

Documentação sprint

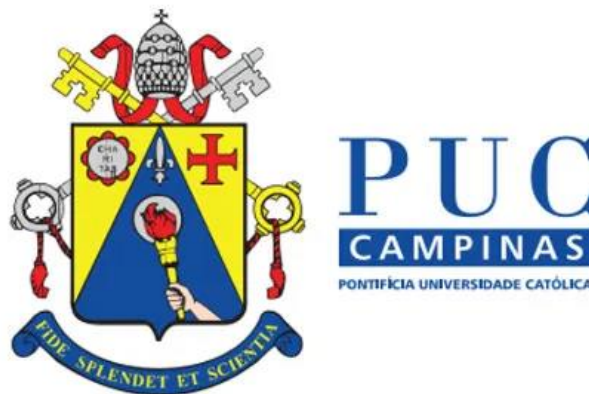
29/09/2022



PUC
CAMPINAS
PONTIFÍCIA UNIVERSIDADE CATÓLICA

Documentação sprint

29/09/2022



Guilherme Marques Brait Garros	19264266
Renan Willian Guimarães	19315613
Alcides Gomes Beato Neto	19060987
Lucas Cicconi Ferreira	17121393
Giulia Cerruti Gonçalves	20106894
Lucas Grass Beraldo	19246925
Derek freire quaresma	18706986
Vitor Ferrara Conti	19019454
Henrique Sartori Siqueira	19240472
Renato Barba dos Santos	19246529
Rafael Silva Barbon	19243633
Victor Luiz ildebrand de Souza	19244227

Sumário

1. Banco de dados	pag.3
1.1 MER	
2. Estrutura do back-end - App Mobile com React Native	pag.3
3. Front-end mobile	pag.4
4. Front-end Web	pag.5
5. Design	pag.7
6. Regras de negócio	pag.8

1.Banco de Dados:

1.1.MER

Foi criado o MER do projeto, mapeando as entidades descobertas até o momento

2.Estrutura do back-end - App Mobile com React Native:

Foi feita a estruturação do app mobile com react native e utilizando os seguintes Libs e Plugin

- Luxon
- React Navigation
- React Native Flash Messages
- Vanilla Masker
- Styled Components

E a seguinte estrutura

Em relação a Estrutura:

O Projeto irá utilizar uma organização utilizando como base a pasta /scr que irá ter as funcionalidades divididas por contexto (components, services, pages e etc...)

- Navegação e Telas

O Projeto irá utilizar stacks de navegação e essas stacks estarão divididas entre as telas já divididas por contexto

Exemplo:

```

pages/
├─ auth/
│  ├─ authStack.tsx
│  ├─ login/
│  │  ├─ index.ts
│  │  ├─ login.tsx
│  │  └─ login.style.ts
│  └─ recovery/
│     ├─ index.ts
│     ├─ recovery.tsx
│     └─ recovery.style.ts
├─ content/
│  ├─ contentStack.tsx
│  ├─ home/
│  │  ├─ homeStack.tsx
│  │  ├─ profile/
│  │  │  ├─ index.ts
│  │  │  ├─ profile.tsx
│  │  │  └─ profile.style.ts
│  │  └─ test/
│  │     ├─ index.ts
│  │     ├─ test.tsx
│  │     └─ test.style.ts
└─ ...
-

```

Componentes:

O Projeto irá utilizar o Atomic Design, em que dividimos os componentes por complexidade e reusabilidade

Controle de Estado:

Decidimos a utilização do Redux como um controlador de estado da aplicação e nele aplicamos o Module Pattern de Organização, já dividindo suas actions / reducers / types por contexto

3.Front-end mobile:

Foi listado todas as telas do APP Mobile, com todos os campos e informações que devem ser exibidos em cada uma.

- Login
CPF e Senha
- Recuperar
CPF ou email
- Home
Listagem de Consultas
Nome Usuário
- Notificações
Listagem de Notificações
- Detalhes Consulta
Tipo Consulta
Local e Horário
Status (Aceita, Pendente, Cancelada)
Opções de Cancelamento e Confirmação
Informações do Paciente / Aluno e Professor

4.Front-end Web:

Telas:

Login

- Campos: E-mail e Senha do usuário
- Informações: Formulário de login, botão para cadastrar e recuperar senha

Register

- Campos: Nome, E-mail, Senha, Confirmar Senha e ID
- Informações: Formulário para se registrar e botão para voltar para tela de login

ForgotPassword

- Campos: E-mail e Confirmar E-mail
- Informações: Formulário para envio de um e-mail com um código para recuperar senha e botão para voltar para tela de login

Confirm

- Campos: 5 espaços para digitar 1 número em cada
- Informações: Formulário para digitar o código recebido no e-mail (tela só tem acesso se o usuário tenha mandado uma requisição de troca de senha)

RecoverPassword

- Campos: Nova Senha e Confirmar nova Senha

- Formulário para troca de senha (tela só tem acesso se o usuário tenha mandado uma requisição de troca de senha)

Telas Privadas:

Home / Agenda

- Campos: Filtros (Especialidade, Dia, Professor, Status, Hora, Aluno e Período da Semana)
- Informações: Listagem dos dias da semana e as consultas que têm em cada dia com status da consulta e o professor que realizará e ao clicar na consulta abre o AppointmentDetails e botão para ver as solicitações (tela Requests)

AppointmentDetails

- Campos: Data da consulta, Nome do paciente, Tel. contato, Nome do acompanhante, Tel. contato acompanhante, Nome do médico (aluno), Nome do médico (professor), Especialidade da consulta e Hora da consulta
- Informações: Dados do paciente e da clínica e botão de voltar para agenda

Requests

- Campos: Nome do aluno, Especialidade, Data Requerida, Data da Consulta, Hora da Consulta e Número da Solicitação
- Informações: Lista de Solicitações, botão para atribuir paciente em cada solicitação e um botão para voltar para tela de Agenda

AssignPatient

- Campos: Nome e CPF do paciente
- Informações: Formulário para atribuir paciente e botão para voltar para tela de Requests

5.Design

6.Regras de negócio:

Foi feito um levantamento das regras de negócio de cada persona levantada no projeto, dentre elas:

- Paciente
- Recepcionista
- Aluno
- Professor
- Responsável técnico