

Achados e perdidos

Trabalho Interdisciplinar - Aplicações Web



Belo Horizonte, Abril/2022

Sumário

**ATENÇÃO → EM VERMELHO OS ITENS DA ENTREGA DA SPRINT 1**

[**Introdução**](#_3znysh7) **[3](#_3znysh7)**

[Problema](#_2lwamvv) [3](#_2lwamvv)

[Objetivos](#_111kx3o) [3](#_111kx3o)

[Justificativa](#_3l18frh) [4](#_3l18frh)

[Público alvo](#_2et92p0) [4](#_2et92p0)

[**Especificação do Projeto**](#_tyjcwt) **[5](#_tyjcwt)**

[Personas](#_3dy6vkm) [5](#_3dy6vkm)

[Histórias de usuários](#_1t3h5sf) [6](#_1t3h5sf)

[Requisitos do Projeto](#_4d34og8) [6](#_4d34og8)

[Requisitos Funcionais](#_2s8eyo1) [7](#_2s8eyo1)

[Requisitos não funcionais](#_17dp8vu) [7](#_17dp8vu)

[Restrições](#_3rdcrjn) [8](#_3rdcrjn)

[**Metodologia**](#_lnxbz9) **[9](#_lnxbz9)**

[Relação de Ambientes de Trabalho](#_35nkun2) [9](#_35nkun2)

[Gestão de código fonte](#_1ksv4uv) [9](#_1ksv4uv)

[Gerenciamento do Projeto](#_44sinio) [10](#_44sinio)

[**Projeto de Interface**](#_z337ya) **[12](#_z337ya)**

[Fluxo do Usuário](#_3j2qqm3) [12](#_3j2qqm3)

[Wireframes](#_1y810tw) [12](#_1y810tw)

[Tela - Home-Page](#_4i7ojhp) [13](#_4i7ojhp)

[Tela - Notícias de Categoria](#_2xcytpi) [14](#_2xcytpi)

[Tela - Resultado de Pesquisa](#_1ci93xb) [14](#_1ci93xb)

[Tela - Leitura Notícias](#_3whwml4) [15](#_3whwml4)

[Tela - Salvar Notícias Preferidas](#_2bn6wsx) [15](#_2bn6wsx)

[Tela - Notícias Preferidas](#_qsh70q) [15](#_qsh70q)

[Tela - Comentários](#_3as4poj) [16](#_3as4poj)

[**Implementação da Solução**](#_2p2csry) **[17](#_2p2csry)**

[Arquitetura da Solução](#_147n2zr) [17](#_147n2zr)

[Diagrama de Componentes](#_3o7alnk) [17](#_3o7alnk)

[Hospedagem](#_23ckvvd) [17](#_23ckvvd)

[Template do Site](#_ihv636) [18](#_ihv636)

[Tela Principal](#_32hioqz) [18](#_32hioqz)

[Modal de Mensagens](#_1hmsyys) [19](#_1hmsyys)

[Tela de Visualização de Notícias](#_41mghml) [19](#_41mghml)

[Funcionalidades do Sistema (Telas)](#_2grqrue) [20](#_2grqrue)

[Visualização de Notícias (RF-01 e RF-02)](#_vx1227) [20](#_vx1227)

[Pesquisa de Notícias (RF-05)](#_1v1yuxt) [21](#_1v1yuxt)

[**Avaliação da Solução**](#_19c6y18) **22**

[6.1 Plano de Testes de Software](#_3tbugp1) [22](#_3tbugp1)

[6.2 Registro de Testes de Software](#_28h4qwu) [23](#_28h4qwu)

[6.3 Plano de Testes de Usabilidade](#_37m2jsg) [23](#_37m2jsg)

[6.4 Registro de Testes de Usabilidade](#_1mrcu09) [23](#_1mrcu09)

[**Referências**](#_46r0co2) **24**

# Introdução

No mundo contemporâneo, a velocidade e precisão da informação têm se tornado cada vez mais cruciais no dia a dia das pessoas e das organizações. A capacidade de acessar dados corretos e oportunos é essencial para navegar em uma realidade complexa e repleta de incertezas. De forma análoga à circulação de informações no espectro político e social, o gerenciamento de pertences no espaço comunitário exige soluções que garantam a integridade e a recuperação eficiente de itens pessoais.

## Problema

Atualmente, muitas instituições e locais públicos possuem sistemas de achados e perdidos que são fragmentados, pouco eficientes e não interconectados. Isso resulta em uma baixa taxa de recuperação de objetos perdidos e em frustração para os usuários que perdem tempo tentando localizar seus pertences sem sucesso. Além disso, a falta de um sistema padronizado e acessível aumenta as chances de itens acabarem esquecidos ou descartados, gerando desperdício e perda.

## Objetivos

O objetivo geral deste projeto é desenvolver um sistema web de achados e perdidos que ofereça uma interface amigável e centralize informações de itens perdidos e encontrados, facilitando a comunicação e a devolução eficiente dos objetos aos seus respectivos donos em uma comunidade específica, como um campus universitário, uma empresa ou um evento.

1. Facilitar Registro e Gerenciamento de Itens: Criar funcionalidades intuitivas para que os usuários possam registrar e gerenciar itens perdidos ou encontrados, incluindo detalhes como descrição, categoria, local e data de perda ou achado.
2. Buscar e Recuperar Itens: Implementar um sistema de busca avançado que permita aos usuários localizarem itens perdidos de maneira eficiente, utilizando filtros e palavras-chave.
3. Promover Interação Segura: Oferecer um sistema de mensagens privadas onde usuários possam comunicar-se de forma segura para coordenar a devolução dos itens.

## Justificativa

Segundo o site do portal da PUC Rio, foi necessário a criação de um Achados e Perdidos para a universidade. Isso pela quantidade de itens que eram perdidos, tanto por alunos, quanto por funcionários. Por mês, eram recebidos mais de 120 itens.  
Com a adoção de uma plataforma digital, espera-se aumentar significativamente a taxa de recuperação de objetos, proporcionando uma experiência mais positiva para os usuários. Além disso, o sistema poderá reduzir o tempo e os recursos gastos pelas instituições ao gerenciar manualmente esses itens. A digitalização do processo de achados e perdidos é um passo importante para a modernização e otimização desse serviço essencial.

[Achados e Perdidos também são esquecidos pelos estudantes e funcionários](http://jornaldapuc.vrc.puc-rio.br/cgi/cgilua.exe/sys/start.htm?infoid=1128&sid=29)

## Público-alvo

Perder itens é um problema que afeta todo mundo, diariamente esquecemos de itens importantes em transportes públicos, lojas, faculdades ou até mesmo dentro de nossa própria casa. Porém, o foco principal da solução apresentada é para estudantes com seus itens pessoais dentro de instituições de ensino, de médio ou superior, e modalidade presencial.

Estes estudantes estão numa fase da vida em que deixam de lidar apenas com problemas pessoais e estudantis e iniciam a vida adulta, trabalhar, ter mais compromissos:

* Pessoas de no mínimo 16 anos.
* Qualquer gênero.
* Atuação em qualquer área.
* Morar no Brasil.
* Frequentam universidades presencialmente e podem acabar perdendo itens.

# Especificação do Projeto

Para que o problema, as necessidades e as especificações do projeto sejam compreendidas de maneira mais clara é essencial levar em consideração o ponto de vista do público-alvo do projeto. Com isso buscamos compreender os desafios enfrentados pelos usuários ao lidar com itens perdidos e encontrados, identificando os pontos cruciais a serem abordados neste projeto. Por fim, ao entender as necessidades dos usuários é possível propor as requisições funcionais, não funcionais e restrições associadas ao projeto, para o planejamento eficaz do desenvolvimento do projeto.

## Personas

|  |  |  |
| --- | --- | --- |
| **Matheus Filipe** | | |
|  | **Idade**: 19  **Ocupação**: Trabalha na empresa Group Software e é estudante de sistemas de informação na PUC Minas. | Aplicativos:   * Instagram * Youtube |
| Motivações   * Trabalhar como dev mobile * Aprender novas tecnologias | Frustrações   * Rotina de trabalho e faculdade simultaneamente. | Hobbies   * Jogar * Consumir conteúdo (vídeos, séries, filmes, livros) * Passar tempo com amigos |

|  |  |  |
| --- | --- | --- |
| **Júlia Almeida** | | |
|  | **Idade**: 18  **Ocupação**: Estudante de sistemas de informação na UFMG. | Aplicativos:   * Facebook * Instagram * Tiktok * Netflix |
| Motivações   * Perspectiva de carreira. * Condição econômica. | Frustrações   * Pressão acadêmica por bons resultados. * Conciliação de vida pessoal e vida acadêmica. | Hobbies, História   * Ler. * Assistir filmes e séries. * Encontrar com a família e amigos. * academia. |

|  |  |  |
| --- | --- | --- |
| **Andréa Ferreira** | | |
|  | **Idade**: 25  **Ocupação**: Estudante de letras na UFRJ e trabalha na área da educação como monitora e professora substituita. | Aplicativos:   * Facebook * Instagram * Whatsapp |
| Motivações   * Conhecer o mundo, * Ampliar e compartilhar conhecimento através da educação | Frustrações   * Desmotivação e falta de esperança na profissão. * Intolerância. | Hobbies   * Assistir filmes * Praia * Trabalhar a criatividade |

## Histórias de usuários

A partir da compreensão do dia a dia das personas identificadas para o projeto, foram registradas as seguintes histórias de usuários.

|  |  |  |
| --- | --- | --- |
| **Eu como … [PERSONA]** | **… quero/desejo …  [O QUE]** | **… para ....**  **[POR QUE]** |
| Matheus Filipe | quero poder registrar um item que encontrei rapidamente, fornecendo detalhes como descrição, local e data em que encontrei | para que o dono do item em questão consiga encontra-lo |
| Matheus Filipe | quero poder marcar um item como encontrado | para sinalizar que o item foi encontrado pelo dono. |
| Júlia Almeida | quero poder procurar entre os itens perdidos | para encontrar o meu item mais facilmente. |
| Júlia Almeida | quero poder pesquisar pelo nome de uma instituição | para encontrar mais facilmente o item no caso de eu saber onde perdi. |
| Júlia Almeida | quero poder filtrar pelo tipo de item (roupas, eletrônicos, documentos…) | para facilitar a achar o meu item perdido. |
| Andréa Ferreira | quero poder cadastrar facilmente uma instituição que quer registrar seus achados e perdidos | para que eu possa ter um achados e perdidos específico da instituição que eu frequento. |
| Andréa Ferreira | quero poder colocar descrição sobre a minha instituição | para informar os usuários sobre dados de contato, por exemplo. |

## Requisitos do Projeto

Levando em conta todas as histórias dos usuários é preciso deixar os requisitos dos projetos mais claros de maneira a definir requisitos funcionais (features, funcionalidades e especificações sobre como funcionará a aplicação), requisitos não funcionais (características do sistema além das funcionalidades, como segurança, desempenho e usabilidade) e restrições do projeto. Dessa forma o processo de desenvolvimento toma forma ao ter tarefas específicas e organizadas por prioridade.

### Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade** |
| RF-01 | O site deve ter a opção de cadastrar uma instituição/estabelecimento com nome, descrição… | Alta |
| RF-02 | O site deve ter a função de cadastrar o item perdido com nome, descrição… | Alta |
| RF-03 | O site deve ter uma tela listando itens perdidos em forma de cards que devem conter um nome e descrição breve da instituição/estabelecimento. | Alta |
| RF-04 | O site deve ter uma tela listando todas as instituições em forma de cards que devem conter um nome e descrição breve da instituição/estabelecimento. | Alta |
| RF-05 | O site deve ter a opção de marcar um item como encontrado. | Alta |
| RF-06 | O site deve ter uma barra de busca para procurar instituições. | Média |
| RF-07 | O site deve ter uma barra de busca para procurar itens perdidos. | Média |
| RF-08 | O site deve ter uma página de achados recentes. | Média |
| RF-09 | O site deve permitir editar as informações de uma instituição e de um item. | Média |
| RF-10 | O site deve ter uma página “Sobre nós” que vai contar um pouco sobre o propósito da equipe e sobre o projeto. | Baixa |
| RF-11 | O site deve ter a opção de colocar tags nos itens. | Baixa |
| RF-12 | O site deve ter a opção de filtrar os itens de acordo com sua respectiva tag. | Baixa |

### Requisitos não funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade** |
| RNF-01 | O site deve ser publicado em um ambiente acessível publicamente na Internet (Repl.it, GitHub Pages, Heroku); | Alta |
| RNF-02 | O site deverá ser responsivo permitindo a visualização em um celular de forma adequada | Alta |
| RNF-03 | O site deve ser compatível com os principais navegadores do mercado (Google Chrome, Firefox, Microsoft Edge) | Alta |
| RNF-04 | O site deve ter bom nível de contraste entre os elementos da tela em conformidade | Média |
| RNF-05 | O site deve ter uma interface do usuário intuitiva e fácil de usar, proporcionando uma experiência de usuário agradável e eficiente para pessoas com diferentes níveis de habilidade técnica. | Média |
| RNF-06 | O código deve possibilitar uma fácil manutenção do sistema, garantindo que atualizações e correções de bugs possam ser implementadas de forma rápida e sem interrupções significativas no serviço. | Média |

### Restrições

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RE-01 | O projeto deverá ser entregue no final do semestre letivo e as sprints devem ser feitas e entregues nos períodos designados a elas. |
| RE-02 | O desenvolvimento da aplicação deve restringir as tecnologias utilizadas apenas as aprendidas em sala, como HTML, CSS, Bootstrap e JavaScript. |
| RE-03 | A equipe não pode subcontratar o desenvolvimento do trabalho ou terceirizar qualquer parte do trabalho. |

# Metodologia

A metodologia contempla as definições de ferramental utilizado pela equipe tanto para a manutenção dos códigos e demais artefatos quanto para a organização do time na execução das tarefas do projeto.

## Relação de Ambientes de Trabalho

Os artefatos do projeto são desenvolvidos a partir de diversas plataformas e a relação dos ambientes com seu respectivo propósito é apresentada na tabela que se segue.

|  |  |  |
| --- | --- | --- |
| **Ambiente** | **Plataforma** | **Link de Acesso** |
| Repositório de código fonte | GitHub | <https://github.com/rommelcarneiro/tiaw-template> |
| Documentos do projeto | Google Drive | <https://docs.google.com/folder/d/1xE9t6zD78VnVkeOSgDfss33QWe85ogqYpx9x-tuG24> |
| Projeto de Interface e Wireframes | MarvelApp | <https://marvelapp.com/4hd6091> |
| Gerenciamento do Projeto | Trello | <https://trello.com/b/LupglzUz/tiaw-template> |

## Gestão de código fonte

Para gestão do código fonte do software desenvolvido pela equipe, o grupo utiliza um processo baseado no **Git Feature Branch Workflow**, mostrado na Figura a seguir. Desta forma, todas as manutenções no código são realizadas em branches separados. Uma explicação rápida sobre este processo é apresentada no site "[5 Git Workflows & Branching Strategy to deliver better code](https://zepel.io/blog/5-git-workflows-to-improve-development/)".

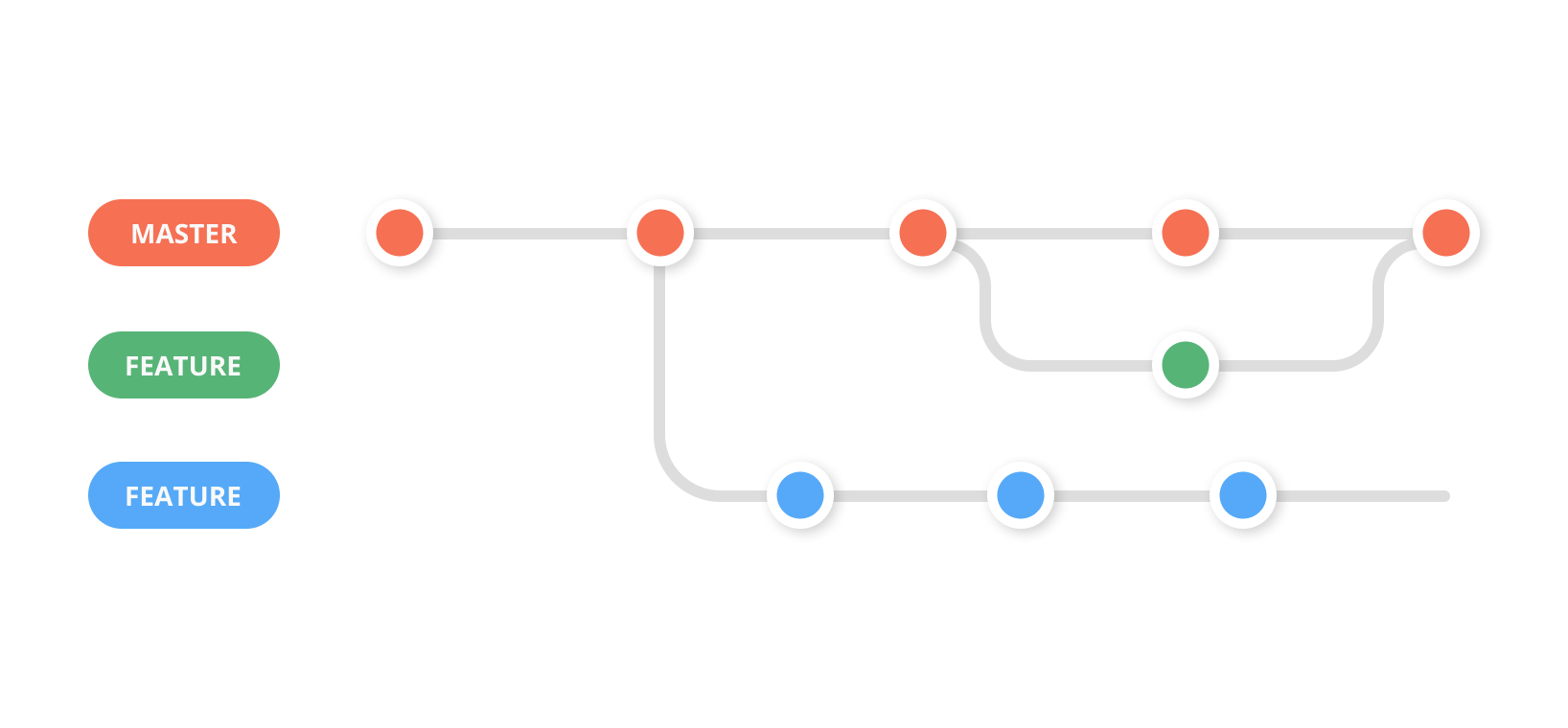


Figura X - Fluxo de controle do código fonte no repositório git

## Gerenciamento do Projeto

A equipe utiliza metodologias ágeis, tendo escolhido o Scrum como base para definição do processo de desenvolvimento.

A equipe está organizada da seguinte maneira:

* Scrum Master: Carlos Augusto
* Product Owner: Will Machado
* Equipe de Desenvolvimento
  + Henrique Silva (Desenvolvedor Front End)
  + ilo Amy Saldanha (Desenvolvedor Back End)
  + Roberto Felipe Dias (Analista de Negócios)
  + Rommel Carneiro (Desenvolvedor Front End)
  + Simone Nogueira (Designer)

Para organização e distribuição das tarefas do projeto, a equipe está utilizando o Trello estruturado com as seguintes listas:

* **Recursos**: esta lista mantém um template de tarefas recorrentes com as configurações padronizadas que todos devem seguir. O objetivo é permitir a cópia destes templates para agilizar a criação de novos cartões.
* **Backlog**: recebe as tarefas a serem trabalhadas e representa o Product Backlog. Todas as atividades identificadas no decorrer do projeto também devem ser incorporadas a esta lista.
* **To Do**: Esta lista representa o Sprint Backlog. Este é o Sprint atual que estamos trabalhando.
* **Doing**: Quando uma tarefa tiver sido iniciada, ela é movida para cá.
* **Test**: Checagem de Qualidade. Quando as tarefas são concluídas, eles são movidas para o “CQ”. No final da semana, eu revejo essa lista para garantir que tudo saiu perfeito.
* **Done**: nesta lista são colocadas as tarefas que passaram pelos testes e controle de qualidade e estão prontos para ser entregues ao usuário. Não há mais edições ou revisões necessárias, ele está agendado e pronto para a ação.
* **Locked**: Quando alguma coisa impede a conclusão da tarefa, ela é movida para esta lista juntamente com um comentário sobre o que está travando a tarefa.

O quadro kanban do grupo no Trello está disponível através da URL <https://trello.com/b/LupglzUz/tiaw-template> e é apresentado, no estado atual, na Figura X. A definição desta estrutura se baseou na proposta feita por Littlefield (2016).

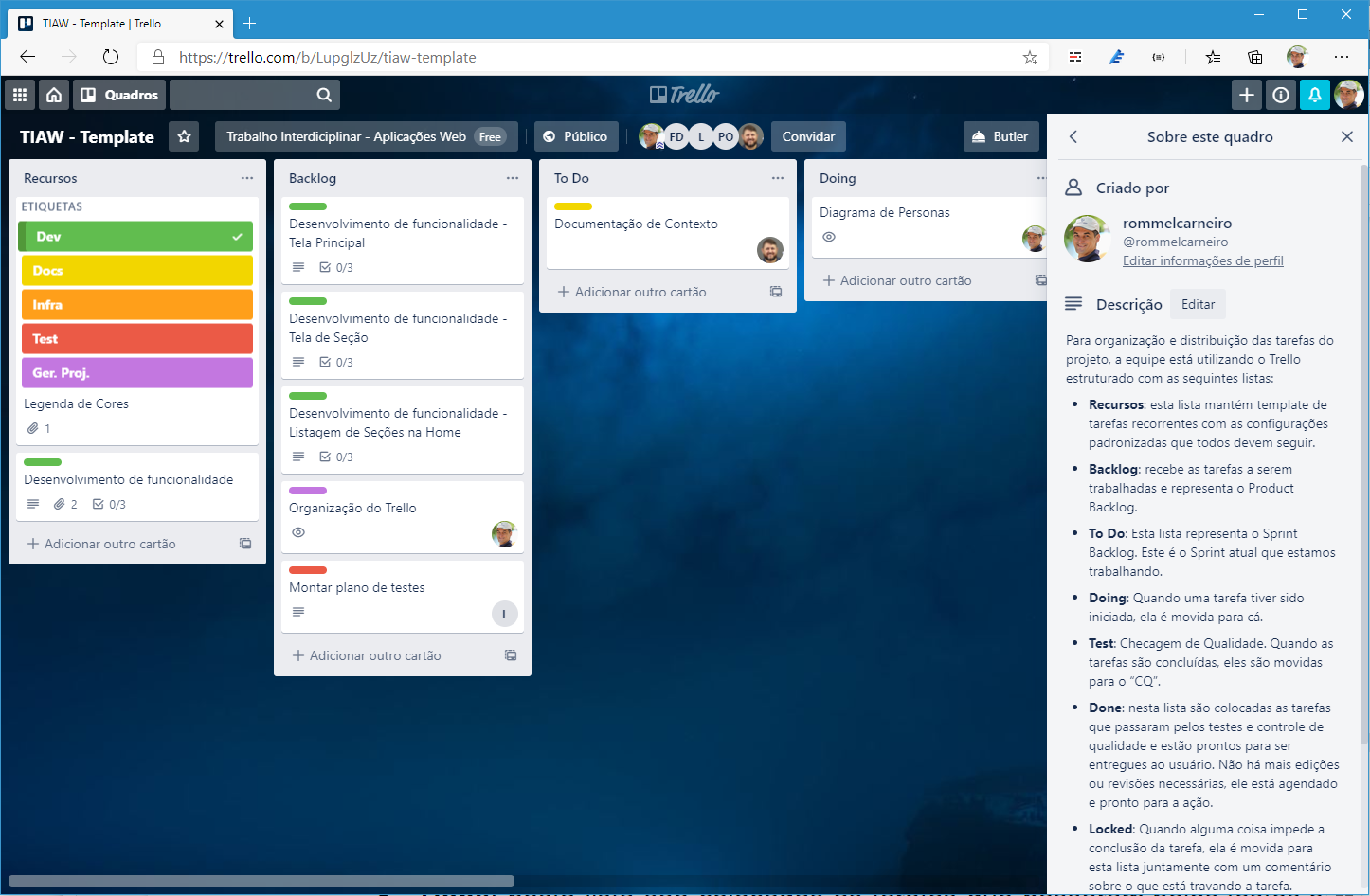
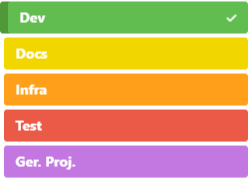


Figura X - Tela do Trello utilizada pelo grupo



A tarefas são, ainda, etiquetadas em função da natureza da atividade e seguem o seguinte esquema de cores/categorias:

* Documentação
* Desenvolvimento
* Infraestrutura
* Testes
* Gerência de Projetos.

# Projeto de Interface

Dentre as preocupações para a montagem da interface do sistema, estamos estabelecendo foco em questões como agilidade, acessibilidade e usabilidade. Desta forma, o projeto tem uma identidade visual padronizada em todas as telas que são projetadas para funcionamento em desktops e dispositivos móveis.

## Fluxo do Usuário

O diagrama apresentado na Figura X mostra o fluxo de interação do usuário pelas telas do sistema. Cada uma das telas deste fluxo é detalhada na seção de Wireframes que se segue. Para visualizar o Wireframe interativo, acesse o [prototipo no figma](https://www.figma.com/proto/yVosVg5piTvUuxNzJVHZLA/Untitled?type=design&t=exi3ic2JdvVbaY7b-1&scaling=scale-down&page-id=0%3A1&node-id=3-37&starting-point-node-id=3%3A37&mode=design).

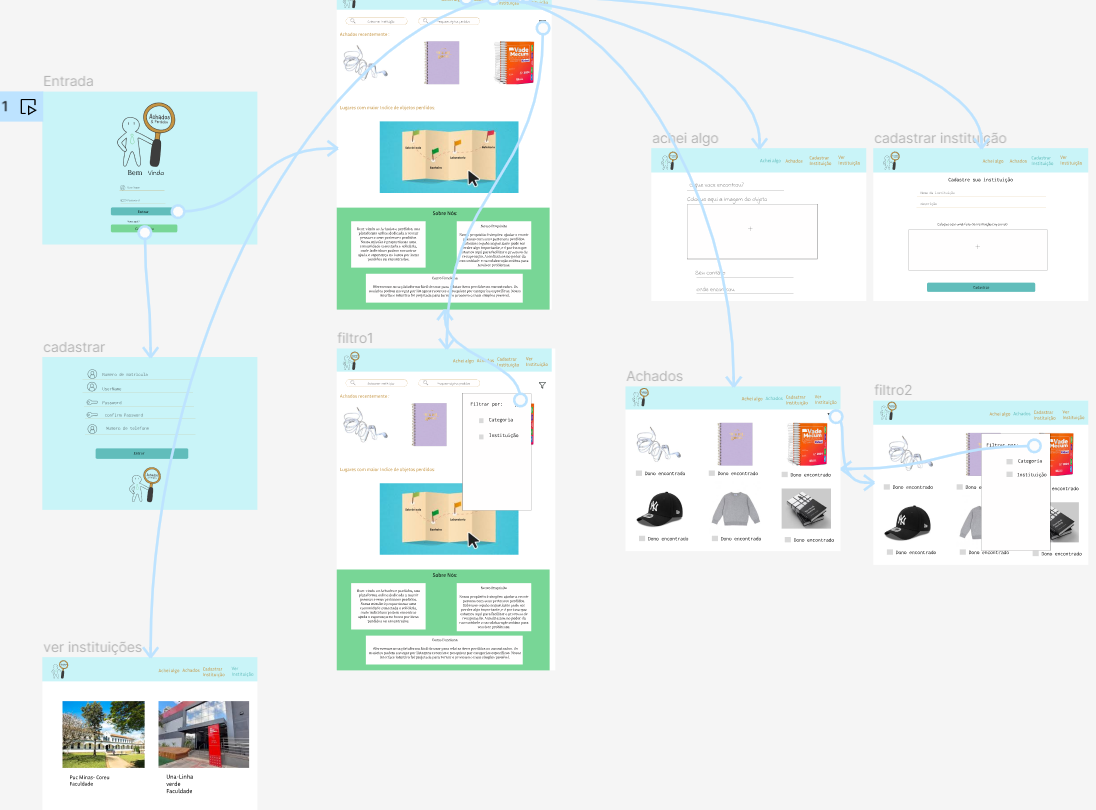


Figura X - Fluxo de telas do usuário

## Wireframes

Conforme fluxo de telas do projeto, apresentado no item anterior, as telas do sistema são apresentadas em detalhes nos itens que se seguem. As telas do sistema apresentam uma estrutura comum que é apresentada na Figura X. Nesta estrutura, existem 3 grandes blocos, descritos a seguir. São eles:

* **Cabeçalho -** local onde são dispostos elementos fixos de identidade (logo) e navegação principal do site (menu da aplicação);
* **Conteúdo -** apresenta o conteúdo da tela em questão;
* **Barra lateral** - apresenta os elementos de navegação secundária, geralmente associados aos elementos do bloco de conteúdo.

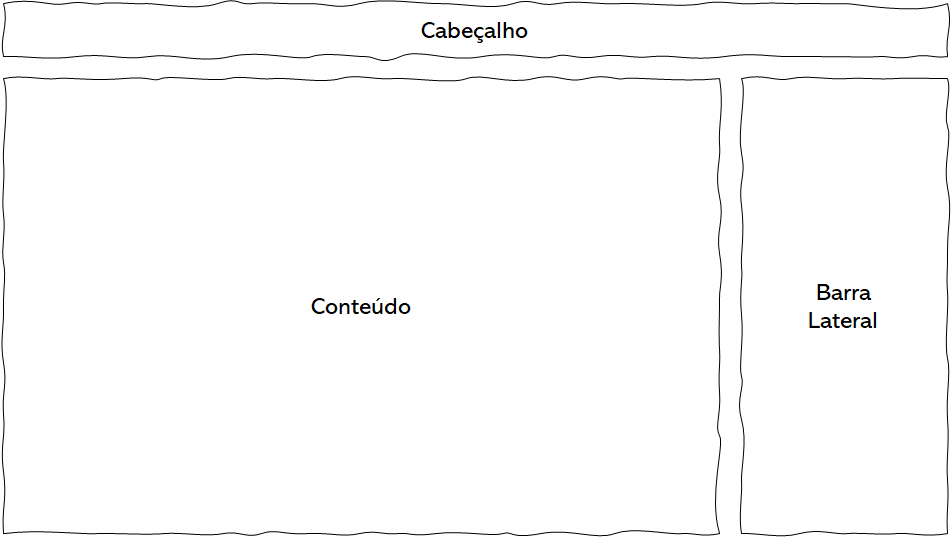
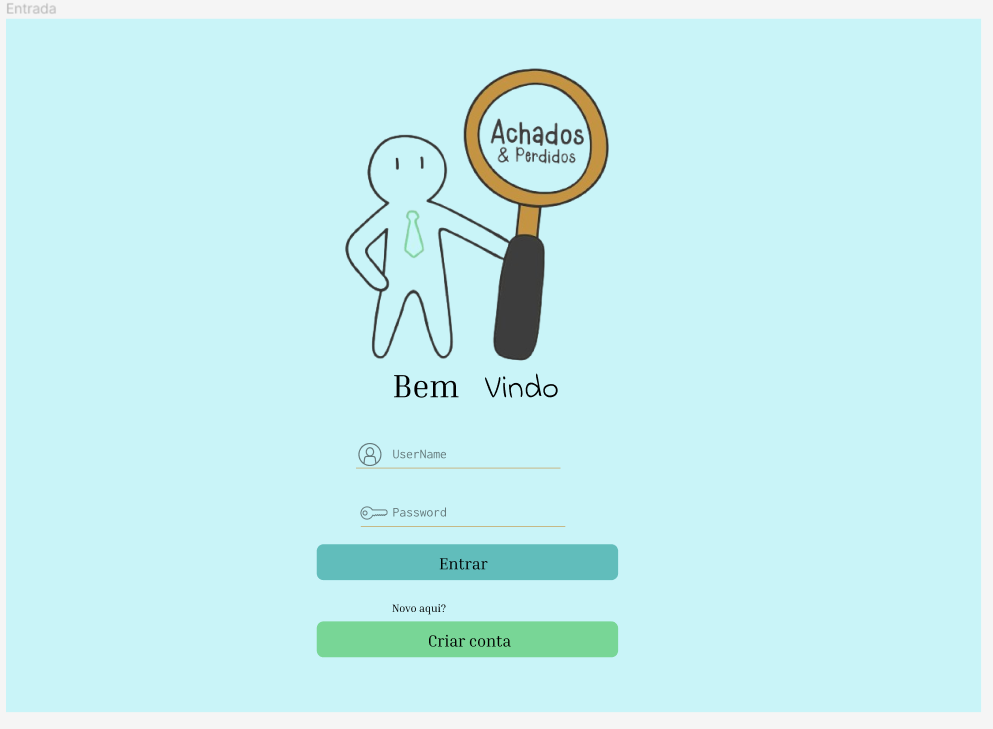


Figura X - Estrutura padrão do site

### Tela – Entrada

A tela de entrada é a primeira tela que o usuário vai ter acesso ao entrar no site, de um jeito bem colorido e interativo para trazer “curiosidade” ao cliente

* Componente de **login** que permite que o usuário acesse ao site, colocando nome e senha;
* Componente de **criar conta**, que permite ao usuário criar uma conta nova no site;



### Tela – Home page

A tela de home page é a que o usuário vai ter acesso ao realizar seu login no site

* Componente de **pesquisa** que permite substituir o conteúdo da página com o resultado da busca solicitada pelo usuário;
* Componente de **menu** que permite ao usuario navegar pelas páginas do site;
* Componente de **imagens** interativas e chamativas ao usuário



### Tela – Achei algo

Assim que o usuário encontra algum objeto perdido, ele entra no site e consegue cadastrar esse objeto

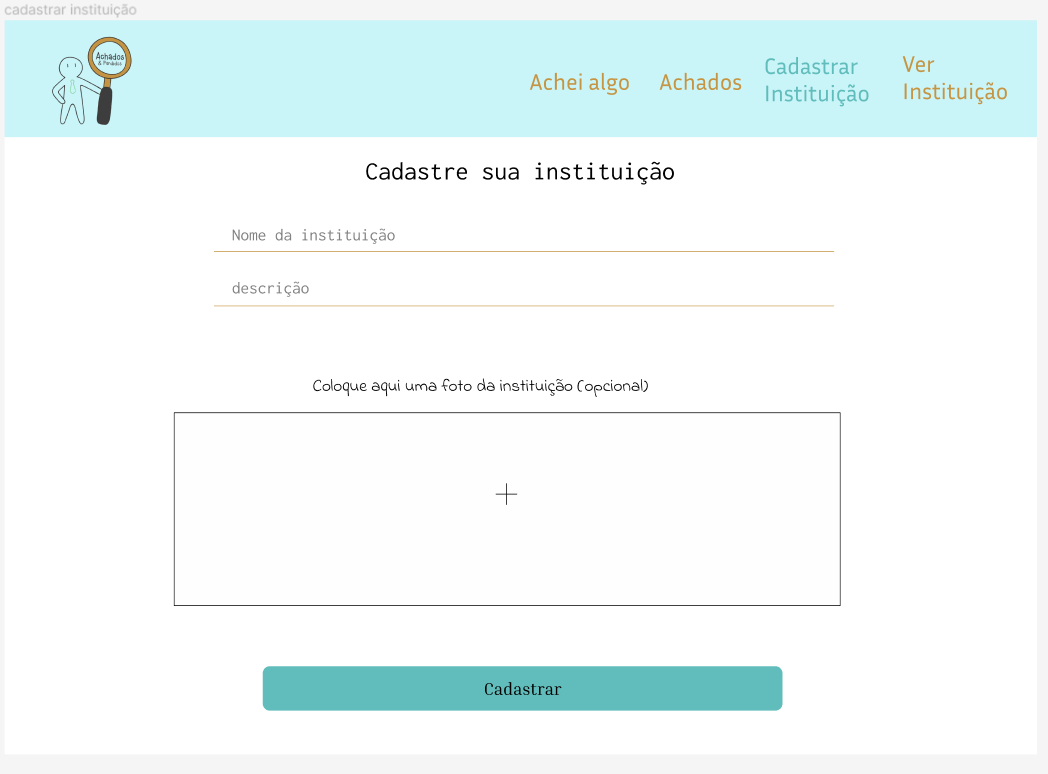
* Componente de **menu** igual da home page
* Componente de **campos** para colocar especificações do objeto
* Componente de **imagem** para colocar foto do objeto



### Tela – Cadastro da instituição

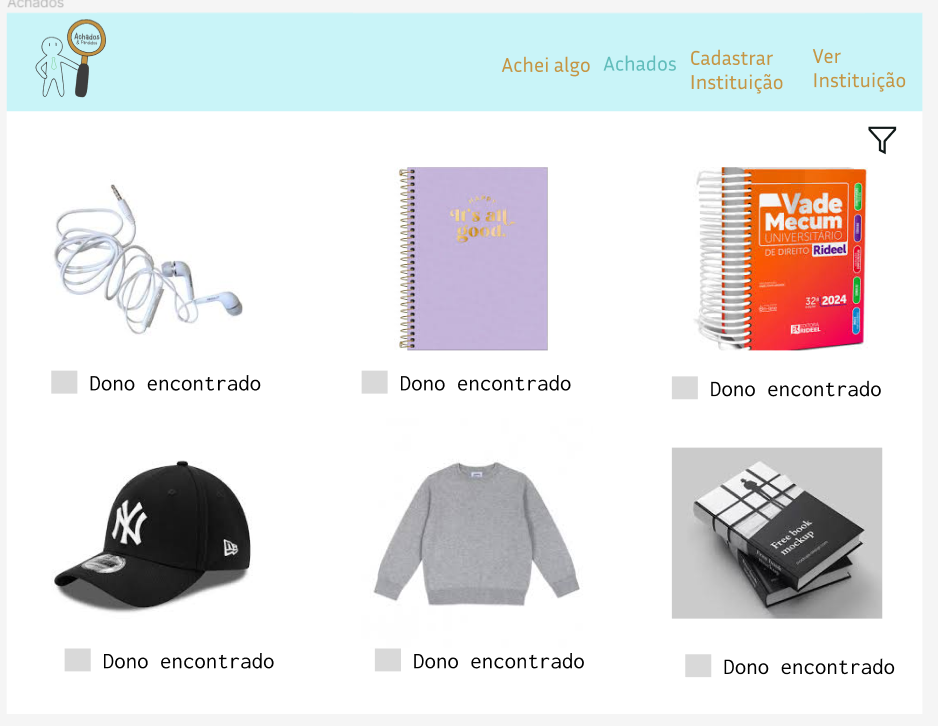
Nessa tela o usuário pode cadastrar uma instituição que quiser que tenha o programa de achados e perdidos

Com componentes iguais aos da página de achei algo;



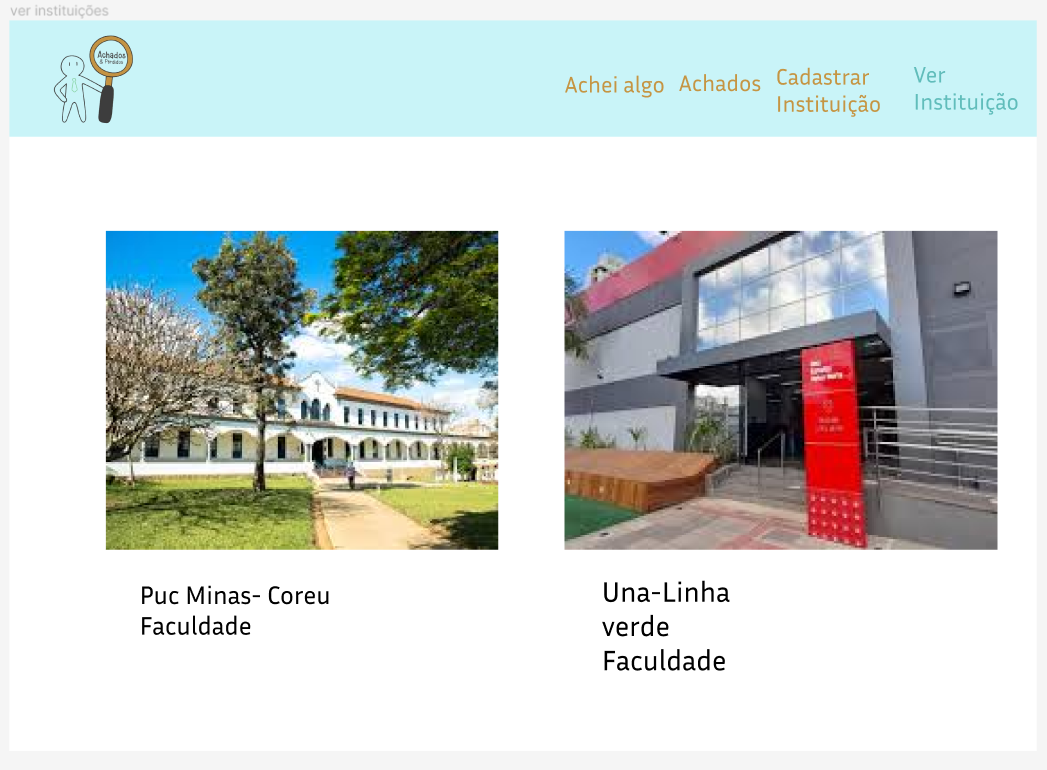
### Tela - Achados

A tela que permite que o usuário que perdeu algo consiga ver se o item que ele perdeu já está cadastrado no site para ele conseguir seu objeto de volta

* Componente de **marcação** se o usuário já tiver encontrado seu item, marcá-lo como encontrado

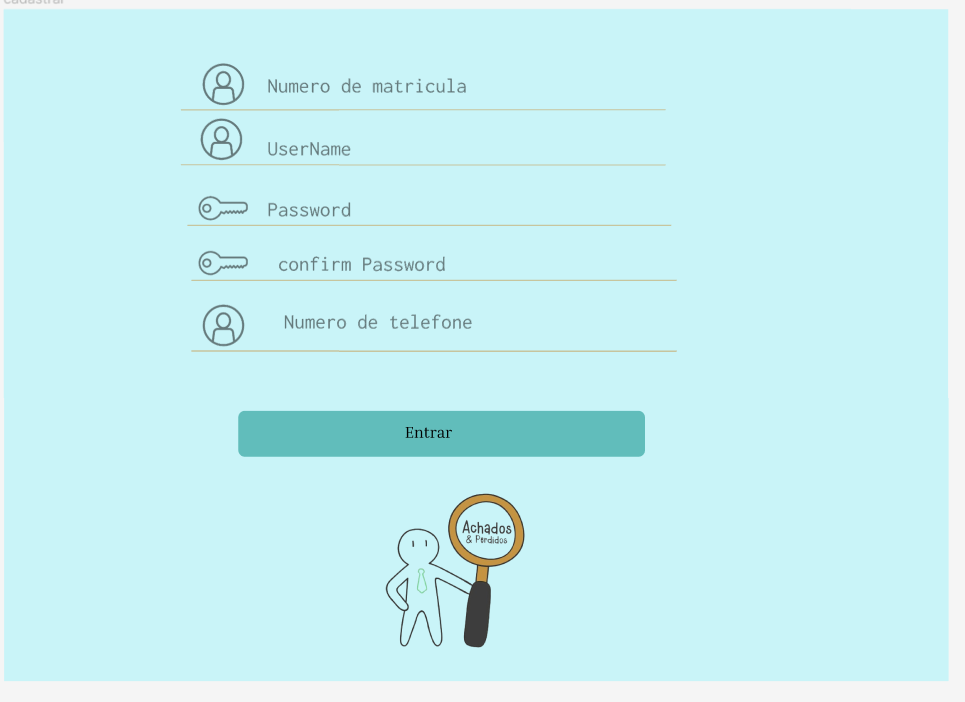
### Tela – Ver instituições

A tela de ver instituições permite ao usuário ver as instituições que foram cadastradas no site para usar o programa de achados e perdidos



### Tela – Cadastro

Na tela que permite ao usuário se cadastrar no site caso ele não tenha login e assim ter acesso a todas as outras páginas do site



# 

# Implementação da Solução

Nesta seção são apresentados os detalhes técnicos da solução criada pelo equipe, tratando da Arquitetura da Solução, as estruturas de dados e as telas já implementadas.

## Arquitetura da Solução

### Diagrama de Componentes

Os componentes que fazem parte da solução são apresentados na Figura que se segue.

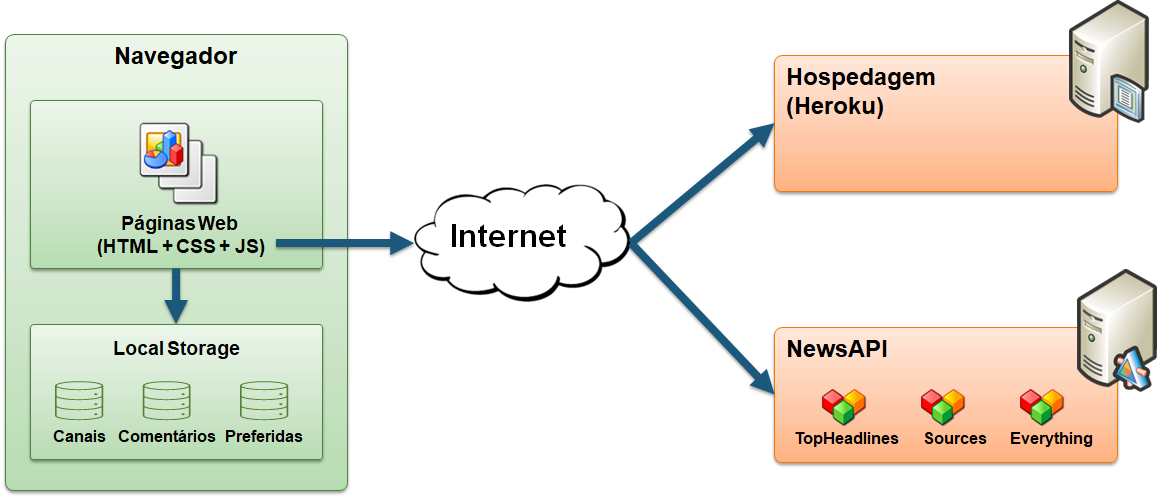


Figura X - Arquitetura da Solução

A solução implementada conta com os seguintes módulos:

* **Navegador** - Interface básica do sistema
  + **Páginas Web** - Conjunto de arquivos HTML, CSS, JavaScript e imagens que implementam as funcionalidades do sistema.
  + **Local Storage** - armazenamento mantido no Navegador, onde são implementados bancos de dados baseados em JSON. São eles:
    - **Canais** - seções de notícias apresentadas
    - **Comentários** - registro de opiniões dos usuários sobre as notícias
    - **Preferidas** - lista de notícias mantidas para leitura e acesso posterior
* **News API** - plataforma que permite o acesso às notícias exibidas no site.
* **Hospedagem** - local na Internet onde as páginas são mantidas e acessadas pelo navegador.

### Hospedagem

O site utiliza a plataforma do Heroku como ambiente de hospedagem do site do projeto. O site é mantido no ambiente da URL:

<https://link_exemplo.herokuapp.com>

A publicação do site no Heroku é feita por meio de uma submissão do projeto (push) via git para o repositório remoto que se encontra no endereço:

<https://git.heroku.com/link_exemplo.git>

## Template do Site

O padrão de layout a ser utilizado pelo site tem correspondência ao projeto de Interface elaborado anteriormente, conforme Figura X.

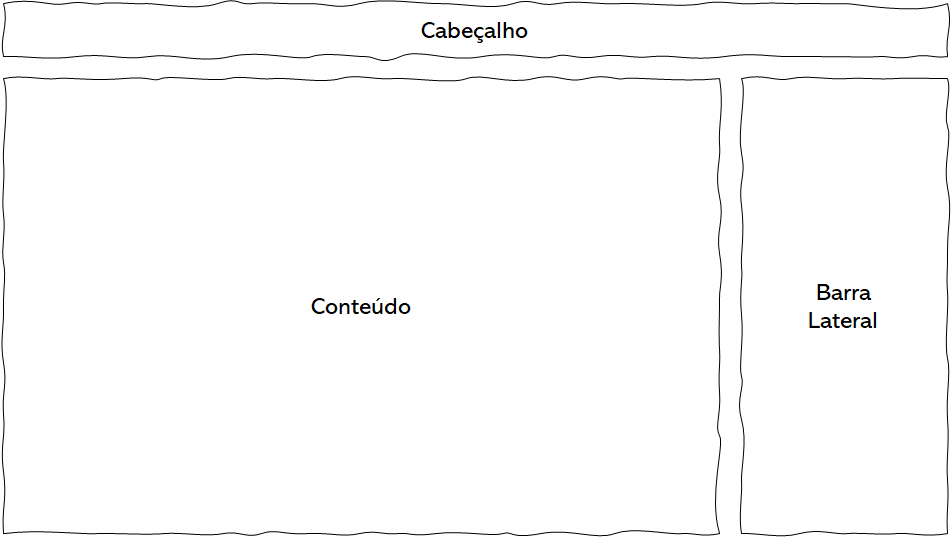


Figura X - Template padrão do site

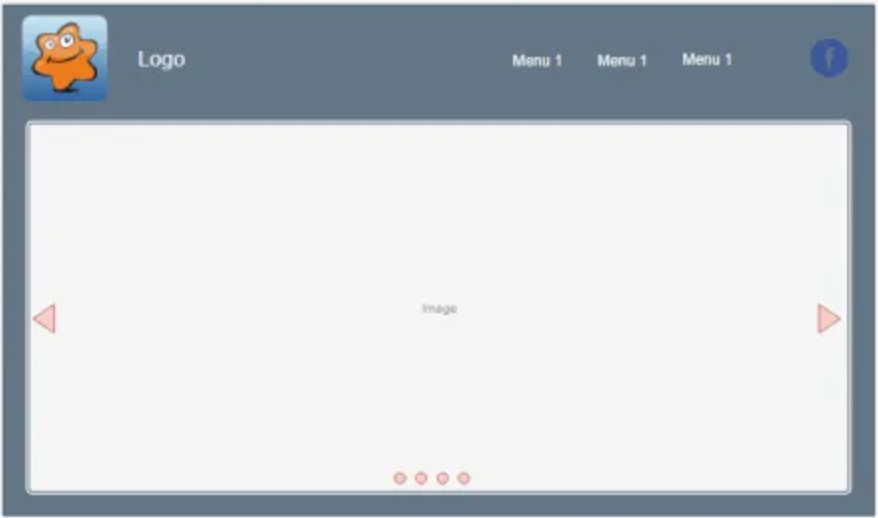
O template criado está disponível no site [http://github.com/…](http://github.com/%E2%80%A6).. e é composto pelos seguintes layouts:

* Tela principal
* Modal de mensagens
* Tela de visualização de notícia

A responsividade segue o padrão do Bootstrap

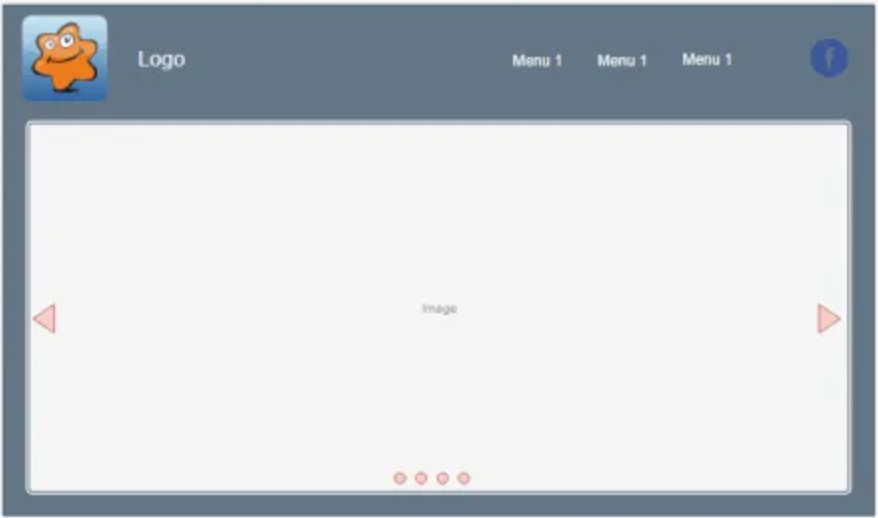
### Tela Principal

Tela que abrange todas as visualizações iniciais do site e a lista de notícias gerais.



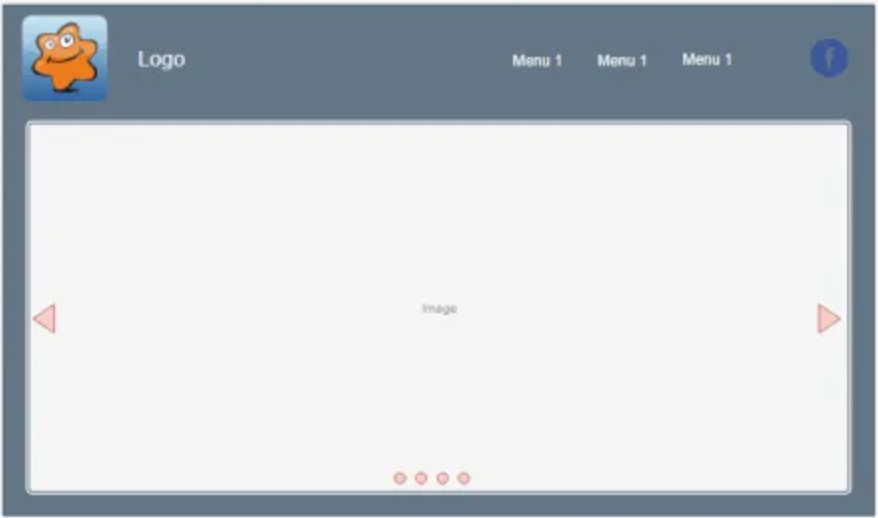
### Modal de Mensagens

Este layout é utilizado para exibição de mensagens do sistema nos diversos locais.



### Tela de Visualização de Notícias

Local onde são exibidas as notícias gerais e de cada uma das categorias.



## Funcionalidades do Sistema (Telas)

Nesta seção são apresentadas as telas desenvolvidas para cada uma das funcionalidades do sistema. O respectivo endereço (URL) e outras orientações de acesso são apresentadas na sequência.

### Entrega Sprint 1 - Visualização de Notícias (RF-01 e RF-02)

Alunos responsável: Danilo Henrique de Souza Riguette, Gustavo Inacio Jeronymo Oliveira

Igor Barroso Almeida, João Paulo Pimentel da Silva, Júlia Saquel Santana de Oliveira e

Pedro Henrique Florêncio da Silva

Link do código: http:///

A tela principal do sistema apresenta notícias de destaque nos principais canais. As notícias são armazenadas no LocalStorage com estruturas dados baseados em JSON. Um exemplo da tela é apresentada na Figura X.

### 

**Requisitos atendidos**

* RF-01 -
* RF-02 -

**Artefatos da funcionalidade**

* index.html
* script\_noticias.js
* style.cssicone\_noticia.jpg

**Estrutura de Dados**

{

"mensagens": [

{

"id": 1,

"titulo": "Trump estende ordem que proíbe Huawei",

"data": "14/05/2020",

"fonte": "Universo Online",

"autor": "Sam Rutherford",

"texto": "Em maio de 2019, o presidente dos Estados

Unidos, Donald Trump, emitiu uma ordem executiva  
 que proibiu várias empresas estrangeiras ..."

}

]

}

**Instruções de acesso**

1. Abra um navegador de Internet e informe a seguinte URL: <https://repl.it/Sf3l@#3ks>
2. A tela de notícias é a primeira funcionalidade exibida pelo aplicativo.

### Pesquisa de Notícias (RF-05)

A tela de pesquisa permite ao usuário informar um texto que será pesquisado na base de notícias e retornar todas as notícias relacionadas com o texto informado pelo usuário.

### 

**Artefatos da funcionalidade**

* pesquisa.html
* script\_pesquisa.js
* style.css
* icone\_lupa.jpg

**Instruções de acesso**

1. Faça o download do arquivo do projeto (ZIP) ou clone do projeto no GitHub;
2. Descompacte o arquivo em uma pasta específica;
3. Abra o Visual Studio Code e execute o Live Server;
4. Abra um navegador de Internet e informe a seguinte URL:  
   <http://localhost:5500/index.html>

# 

# Avaliação da Solução

O processo de realização dos testes da solução desenvolvida está documentado na seção que se segue e traz os planos de testes de software e de usabilidade, na sequência, o registro dos testes realizados.

## 6.1 Plano de Testes de Software

Requisitos para realização do teste:

* Site publicado na Internet
* Navegador da Internet - Chrome, Firefox ou Edge
* Conectividade de Internet para acesso às plataformas (APIs)

Os testes funcionais a serem realizados no aplicativo são descritos a seguir.

|  |  |
| --- | --- |
| **Caso de Teste** | **CT-01 - Visualizar notícias principais** |
| **Requisitos Associados** | RF-01 - O site deve apresentar na página principal notícias dinâmicas obtidas por meio de canais de notícias da Internet (API)  RF-02 - O site deve apresentar, para cada notícia, uma imagem correspondente ao assunto apresentado (*thumbnail*) |
| **Objetivo do Teste** | Verificar se a carga de notícias está acontecendo corretamente |
| **Passos** | 1) Acessar o Navegador  2) Informar o endereço do Site  3) Visualizar a página principal  4) Verificar o processo de carga |
| **Critérios de Êxito** | * Deve haver uma requisição AJAX no painel NETWORK das ferramentas do Desenvolvedor (recurso do Navegador). * As notícias devem ser exibidas corretamente no site, sendo necessárias pelo menos 3 notícias sendo apresentadas * As notícias devem trazer imagens visíveis associadas ao assunto da notícia |

|  |  |
| --- | --- |
| **Caso de Teste** | **CT-02 - Visualizar detalhes da notícia** |
| **Requisitos Associados** | RF-03 - O site deve permitir ao usuário visualizar o texto completo da notícia com todos os detalhes da publicação |
| **Objetivo do Teste** | Verificar se as notícias estão sendo apresentadas completamente na página de leitura da notícia |
| **Passos** | 1) Acessar o Navegador  2) Informar o endereço do Site  3) Visualizar a página principal  4) Clicar em uma notícia |
| **Critérios de Êxito** | A página deve apresentar o título da notícia, imagem de destaque da notícia, data da notícia, autor da notícia e o texto completo da notícia |

**--------------- IMPORTANTE ---------------**  
Coloque todos os casos de testes planejados para o projeto.

**----------------------------------------------------**

## 6.2 Registro de Testes de Software

Os resultados obtidos nos testes de software realizados são descritos na Tabela a seguir.

## 

## 6.3 Plano de Testes de Usabilidade

O planejamento dos testes de usabilidade a serem realizados com usuários são descritos na Tabela a seguir.

* Identificação de objetivos do teste de usabilidade
* Detalhes do público envolvido nos testes
* Roteiro detalhado das tarefas que foram desempenhadas pelos usuários

## 6.4 Registro de Testes de Usabilidade

Os resultados obtidos nos testes de usabilidade realizados são descritos na Tabela a seguir.

* Relatório com detalhamento das pessoas envolvidas no teste (indicar claramente o número de usuários que testaram o sistema)
* Relatório com registro dos testes feitos: fotos, prints de telas, relatos dos usuários
* Relatório de problemas identificados nos testes e propostas de correções/ajustes

# Referências

A lista a seguir traz as referências utilizadas nesse trabalho. são elas:

* Bakshy, E.; Messing, S.; Adamic, L. A. **Exposure to ideologically diverse news and opinion on Facebook**. Science. 2015.
* Littlefield, A. **Guia da metodologia ágil e scrum para iniciantes**. 2016. Disponível em: <https://blog.trello.com/br/scrum-metodologia-agil>. Acessado em 26/05/2020.
* Newman, N.; Fletcher, R.; Kalogeropoulos, A.; Nielsen; R. K. **Reuters Institute Digital News Report 2019**. Reuters Institute for the Study of Journalism. 2019. Disponível em: <https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2019-06/DNR_2019_FINAL_0.pdf>. Acessado em 26/05/2020.