links

## 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.



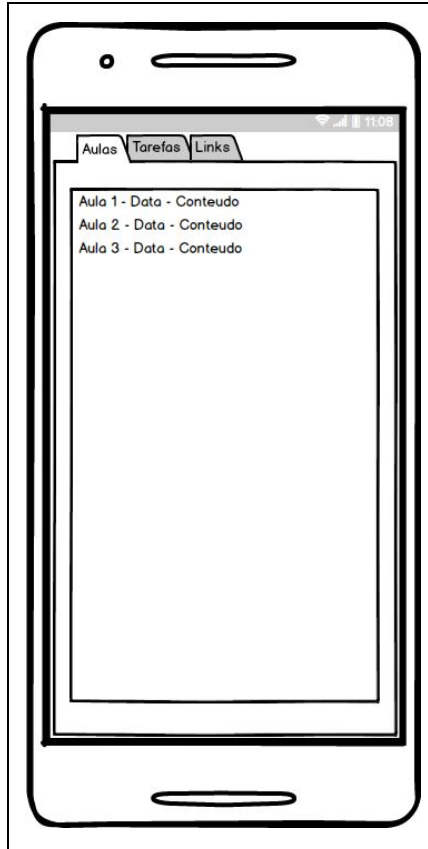
Login Inicial, é possível logar como Aluno ou Professor.



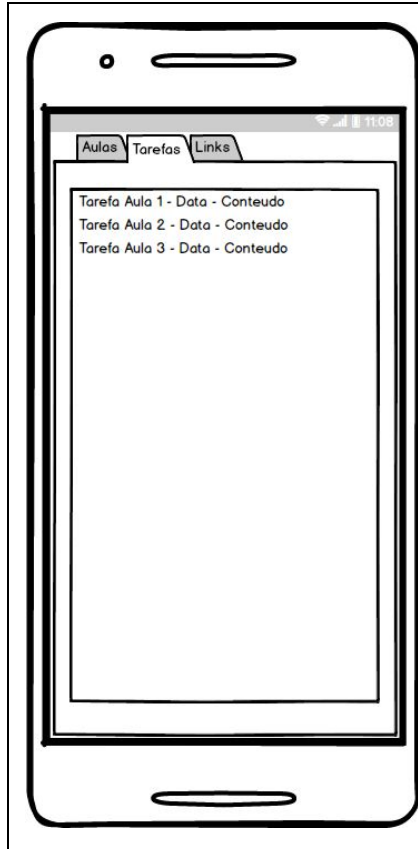
Caso selecione aluno ele deve entrar com email para logar, se for o primeiro acesso ele deve clicar em “Sou novo no App”

### Tela 3

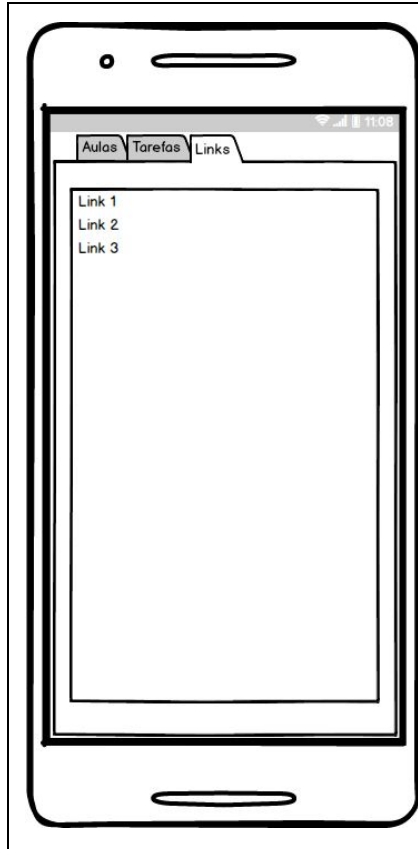
 <p>← Sou Novo</p> <p></p> <p>Criar Senha</p> <p>Confirmar Senha</p> <p>Salvar</p>	<p>Aluno cria uma nova senha para seu acesso.</p>
--	---



Após o Login o aluno tem acesso a 3 abas: Aba 1 de 3 - Aulas

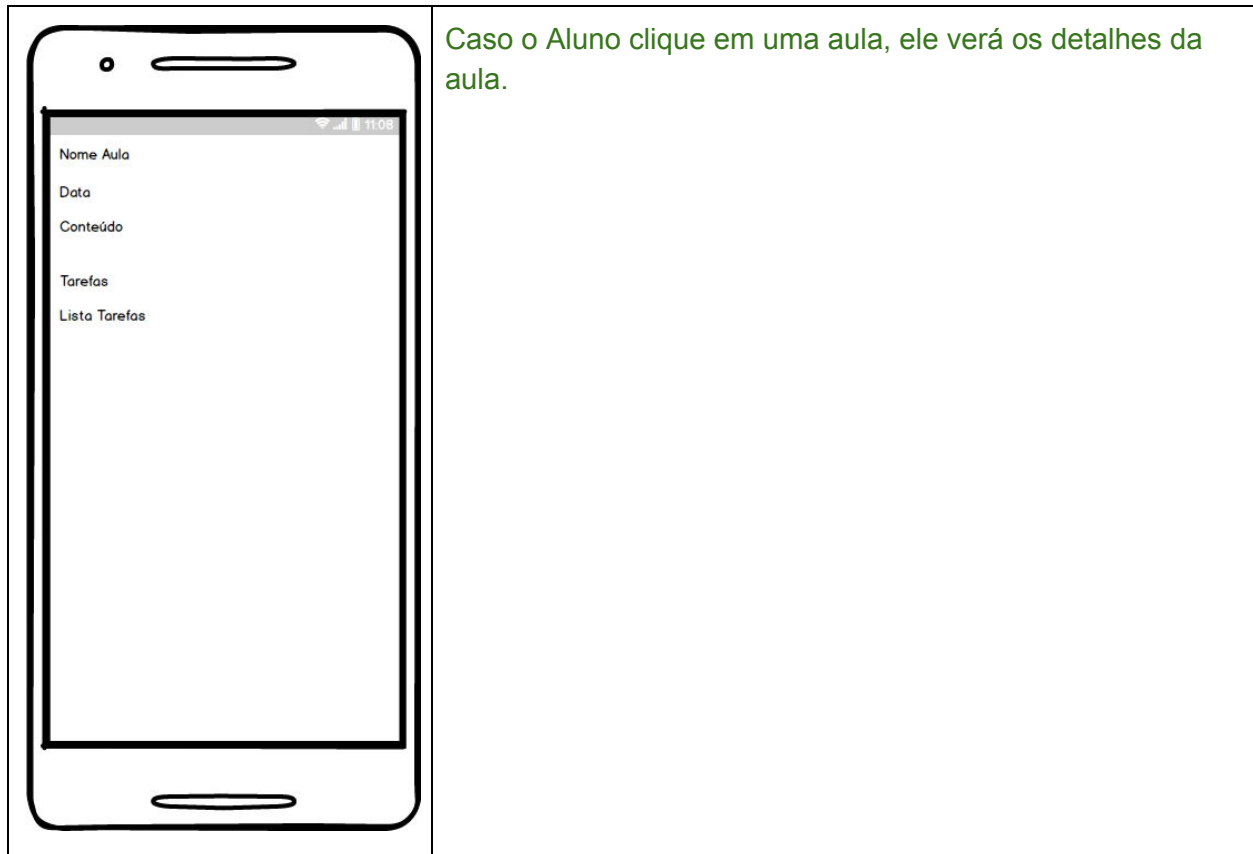


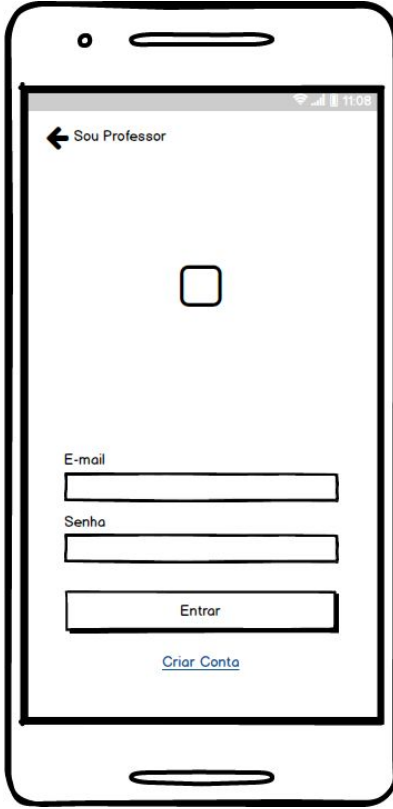
## Aba 2 de 3 - Tarefas

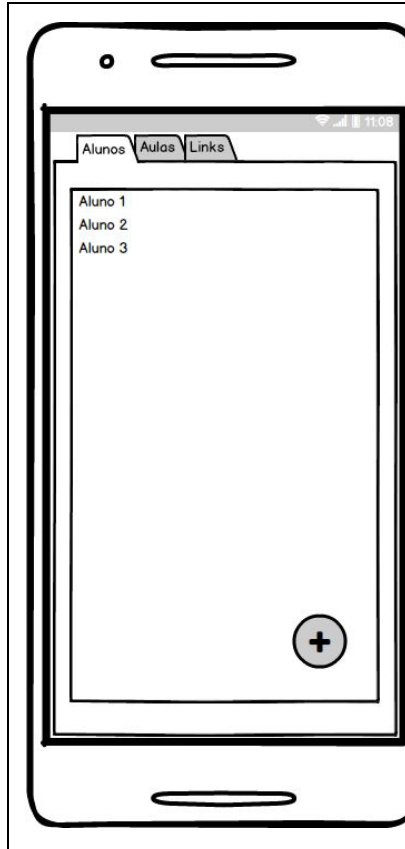


Aba 3 de 3 - Links

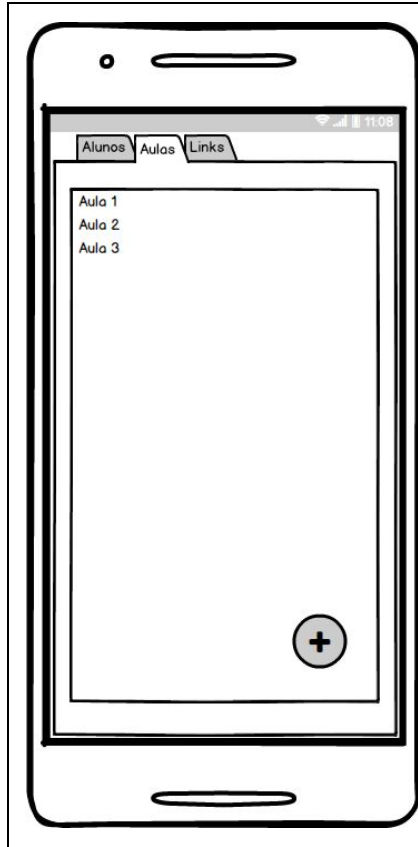




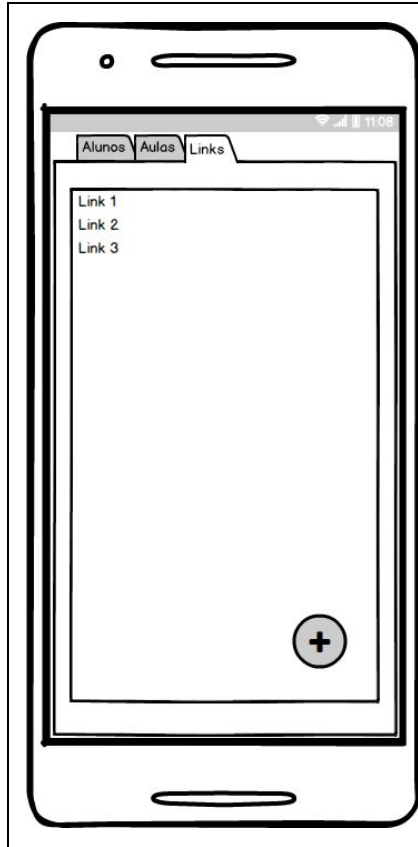
 <p>The image shows a mobile application interface for a login screen titled "Sou Professor". At the top left, there is a back arrow icon. Below the title, there is a square icon representing a fingerprint scanner. Underneath, there are two input fields: the first is labeled "E-mail" and the second is labeled "Senha". Below these fields is a button labeled "Entrar". At the bottom, there is a link labeled "Criar Conta" in blue text.</p>	<p>Caso selecionada a opção “Sou Professor” o usuário deverá entrar com email e senha ou clicar em “Criar Conta”</p>
---	--



Tela inicial de professor, com 3 Abas: 1 de 3 - Alunos

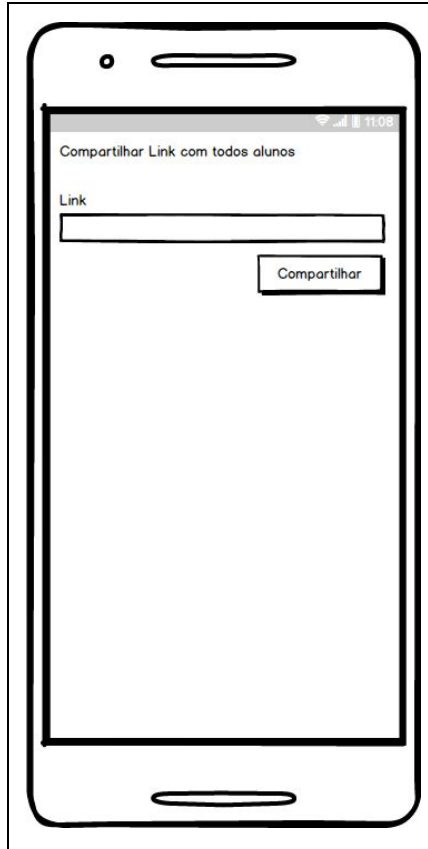


Aba 2 de 3 - Aulas



Aba 3 de 3 - Links

 <p>The mockup shows a mobile application interface for adding a student. At the top, the status bar displays the time as 11:08. The app's title bar reads 'Adicionar Aluno'. Below the title, there are two input fields: the first is labeled 'Nome' and the second is labeled 'E-mail'. A button labeled 'Adicionar' is positioned to the right of the 'E-mail' field. The entire interface is framed by a black border representing the phone's screen.</p>	<h2>Adição e Edição de Aluno</h2>
---	-----------------------------------

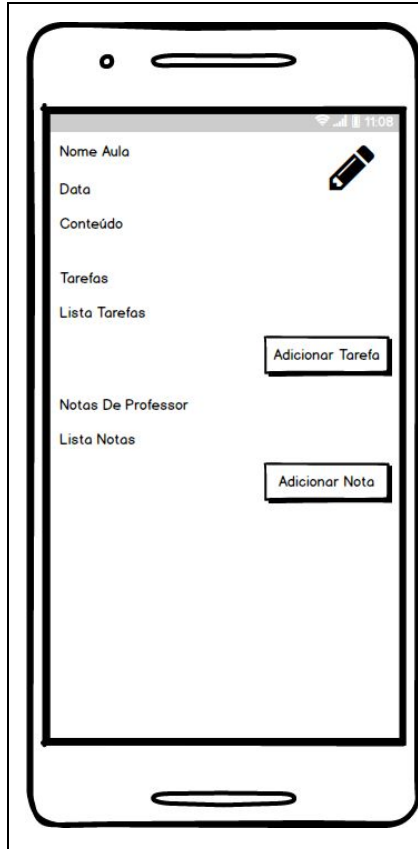


Adição de Link que será compartilhado com todos os alunos

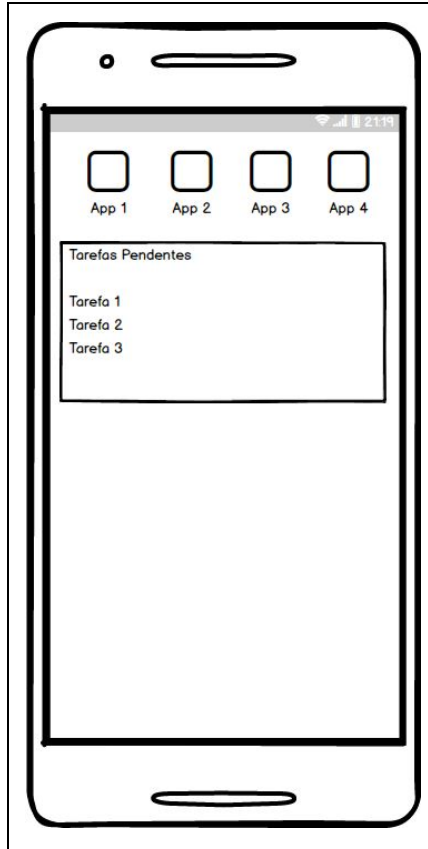
 <p>The mockup shows a mobile application interface. At the top, the title 'Adicionar Nota de Aula' is displayed. Below the title is a text input field labeled 'Nota'. To the right of the input field is a button labeled 'Adicionar'. The interface is framed by a black border representing the phone's screen.</p>	<p>Adição de Nota de Aula</p>
---	-------------------------------



 <p>The mockup shows a mobile application interface for adding a task. At the top, the status bar displays the time as 11:08 along with signal and battery icons. The app's title bar is labeled 'Adicionar Tarefa'. Below the title, there is a text input field with the placeholder 'Descrição da Tarefa'. To the right of this field is a button labeled 'Adicionar'. The entire interface is framed by a black border representing the phone's screen.</p>	<h2>Adição e Edição de Tarefa</h2>
---	------------------------------------



Tela de Detalhes das Aula



Widget Aluno, mostra uma lista com as tarefas pendentes



## Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Será utilizado Firebase Realtime Database para tratamento dos dados de maneira online e offline.

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

O Aluno entra no app com seu email e senha e pode verificar as aulas agendadas, as tarefas pendentes e os links compartilhados pelo professor.

Caso o Aluno ainda não tenha senha ele deve criar uma nova senha para o primeiro acesso.

O Professor pode criar aulas, tarefas das aulas, notas sobre a aula e links.

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

- Glide para tratar o carregamento e cache das imagens.
- Firebase Realtime Database para sincronização de dados online e offline.
- Google Cloud Messaging pra notificações push
- Butter Knife para o bind de views.
- Retrofit para requisições web.

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

Realtime Database do Firebase será utilizado e Google Cloud Messaging.

## Próximos Passos: Tarefas Necessárias

### Tarefa 1: Configuração do Projeto/Project Setup

- Criar Projeto
- Adicionar Dependências Gradle necessárias

### Tarefa 2: Implementar Estrutura de Dados

- Criar classes entidade
- Criar classes de comunicação com banco de dados
- Criar classes de comunicação com API

### Tarefa 4: Implementar Firebase

- Configurar Firebase

### Tarefa 4: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

- Implementar Activity de Login
- Implementar Activity de Início
- Implementar Activity de Edição e detalhes de Aulas, Alunos e Tarefas

### Tarefa 4: Testes

- Testar fluxo de criação de Alunos e professores
- Criação de Aulas e tarefas
- Criação e compartilhamento de Links

---

### 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”**

