



Programa de estágio



Desafio

Encerramento da primeira fase

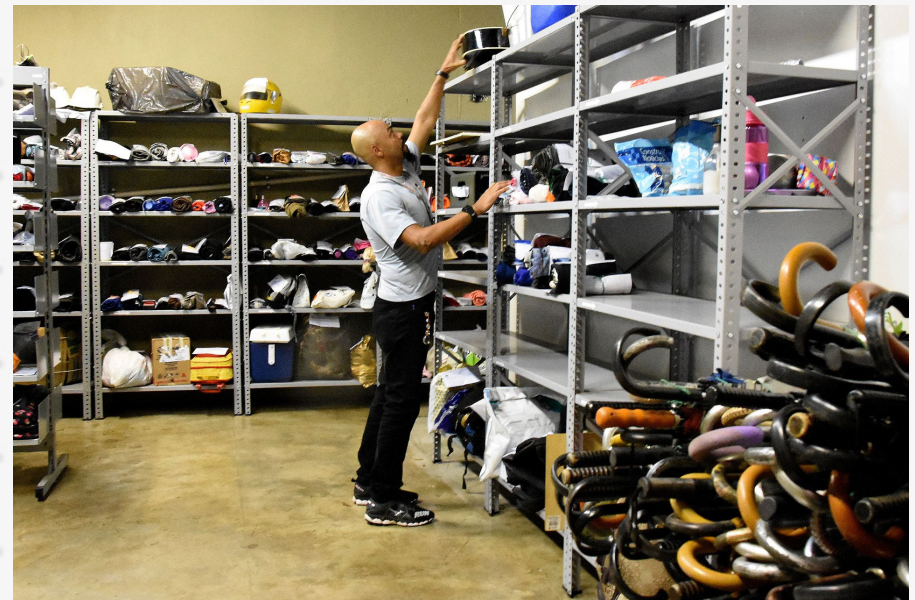
Programa de Estágio



Escopo

Agora é a hora de começarmos por em prática a teoria!

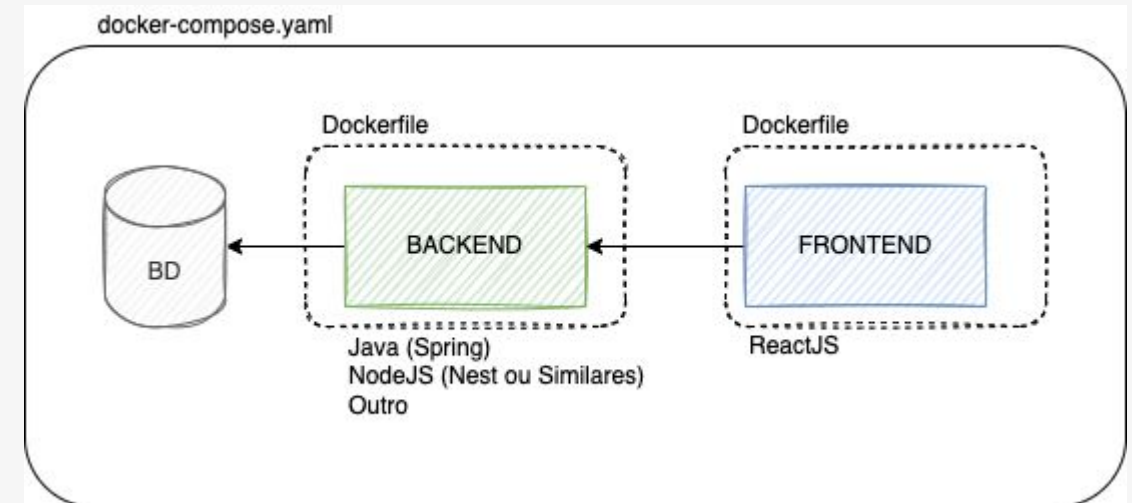
Nosso escopo consiste em um desenvolvimento de um site para gestão de **achados e perdidos!**



Arquitetura

Em nossa arquitetura utilizaremos:

- ❑ API REST
- ❑ Banco de Dados (noSQL ou SQL)
- ❑ ReactJS



Requisitos Funcionais


- ❑ Cadastro de item Perdido
 - ❑ Informar dados do Item: Descrição, Data/Hora, Local (**Mapa**) e Contato
- ❑ Consulta de item Perdidos (area logada / backoffice)
 - ❑ Listagem no Mapa / Tabela
 - ❑ Paginação
- ❑ Alteração de item Perdido
 - ❑ Permitir adicionar histórico

Mockup (Área Deslogada)

Page 1

https://www.draw.io

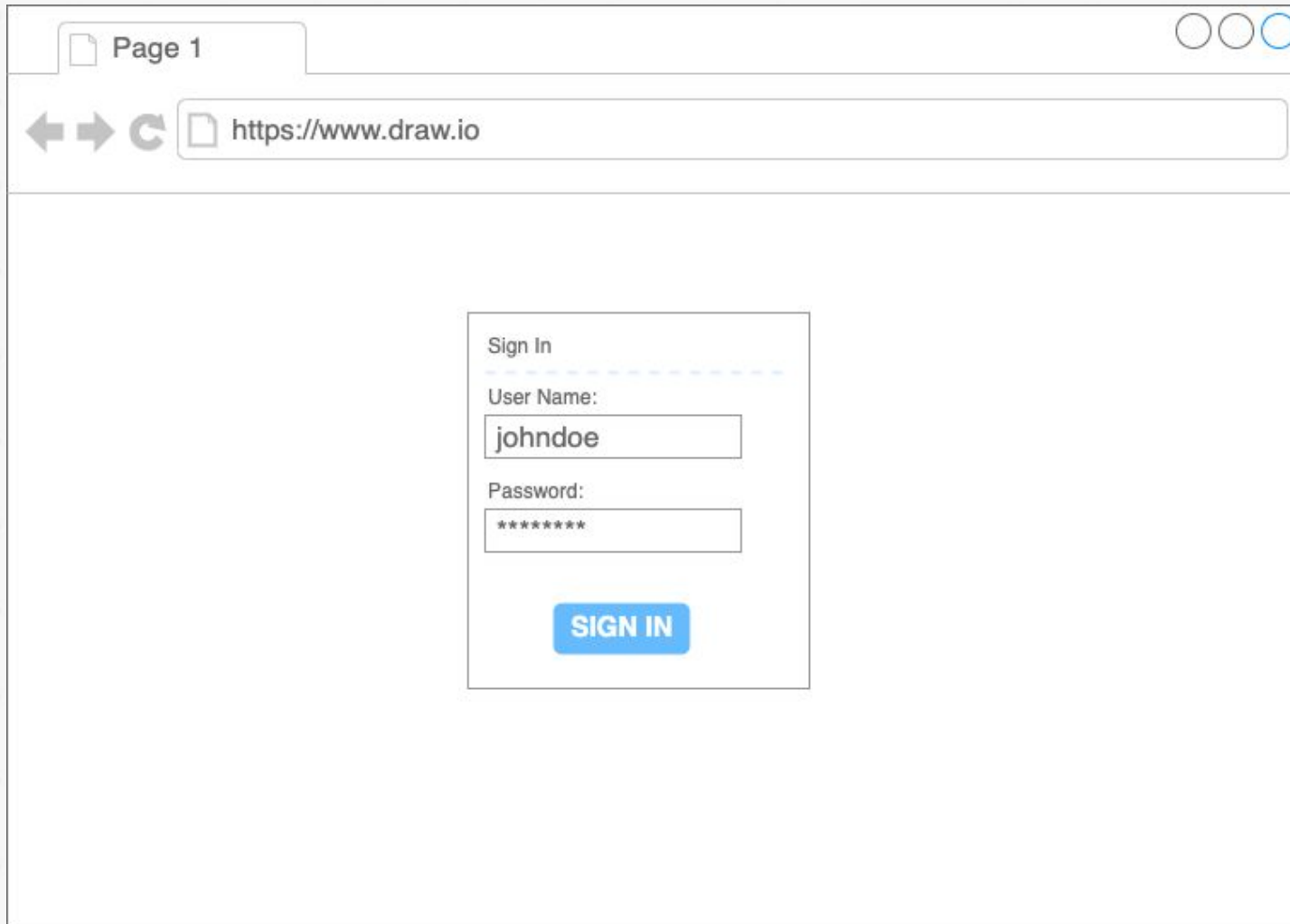
Achados e Perdidos



October 2014

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Mockup (Área Logada)



The mockup shows a web browser window with a single tab labeled "Page 1". The address bar displays "https://www.draw.io". The main content area features a "Sign In" form with a dashed line separator. The form includes a "User Name:" label, a text input field containing "johndoe", a "Password:" label, and a password input field filled with asterisks. A blue "SIGN IN" button is positioned below the password field.

Page 1

https://www.draw.io

Sign In

User Name:

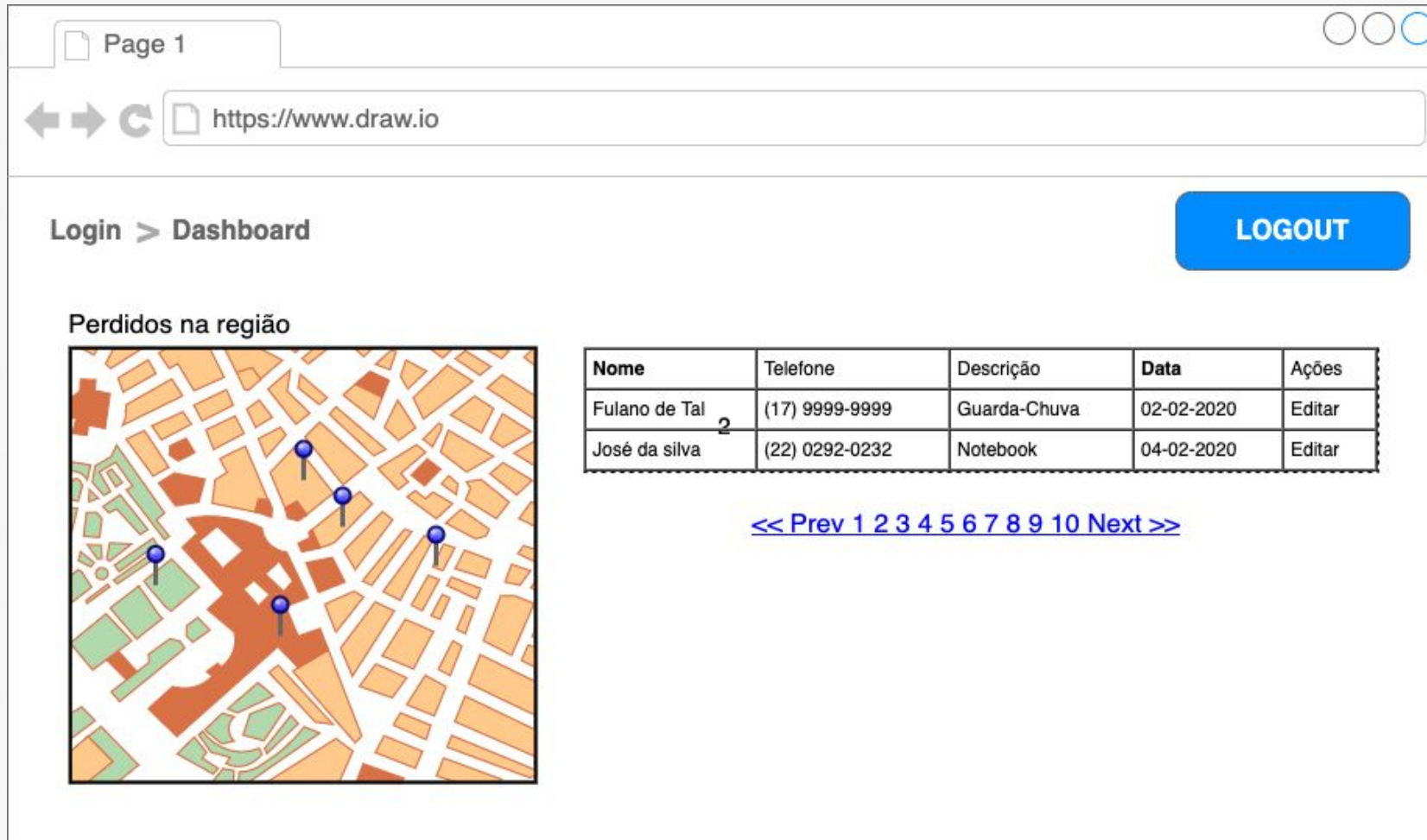
johndoe

Password:

SIGN IN

- ❑ Login Simples
- ❑ Pré Cadastrado

Mockup (Área Logada)



- Items no Mapa
- Listagem de Itens
- Paginação

Mockup (Área Logada)

The mockup shows a web browser window with a single tab labeled 'Page 1'. The address bar contains the URL 'https://www.draw.io'. The page content includes a breadcrumb trail 'Login > Dashboard > Editar' and a blue 'LOGOUT' button. The main section is titled 'Editando Item XXX' and contains four input fields: 'Nome completo', '(xx) xxxxx-xxxx', 'Data', and 'Descrição'. At the bottom, there are two blue buttons: 'FINALIZAR' and 'Histórico'.

Page 1

https://www.draw.io

Login > Dashboard > Editar

LOGOUT

Editando Item XXX

Nome completo

(xx) xxxxx-xxxx

Data

Descrição

FINALIZAR

Histórico

- Edição de Item
- Adição de Histórico

Mockup (Área Logada)

The mockup shows a web application interface for a logged-in user. The interface is divided into a sidebar and a main content area.

Sidebar:

- Page 1
- Navigation links: Login > Dashboard
- Form fields for editing an item: Editando Item XXX, Nome completo, (xx) xxxxx-xxxx, Data, and Descrição.
- Buttons: Salvar and Histórico.

Main Content Area:

- Window Title: Window Title
- Buttons: LOGOUT
- History list:
 - 02-02-2022 15:00 <Admin> - Item foi solicitado - Status SOLICITADO
 - 02-02-2022 16:00 <Admin> - Foi solicitado prova de propriedade do item
 - 02-02-2022 16:00 <Admin> - Item foi solicitado - Status DEVOLVIDO
- Form for adding history: Adicionar Histórico, DEVOLVIDO (dropdown), and a text area with the message "O item foi devolvido corretamente, a pessoa apresentou a nota fiscal".
- Buttons: ADICIONAR and Histórico.

- Edição de Item
- Adição de Histórico

Regras

- ❑ **Usem a criatividade**

- ❑ Mas não deixem a simplicidade de lado! (Não é necessário se prender ao Mockup!)

- ❑ Para o **Frontend** usem React

- ❑ Para o **Backend** sintam-se à vontade em escolher entre Java ou NodeJS

- ❑ Documentar o setup do projeto no README.MD!!!!

- ❑ Vamos trabalhar com FORK e Merge Request

- ❑ Lembrar de fazer commits graduais para vermos o progresso da codificação!!!

- ❑ Façam o uso de Frameworks

- ❑ Ta tudo liberado!!!!

Critérios

- ❑ Backend
 - ❑ Uso correto dos verbos / status code HTTP
 - ❑ Semântica do código
 - ❑ Tratamento de Erros
 - ❑ Autenticação / Segurança
 - ❑ Modelagem de Banco de Dados

Critérios

❑ Frontend

- ❑ Uso semântico do HTML
- ❑ Estruturação de layout, componentes e tags CSS
- ❑ Layout responsível é um BONUS
- ❑ Utilização de ContextAPI, Hooks e Local Storage

Objetivos

- ❑ Avaliar o progresso/eficácia do plano de estudos e programa de estágio.
- ❑ Avaliar independência e capacidade de resolução de problemas.
- ❑ Identificar deficiências no programa.

Prazos

- ❑ Individual (mas pode pedir ajuda pro coleguinha)
 - ❑ Não é competição, é individual e não comparativa
- ❑ **4 á 8 semanas** para desenvolvimento backend e frontend
 - ❑ Checkpoint a cada 2 semanas.
 - ❑ Para os alocados em projeto, haverá um prazo maior (8 semanas)
- ❑ No final do desenvolvimento realizaremos uma apresentação dos projetos para os Tech Managers.

Ficou com alguma dúvida?



