

UNIVERSIDADE FEDERAL DA GRANDE DOURADOS

ANTONIO NETO DIONISIO
PAULO HENRIQUE NASCIMENTO DE SOUZA

***CODE FOR PANTANAL: DESENVOLVIMENTO DE TECNOLOGIAS CÍVICAS
PARA O MATO GROSSO DO SUL***

DOURADOS – MS

2021

ANTONIO NETO DIONISIO
PAULO HENRIQUE NASCIMENTO DE SOUZA

***CODE FOR PANTANAL: DESENVOLVIMENTO DE TECNOLOGIAS CÍVICAS
PARA O MATO GROSSO DO SUL***

Trabalho de Conclusão de curso de graduação apresentado para obtenção do título de bacharel em Sistemas de Informação, Faculdade de Ciências Exatas e Tecnologia da Universidade Federal da Grande Dourados.

Orientador: Prof. Msc. Felipe José Carbone

Coorientador: Luiz Augusto Volpi
Nascimento

DOURADOS – MS

2021

ANTONIO NETO DIONISIO
PAULO HENRIQUE NASCIMENTO DE SOUZA

**CODE FOR PANTANAL: DESENVOLVIMENTO DE TECNOLOGIAS CÍVICAS
PARA O MATO GROSSO DO SUL**

Trabalho de Conclusão de Curso aprovado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação na Universidade Federal da Grande Dourados, pela comissão formada por:

Orientador: Prof. Msc. Felipe José Carbone
UFGD – FACET

Coorientador: Luiz Augusto Volpi Nascimento

Prof. Dra. Valguima Victoria Viana Aguiar Odakura
UFGD – FACET

Prof. Dr. Vanderson Hafemann Fragal
UFGD – FACET

Dourados, 12 de Maio de 2021

RESUMO

Na contemporaneidade, as tecnologias digitais e o acesso à internet fazem parte da vida da maioria das pessoas. A expansão da internet aliada às novas ferramentas de desenvolvimento em programação tem permitido o desenvolvimento comunitário de aplicações para resolução dos mais variados problemas urbanos. Este trabalho descreve a implementação de uma plataforma web para divulgação de aplicações baseadas em tecnologias cívicas, por meio de desenvolvimento comunitário open-source em Mato Grosso do Sul, intitulado projeto “Code for Pantanal”. Tal plataforma está sendo concebida através da utilização de ferramentas de código aberto, sendo que no *back-end* é com uma API NodeJS utilizando o *framework* ExpressJS que está interligando banco de dados em MongoDB, reportando JSON para *front-end* em ReactJS. Dentre as funcionalidades desenvolvidas estão: i) visualização de projetos cadastrados, ii) solicitação para participação em projeto, iii) indicação à novos projetos, iv) resumo da documentação de projetos, v) informações sobre a equipe, vi) material didático, abrangendo tutoriais e videoaulas sobre projetos, compartilhamento de código-fonte e primeiros passos na programação. O projeto, além de educativo à formação em tecnologia dos autores, mostra-se de grande contribuição para a sociedade, já que há a possibilidade deste projeto encurtar a relação entre comunidade e entidades municipais, acelerando a resposta de solução dos problemas urbanos.

Palavras-chave: Tecnologias Cívicas, *Open Source*, Problemas Urbanos.

ABSTRACT

Nowadays, digital technologies and internet access are part of most people's lives. The expansion of the internet combined with new development tools in programming has allowed the community development of applications to solve the most varied urban problems. This work describes the implementation of a web platform for the dissemination of applications based on civic technologies, through open-source community development in Mato Grosso do Sul, entitled "Code for Pantanal" project. Such a platform is being conceived through the use of open source tools, and the back-end is with a NodeJS API using the ExpressJS framework that is linking the database in MongoDB, reporting Json to the front-end in ReactJS. Among the functionalities developed are: i) visualization of registered projects, ii) request to participate in a project, iii) indication for new projects, iv) summary of project documentation, v) information about the team, vi) didactic material, including tutorials and video lessons on projects, sharing source code and first steps in programming. The project, in addition to educating the authors' training in technology, proves to be of great contribution to society, since there is the possibility of this project shortening the relationship between the community and municipal entities, accelerating the response to solve urban problems.

Keywords: Civic Technologies, Open Source, Urban Problems.

LISTA DE FIGURAS

Figura 1 – Ilustração funcionamento NodeJS (Esquerdo) e web tradicional (direito). Fonte: Opus, 2021.....	16
Figura 2 – Arquitetura plataforma Web <i>Code For</i> Pantanal.	18
Figura 3 – Diagramas de casos de usos proposto inicialmente para a plataforma <i>Code for</i> Pantanal.	20
Figura 4 – Diagramas de classes proposto para plataforma Web <i>Code for</i> Pantanal, com atributos obrigatórios (+) e não obrigatórios (-) e métodos (\).	22
Figura 5 – Tela inicial da plataforma	23
Figura 6 – Tela de Login	24
Figura 7 – Tela de cadastro de usuário/colaborador.....	25
Figura 8 – Tela de cadastro de Sugestão	26
Figura 9 – Protótipo de <i>Dashboard</i> do usuário.	27
Figura 10 – Motivos para utilizar o <i>Code for</i> Pantanal	29

SUMÁRIO

1. INTRODUÇÃO.....	7
2. REVISÃO DE LITERATURA	9
2.1. Tecnologias cívicas no desenvolvimento de cidades	9
2.2. Desenvolvimento ‘ <i>OPEN SOURCE</i> ’: Programação colaborativa.....	11
2.3. Trabalhos Relacionados: O ‘ <i>Code for</i> ’ pelo Mundo	12
2.4. Tecnologias de desenvolvimento de software no <i>back-end</i> e <i>front-end</i>	14
2.4.1. ReactJS	14
2.4.2. NodeJS.....	15
2.4.3. MongoDB	17
3. MODELAGEM DO SISTEMA	17
3.1. Arquitetura.....	17
3.2. Descrição dos atores (casos de uso do sistema)	18
3.3. Diagramas de Casos de Uso	19
3.4. Diagramas de Classes	20
4. RESULTADOS E DISCUSSÃO	23
4.1. Tela Inicial.....	23
4.2. Tela de Login.....	24
4.3. Tela de Cadastro de Pessoas	24
4.4. Tela de cadastro de Sugestão.....	26
4.5. Tela de acompanhamento do status da Sugestão.....	27
4.6. Proposta de módulo de cursos e videos-tutoriais de projetos.....	27
5. CONSIDERAÇÕES FINAIS	29
REFERÊNCIAS BIBLIOGRÁFICAS	32

1. INTRODUÇÃO

O efeito da concentração da população na zona urbana vem sendo demonstrado nas últimas décadas. Em 1950, apenas 30% da população mundial residia no âmbito urbano, já em 2018 o valor registrado foi de 55% e a perspectiva para 2050 é de que ao menos dois terços (68%) da população estejam concentradas na cidade (ONU, 2020). A grande concentração de pessoas nestes locais, bem como ampliação das cidades e vasta extensão traz consigo inúmeros desafios para ordenamento urbano, monitoramento e gerenciamento dos serviços públicos. Neste cenário torna-se cada vez mais importante o encurtamento da tripla ligação “sociedade-tecnologia-administração pública”, de modo a poder melhorar a qualidade de vida (em âmbito social) da população.

Entre os problemas comumente observados nas cidades tem-se: focos de *Aedes Aegypti* em vias públicas (PACHECO et al., 2019), iluminação pública: postes ligados durante o dia ou desligados (lâmpadas queimadas) no período noturno (CLEMENTE et al., 2018), dificuldades no uso e acesso ao transporte público (SILVA et al., 2019), e problemas regionalizados na pavimentação pública (PEQUENO & ELIAS, 2020) e trânsito em geral. Além dos problemas relacionados à manutenção de locais públicos, alguns autores apontam a utilização de ferramentas tecnológicas para encurtar o laço com as entidades governamentais. Gastil & Meinrath (2018), mencionam problemas de acessibilidade que dificultam a participação cívica nas tomadas de decisões governamentais, os autores defendem a criação de um portal de comunicação na qual a comunidade possa contribuir diretamente e enviar sugestões. Para Boella et al. (2019), a população está cada vez mais conectada à internet, através do uso dos *smartphones* e mídias sociais, o que possibilita a construção de projetos que combinem informações geográficas voluntárias (VGI), seguindo uma abordagem de design participativo em contribuição com a administração pública.

Neste cenário, dois termos vêm sendo difundidos em meio a literatura nos últimos anos buscando aprimorar os serviços em utilização na área urbana. As chamadas “Cidades Inteligentes” (*Smart cities*), sendo conceituadas como a utilização de tecnologias da informação como meio para prestar de forma mais eficiente os serviços urbanos e melhorar a qualidade de vida dos cidadãos (BARACHO, 2020; CUNHA et al. 2016; CUNHA et al., 2017) e as tecnologias cívicas (*Civic Technologies*), que partem da iniciativa das cidades inteligentes, atribuindo as propriedades dos movimentos de código livre (*open source*), visando a tecnologia à serviço da população (ROSA et al., 2014). Isso tudo mostra que é de grande potencial para

sociedade e, também, para a comunidade envolvida com o desenvolvimento de tecnologias, a busca de novas tecnologias cívicas fomentadas pela sua população, contribuindo com o crescimento do meio urbano e encurtando laços com entidades governamentais.

Este trabalho tem como objetivo geral, desenvolver e documentar uma plataforma de gerenciamento e divulgação de projetos de tecnologias cívicas no estado de Mato Grosso do Sul. Deste modo, busca expandir o projeto intitulado *Code for Pantanal* para o desenvolvimento de tecnologias cívicas. Constitui-se em sua base como uma derivação do projeto *Code for America* e, também, do *Code for Curitiba*, ambos de similar segmento, objetivando a implementação de aplicações *open source* em busca de criar soluções à problemas cívicos, partindo da participação de qualquer pessoa que queira contribuir com sua construção ou melhoria e de forma gratuita.

Pretendemos, com esse projeto, o “*Code for Pantanal*”, gerar presteza significativa para comunidade em geral, por meio das aplicações cívicas desenvolvidas e, também, como uma plataforma educativa, já que é, de certa forma, uma possibilidade para o aprendizado prático na área relativa dos interessados na contribuição dos projetos propostos.

Para objetivos específicos, determinou-se:

- i) Desenvolver uma plataforma web, utilizando NodeJS, com MongoDB (*back-end*) e ReactJS (*front-end*), para participação colaborativa no desenvolvimento de aplicações cívicas;
- ii) Possibilitar que seja encurtado o laço comunidade e entidades de interesse social, na busca de solução de problemas urbanos nas cidades de Mato Grosso do Sul.

A primeira parte deste trabalho apresenta uma revisão de literatura, abrangendo a utilização das tecnologias cívicas no desenvolvimento das cidades, a descrição do movimento ‘*Code for*’ e as defenições relativas as tecnologias utilizadas na aplicação. Em sequência, iremos descrever a metodologia no desenvolvimento da plataforma.

2. REVISÃO DE LITERATURA

Neste capítulo, será apresentado uma revisão de literatura das principais tecnologias empregadas no desenvolvimento da plataforma *Web Code for Pantanal*, do contexto de origem do movimento ‘*Code*’ pelo mundo e do uso de tecnologias cívicas no avanço das chamadas cidades inteligentes (*Smart cities*).

2.1. Tecnologias cívicas no desenvolvimento de cidades

O avanço em velocidade de internet, especialmente da cobertura 4G e, mais recentemente e por enquanto menos difundido, o 5G, tem potencializado a utilização de aplicações tecnológicas de modo geral, mas especialmente no ambiente urbano, no sentido de busca para melhorias urbanas. Nas últimas duas décadas, um grande número de inovações da administração pública vem sendo impulsionadas por tecnologias digitais, de forma a permitir a participação do cidadão para melhorias dos sistemas de gestão governamental.

Os relatos na literatura demonstram o potencial do uso de aplicações tecnológicas no desenvolvimento de diferentes setores em ambiente urbano. Junior et al. (2016) citam as possibilidades de utilização da tecnologia digital para participação da população, tais como democracia eletrônica, atendimento ao cidadão, votação online, legislação participativa. Os mesmos autores, descrevem a possibilidade do que chamam de ‘Economia colaborativa’, referindo-se a iniciativas que utilizam tecnologias digitais para promover modelos socioeconômicos colaborativos que permitam o compartilhamento de habilidades, conhecimento, alimentação, vestuário e habitação. Olivier et al. (2018) desenvolveram a aplicação ‘*App Movement*’, na qual o objetivo central é o envio de propostas de melhorias e contagem de apoiadores, para fomento da administração pública. Asad & Dantec (2017) descrevem o desenvolvimento de uma aplicação visando ampliar a segurança no ciclismo urbano e a aplicação de projetos, infraestrutura e políticas que incentivem as pessoas a adotar a bicicleta como forma alternativa de transporte. Essa iniciativa tenta integrar com segurança o tráfego de bicicletas ao tráfego de veículos por meio de projetos de engenharia civil e de tráfego.

Aplicações para melhorias no sistema de transporte também vêm sendo desenvolvidas de modo a melhorar a experiência do cidadão. Neto & Amorim (2019) descrevem o

desenvolvimento de um modelo de aplicativo para celulares com rotas e disponibilidade de horários na Cidade de São José dos Campos – SP, Brasil. Outro aplicativo de mesma finalidade foi desenvolvido por Calandre et al. (2018), especificamente para cidade de Curitiba – PR, Brasil. Ambos os autores relatam que a utilização de aplicativos com tal finalidade tornam o transporte público coletivo atrativo aos não usuários. Percebe-se que algumas aplicações desenvolvidas para uma cidade específica, mostram-se de potencial para utilização em forma generalizada em outras regiões. Outras funcionalidades de aplicações relatadas em literatura são: identificação de focos de dengue (MACHADO et al., 2017), identificação de pontos de coleta de resíduos sólidos (STOTHMANN et al., 2018), registro de lâmpadas queimadas de iluminação pública (ARAUJO et al., 2018), inclusão de pessoas com deficiência em cidades inteligentes (MACIEL et al., 2018), registro de histórico de cidades e de pontos turísticos (MASSONI et al., 2017), entre outras.

Para envolver a participação comunitária as aplicações desenvolvidas precisam de design atrativo e facilidade de utilização. Os projetistas devem ter o entendimento da interação-humano-computador (IHC), de modo a poder aplicar o entendimento real do uso das aplicações pela população. Corbertt (2018), avaliando a importância da IHC para subsidiar a participação da população no uso de tecnologias cívicas, verificaram que ferramentas de *feedback* de usuários e aplicação de interfaces gráficas para divulgação de projetos elevaram a confiabilidade na participação comunitária no uso e desenvolvimento de tecnologias cívicas.

O público jovem é um grande aliado na utilização de ferramentas de administração pública. Cho et al. (2020) relatam que além das vantagens à administração pública, existem vantagens relacionadas ao fator social, na qual os autores citam que: i) O engajamento cívico por adolescentes os expõe à questões cívicas em uma idade precoce e contribui para um senso de fortalecimento sócio-político; ii) Jovens que se envolvem em política participativa digital são muito mais propensos a envolver-se na política offline "real", como votação. Os jovens também têm motivação própria na participação no desenvolvimento de projetos de tecnologias cívicas. Silva et al. (2020), analisando as motivações para contribuição de alunos em um projeto de programação de verão denominado ‘*Google Summer of Code*’, concluíram que, majoritariamente, os alunos participam para obter experiência de trabalho e capacidade de aprimorar seus currículos, além do contato com grandes profissionais da área de programação.

Deste modo, vislumbra vantagens em âmbito econômico-social, a sociedade que absorve do uso da tecnologia cívica, justificando a busca e entendimento das possibilidades pela participação em programação de projetos de código fonte aberto, o chamado ‘*open source*’.

2.2. Desenvolvimento ‘OPEN SOURCE’: Programação colaborativa

O termo *open source* é usado para caracterizar todo software cujo código-fonte é aberto para qualquer pessoa. Isso significa que terceiros podem analisar, auditar e em alguns casos até fazer modificações para atender suas necessidades ou realizar melhorias (PERENS et al., 1999; REDHAT, 2017). O *open source* não deve ser confundido com o Software Livre, que apresenta características semelhantes: basicamente todo software livre precisa ser de código aberto, mas o contrário nem sempre é verdade (NATAHOUSE, 2020).

Considera-se que o movimento *open source* surgiu em meados de 1983, a partir do movimento software livre iniciado por Richard Stallman, que na época trabalhava no laboratório de inteligência artificial do MIT (REDHAT, 2017). Este, propôs uma alternativa gratuita ao sistema operacional Unix, que na época era um sistema proprietário da AT&T. Nesta ideia surgiu o GNU (*GNU's not Unix*), sistema operacional concebido por Stallman, sob a filosofia do software livre. Uma década depois surgiram as primeiras versões do Kernel Linux, aumentando o acesso às tecnologias provenientes dos sistemas operacionais (NATAHOUSE, 2020).

Feller et al. (2020) estabelece algumas diretrizes para projetos *open source*: o fornecedor deve manter a integridade do código-fonte do autor, fornecendo '*patches*' que fazem alterações quando o software é 'construído' na máquina do usuário, em vez de modificá-lo antes da distribuição; que não deve haver discriminação contra pessoas ou grupos de pessoas; e que o usuário deve ser capaz de modificar o código-fonte e compartilhar essas modificações com o autor e com outros.

A cultura colaborativa facilita o desenvolvimento das sociedades, e faz com que seja possível desenvolver novos produtos e iniciativas em maior velocidade, com menos chance de erros e que alcançam um número maior de pessoas. Alguns softwares famosos foram concebidos nessa “cultura” e são mantidos assim, por exemplo, o navegador de internet Mozilla Firefox foi construído em formato *open source* (KHOMH et al., 2012), assim como o Gimp - editor de imagens e Audacity - editor de arquivos de áudio (JAMES, 2009). Um exemplo de plataforma bem atual que também é um repositório de código aberto é o *SourceForge*, que contém mais de 500 mil projetos e milhões de usuários registrados (SOURCEFORGE, 2020).

No cenário de pandemia do covid-19 no momento em que se escreve este trabalho, as funcionalidades *open source* se mostram importantes na divulgação de dados e em questões de

segurança. Javaheri et al. (2020) descreve a utilização da aplicação *open source* CovidCTNet que analisa imagens de tomografia e diferencia o Covid-19 da pneumonia adquirida ou outras doenças pulmonares. Também foram desenvolvidas plataformas de registros de dados com modelos de previsão de surto e avanço da covid-19 em diversas cidades (ABDULRAHMAN et al., 2020; MOHAMADOU et al., 2020).

A pandemia também forçou o fechamento de escolas e unidades de ensino públicas. Visto a necessidade de reformulação no ambiente remoto de ensino, impulsionou-se o desenvolvimento de aplicações *open source* para o ensino, a exemplo de: plataforma de aulas e avaliação no ensino de graduação (DUAN & LEE, 2020), aplicação para ensino e resolução de atividades didáticas de matemática e ciência (DEMIR & SHARMA, 2020), aprendizagem móvel como formas e métodos para aumentar a eficácia da educação e ensino (RAELOVICH et al. 2020; SATTAROV et al., 2020), registro eletrônico de saúde de código aberto para o ensino de apoio à decisão em Informática Médica (MEDLOCK et al, 2020).

Nesse sentido, acredita-se que por estar em aberto, a qualidade do software e processos envolvidos tendem a ser melhores, ou pelo menos mais amplos ou de maior alcance. Isso pode ser embasado também de acordo com Cio, 2019:

“Sem dúvida, as mais entusiasmantes inovações tecnológicas que ocorrem atualmente estão a tomar forma nas comunidades *open source*. Grandes desenvolvimentos em *big data*, automação, *machine learning* e inteligência artificial são resultado de equipes inteligentes e distribuídas globalmente que partilham recursos, estabelecem ligações pouco convencionais e se desafiam a fazer avançar o estado da arte”. (CIO, 2019).

Como percebe-se, o *open source* pode ser empregado nas mais diversas áreas do conhecimento e na busca de solução para problemáticas da sociedade. Este fato, tem impulsionado o surgimento de entidades no mundo para busca de soluções locais, com destaque ao *Code for America* e suas derivações, assunto do próximo tópico.

2.3. Trabalhos Relacionados: O ‘Code for’ pelo Mundo

Como relatado até aqui, a evolução da internet e das ferramentas de programação fomentaram a busca de soluções tecnológicas para problemas comunitários. Neste ponto, houve o surgimento de organizações não governamentais para desenvolvimento de aplicações de código aberto visando melhorias para serviços de administração pública. O *Code for America*, por exemplo, surgiu em 2009 nos Estados Unidos, reunindo profissionais de tecnologia cívica

e design, atendendo e desenvolvendo projetos com soluções em trabalho com governo das cidades (CODE FOR AMERICA, 2020). Entre os projetos desenvolvidos por esta comunidade estão: i) Adote um Hidrante – aplicativo para que voluntários da cidade de Boston pudessem se inscrever para “adotar” (retirar e gerir) hidrantes após tempestades, ii) Honolulu Answers – aplicativo de perguntas e respostas e iii) página GetCalFresh.org – desenvolvida para acelerar inscrição em programa de assistência nutricional na Califórnia (CODE FOR AMERICA, 2020).

Há várias outras iniciativas pelo mundo: *Code for All* (codeforall.org), que indexa vários outros; *Code for Canada*; *Code for Japan*, etc. Outra comunidade em destaque em âmbito mundial é a *Public Lab*, que foi fundada em 2010 e é uma organização sem fins lucrativos voltada ao uso de tecnologias em questões socioambientais identificadas pela própria comunidade. Entre os projetos desenvolvidos está o *Urban greening* in Nottingham, uma aplicação de mapeamento do ecossistema urbano e manutenção de arborização (PUBLIC LAB, 2020). A nível de Brasil, tem-se o *Code for Curitiba*, comunidade com grupo de desenvolvedores brasileiros. Destaca-se do *Code for Curitiba* a aplicação de registro de translocamento de ônibus no transporte público e o mais recente sistema de atualização de dados epidemiológicos da Covid-19 (CODE FOR CURITIBA, 2020). Enfim, há várias iniciativas em torno do mesmo objetivo: utilizar tecnologia *open source* para garantir boa governança.

Cabe fazer menção aqui à real importância do desenvolvimento *open source* na atualidade. Assim, percebemos que manter o código aberto e facilmente acessível, trará resultados mais amplos, podendo ser muito mais abrangente no momento de utilização como aplicação social de fato. Mesmo com as comunidades já existentes (citadas ou não), percebe-se a necessidade da difusão do trabalho na aplicação de tecnologias cívicas na região de Mato Grosso do Sul, a fim de contribuir com soluções relatadas para comunidade local. Além disto, baseando-se nas comunidades existentes, percebe-se a necessidade de ferramentas nos canais dos projetos na qual incentivem a comunidade ao desenvolvimento, tais como: Fórum para discussão e dúvidas, arquitetura base dos projetos a serem desenvolvidos, informações iniciais sobre a linguagem em utilização, tutoriais de configuração do ambiente de programação e informações sobre compartilhamento do código-fonte. Observação similar foi descrita por Knutas et al. (2019) em avaliação de processos de desenvolvimento do *Code for Ireland*, na qual os autores descrevem que as limitações de desenvolvimento compreenderam a escassez de ferramentas para envolvimento e incentivo a participação, além de melhorias no processo para obtenção de requisitos.

Portanto, considerando o aspecto social, as eventuais melhorias advindas dos projetos a serem desenvolvidos, bem como o aprimoramento dos modelos e a formação prática de todos

aqueles que vierem a contribuir com o projeto, este trabalho também tem como finalidade a proposta de impulsionamento de uma rede comunitária de programação de tecnológicas cívicas para o Mato Grosso do Sul, que aqui denominamos como ‘*CODE FOR PANTANAL*’.

2.4. Tecnologias de desenvolvimento de software no *back-end* e *front-end*

Na rotina de programação vários termos são utilizados no decorrer do desenvolvimento de um projeto. Entre os termos mais difundidos estão: *back-end* e *front-end*, que se referem basicamente às duas partes que compõem uma aplicação, ou nas quais ela pode ser dividida.

Entende-se *front-end* como sendo a parte da aplicação na qual o usuário pode interagir diretamente com a interface, também denominado lado do cliente, ou *client-side*. Assim, em uma aplicação web, por exemplo, seria a parte do código do sistema que é interpretado pelo navegador (Rocha et al., 2019). Para tal, cita-se como exemplos, as linguagens HTML, CSS, JavaScript, ReactJS, comumente utilizados na etapa de desenvolvimento em *front-end*.

Em outra face, desenvolve-se o *back-end*, conhecido também como lado do servidor, ou *server-side*. Nele é implementado toda a regra de negócios do sistema, além de operações de acesso ao banco de dados e tratamento de questões relacionadas à segurança (Rocha et al., 2019). Como no *front-end*, existe uma infinidade de linguagens de programação disponíveis, Java, JavaScript, Python, bem como *frameworks* relacionados, NodeJS, entre outros.

Visto a infinidade das linguagens e *frameworks* utilizados em desenvolvimento de software, as próximas subseções serão focadas em uma sucinta abordagem das tecnologias utilizadas no desenvolvimento deste trabalho, sendo: ReactJS (*front-end*) e NodeJS e MongoDB (*back-end*).

2.4.1. ReactJS

React é uma biblioteca de interface do usuário, *open source*, desenvolvida pela empresa Facebook, para facilitar a criação de arquivos interativos, com estado e componentes reutilizáveis. O fundamento básico por trás do React é o conceito de *Document Object Model*

(Modelo de Objeto de Documento) – DOM Virtual. Ele é uma API que nos permite acessar e manipular documentos HTML e XML válidos. O ReactJS usa efetivamente o DOM virtual, que pode ser renderizado tanto no lado do cliente ou no lado do servidor e se comunicar entre si. As subárvores de renderização do DOM Virtual de nós tem base em mudanças de estado. A API permite menor quantidade de manipulação DOM possível em ordem para manter seus componentes atualizados. React, além de ser leve para desenvolvimento, é preenchido com o mínimo de condições e elimina a necessidade de usar elementos extras, como *plugins*, facilitando o desenvolvimento *front-end* (KUMAR et al., 2016).

2.4.2. NodeJS

Também chamado de Node - é um ambiente JavaScript do lado do servidor (consulte <http://NodeJS.org>). É baseado no “V8”, o interpretador JavaScript de alto desempenho e de código aberto do Google. V8 e Node são implementados principalmente em C e C++, com foco em desempenho e pouca memória de consumo. Mas, enquanto o V8 suporta principalmente JavaScript no navegador (principalmente o Google Chrome), o Node tem como objetivo oferecer suporte a longa duração dos processos do servidor (CANTELON et al., 2017). Ao contrário da maioria dos outros ambientes modernos, um processo Node não depende de *multithreading* para apoiar a execução simultânea de processos lógicos, sendo baseado em um modelo de eventos de E/S assíncrono. No modelo NodeJS, apenas uma *thread* é responsável por tratar as requisições. Essa *thread* é chamada de *Event Loop*, e leva esse nome pois cada requisição é tratada como um evento. O *Event Loop* fica em execução esperando novos eventos para tratar, e para cada requisição, um novo evento é criado (TILKOV & VINOSKI, 2010).

Em um servidor web utilizando linguagens tradicionais, para cada requisição recebida é criada uma nova *thread* para tratá-la, demandando recursos computacionais (memória RAM, por exemplo). Uma vez que esses recursos são limitados, as *threads* não serão criadas infinitamente, e quando esse limite for atingido, as novas requisições terão que esperar a liberação desses recursos alocados para serem tratadas. Apesar de ser *single-threaded*, no NodeJS é possível tratar requisições concorrentes em um servidor NodeJS. Enquanto o servidor tradicional utiliza o sistema *multi-thread* para tratar requisições concorrentes, o NodeJS consegue o mesmo efeito através de chamadas de E/S (entrada e saída) não-bloqueantes. Isso significa que as operações de entrada e saída (ex: acesso a banco de dados e leitura de arquivos

do sistema) são assíncronas e não bloqueiam a *thread*. Diferentemente dos servidores tradicionais, a *thread* não fica esperando que essas operações sejam concluídas para continuar sua execução (OPUS, 2020).

A Figura 1, a seguir, representa a diferença de funcionamento de um servidor web tradicional e um NodeJS.

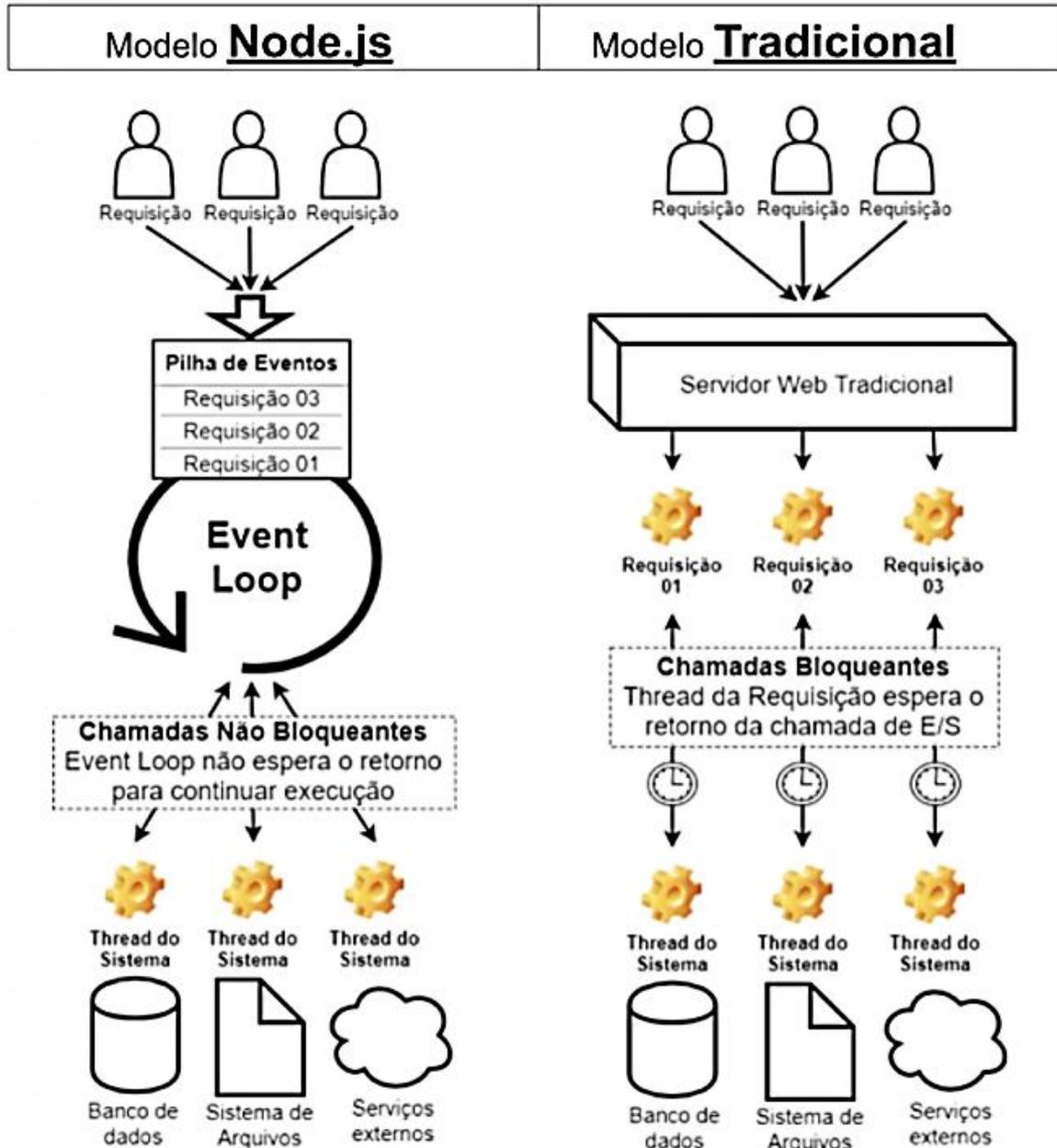


Figura 1 – Ilustração funcionamento NodeJS (Esquerdo) e web tradicional (direito). Fonte: Opus, 2021.

2.4.3. MongoDB

MongoDB é um banco de dados NoSQL, ou seja, não adota esquema baseado em modelo relacional para representar seus dados, sendo feito armazenamento em documentos flexíveis semelhantes ao JSON facilitando o trabalho com os dados. Ele é *open source*, possui uma vasta documentação, esquema flexível e fácil escalabilidade horizontal. O MongoDB adiciona preenchimento dinâmico a documentos e pré-aloca arquivos de dados para negociar o uso de espaço extra para desempenho consistente. É desenvolvido pela MongoDB *Inc.* e publicado sob uma combinação da GNU Affero General Public License e Licença Apache (MONGODB, 2021).

Suas características permitem com que, nas aplicações, sejam modeladas informações de modo muito mais natural, pois os dados podem ser aninhados em hierarquias complexas e continuar a ser indexáveis e fáceis de buscar.

3. MODELAGEM DO SISTEMA

Nesta seção estão descritos os requisitos do sistema, o diagrama de casos de uso, demonstração da prototipação do sistema e demonstração do desenvolvimento do *back-end* e *front-end* da aplicação.

3.1. Arquitetura

A arquitetura da plataforma Web *Code For Pantanal* (Figura 2), possui o *front-end* (ReactJS), o MongoDB como banco de dados, conectados bidirecionalmente com o *back-end* (NodeJS) implementado em rede local (Localhost), na qual posteriormente será direcionado à plataforma de hospedagem Web.

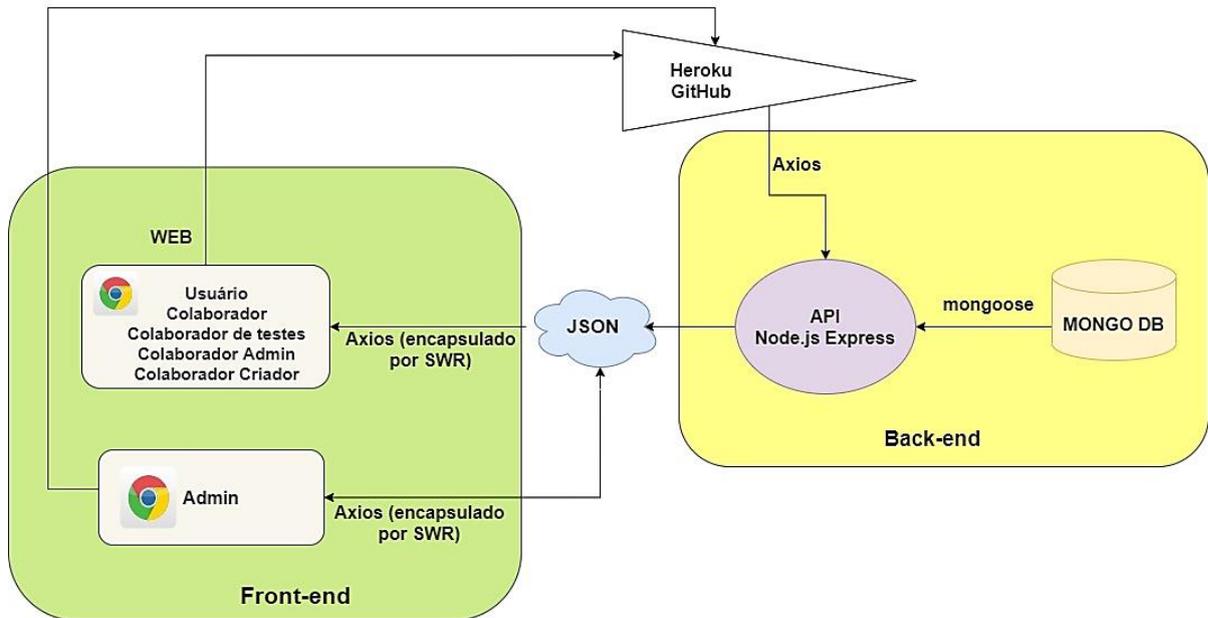


Figura 2 – Arquitetura plataforma Web Code For Pantanal.

Quando o usuário executa um evento na plataforma Web, o ReactJS através da biblioteca do Axios (um cliente HTTP) lida com a primeira interação enviando uma requisição HTTP REST ao NodeJS. O Node por sua vez irá tratar a requisição realizando uma operação com o banco de dados MongoDB (através do framework de modelagem Mongoose) ou buscando um arquivo no servidor devolvendo a resposta ao usuário por meio do ReactJS.

3.2. Descrição dos atores (casos de uso do sistema)

Abaixo segue uma descrição sucinta dos atores que terão ações específicas no sistema:

- **Admin:** são os responsáveis diretamente pela plataforma, com acesso total. Poderão cadastrar cursos/vídeos e outros materiais, também poderão cadastrar novos projetos (neste caso com aprovação direta, sem passar pela avaliação de outras pessoas).
- **Visitante:** este ator é qualquer pessoa na web com o link de acesso à plataforma. Poderá visualizar a página web, os cursos e materiais oferecidos bem como algumas informações sobre projetos que estão cadastrados na plataforma.
- **Usuário:** é qualquer pessoa que tenha interesse em criar conta na plataforma para poder: cadastrar sugestões, acompanhar progressos da sua sugestão ou de outros projetos aprovados, ver detalhes específicos sobre projetos em andamento ou finalizados, poder

relatar os testes de homologação quando seu projeto estiver em fase de conclusão. Um Usuário pode se tornar um colaborador em dois casos: por escolha própria, podendo escolher ser um colaborador desenvolvedor ou colaborador de testes, ou quando cadastrar uma proposta de projeto, neste caso o sistema irá determiná-lo como colaborador criador. Pode-se fazer uma analogia do ator Usuário do *Code* com um cliente que se cadastrou num sistema de vendas só para ter mais detalhes sobre um produto e/ou promoção e eventualmente efetuar compra, mas não detém privilégios relativos a colaboradores.

- Colaborador desenvolvedor: pode ser qualquer pessoa que tenha interesse em contribuir com os projetos cadastrados na plataforma.
- Colaborador de testes: qualquer pessoa que tenha interesse em se cadastrar especificamente para fazer testes nas aplicações que forem para fase de testes.
- Colaborador Gerente: é o colaborador que é responsável direto por um projeto. Este deverá alimentar a plataforma com detalhes específicos de um projeto.
- Colaborador Criador: é o Usuário que criou um projeto na plataforma, porém só poderá participar dos testes de protótipos ou de homologação.

3.3. Diagramas de Casos de Uso

A Figura 3 ilustra o diagrama de casos de uso geral para aplicação web do *Code for Pantanal*.

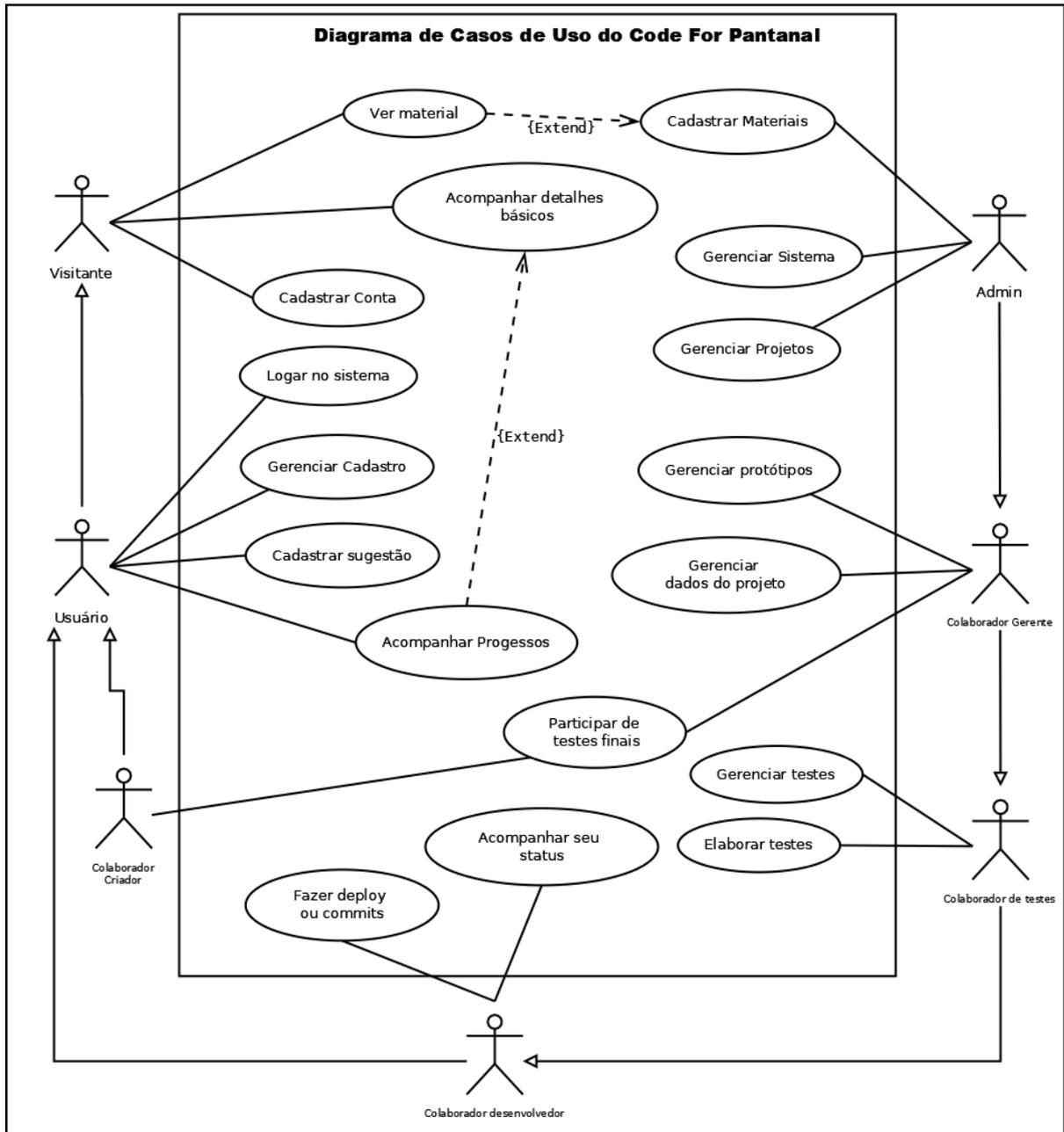


Figura 3 – Diagramas de casos de usos proposto inicialmente para a plataforma *Code for Pantanal*.

3.4. Diagramas de Classes

O Diagrama de classes está elucidado na Figura 4 a seguir. O acesso ao sistema será concedido inicialmente por meio de tela de login, que efetuará busca na tabela de pessoas retornando se os dados estão cadastrados e qual o nível de permissão de telas. Como a relação de projetos e pessoas é n:n (uma ou n pessoas podem estar relacionadas até n projetos e vice-versa), a classe 'PessoasObjetos' faz união do projeto e define a função delimitada a cada

usuário naquele projeto. Todos os indivíduos do sistema estão registrados na classe Pessoas, que herda informações para usuários com função de: Admin, Usuário, Colaborador.

Após o login, os usuários do sistema terão acesso a página que apresentará informações da classe Cursos, com opções de tutoriais e cursos introdutórios à programação, como forma de incentivo à contribuição e a programação. Esta classe foi adicionada ao longo da evolução do sistema, visto que, como estudantes que somos, sabemos da necessidade de haver cada vez mais treinamentos relativos a uma tecnologia a qual estamos dispostos a aprender. Entre as dificuldades que nós nos deparamos durante o decorrer deste curso ou de nossa experiência pessoal, antes e durante esta graduação, está o fato de que um dos pontos que leva as pessoas a não contribuir em projetos *open source* é a falta de informações sobre padronização, programação, dificuldade no entendimento das regras de negócio e estruturação de projeto, entre outros, que podem ser amenizados com o fornecimento de materiais didáticos e/ou de informações necessárias e suficientes para a conclusão devida de um projeto.

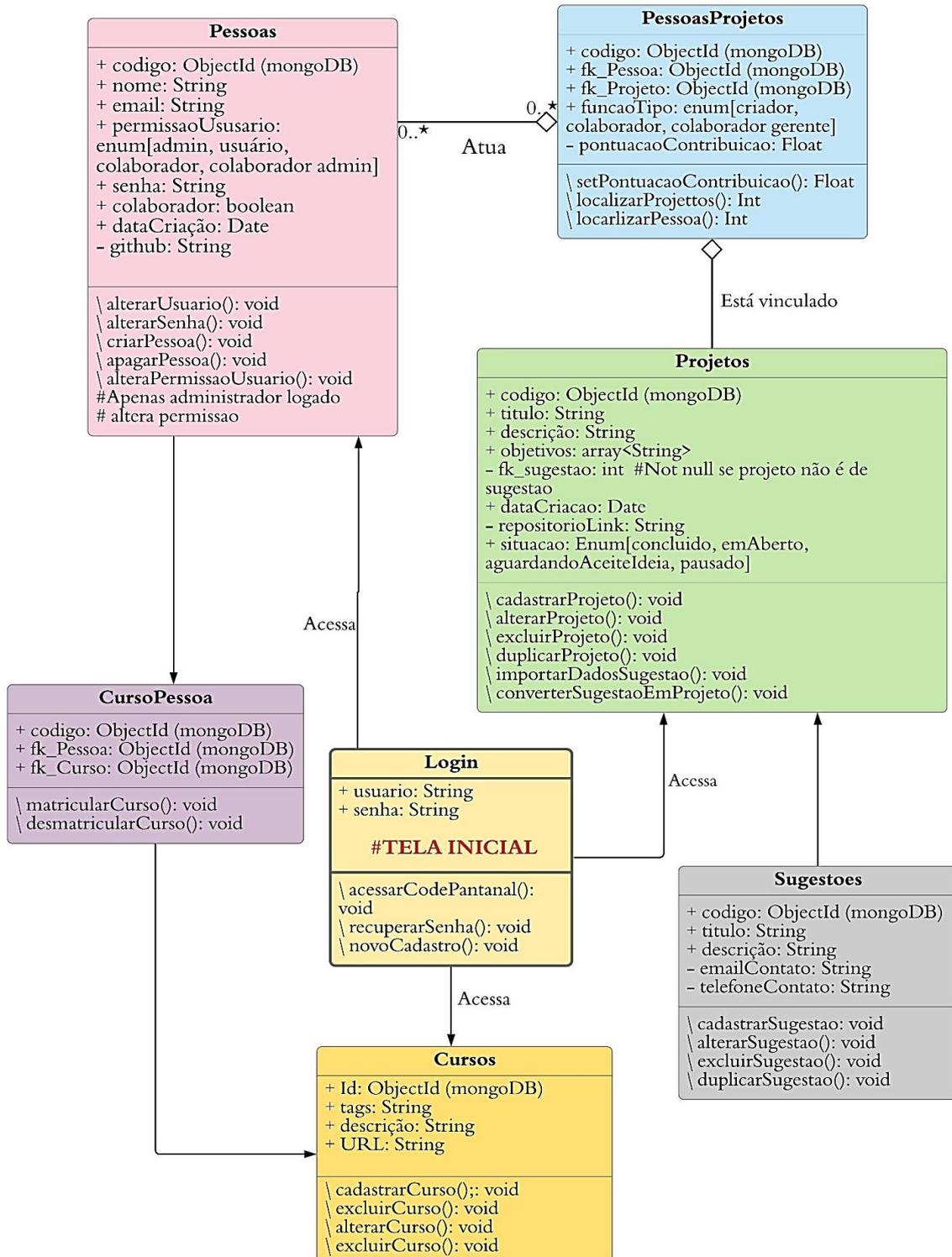


Figura 4 – Diagramas de classes proposto para plataforma Web *Code for Pantanal*, com atributos obrigatórios (+) e não obrigatórios (-) e métodos (\).

4. RESULTADOS E DISCUSSÃO

A implementação da plataforma Web *Code For* Pantanal teve seu início em Janeiro de 2021. Inicialmente, foi desenvolvido o protótipo escopo referente às telas principais (cadastros de projetos e usuários). Posteriormente, com o surgimento de novos requisitos, efetuou-se a adição de tela de cadastro de ideias pela comunidade e projetou-se a tela de cursos e aprendizado de programação. A sequência de subseções a seguir, ilustra as funcionalidades da plataforma web. O código da aplicação pode ser encontrado no repositório do Git no endereço: <https://github.com/code4pantanal>.

4.1. Tela Inicial

Essa imagem a seguir (Figura 5) é a tela ao qual qualquer pessoa (ator Visitante) deverá enxergar ao acessar o *Code for* Pantanal:

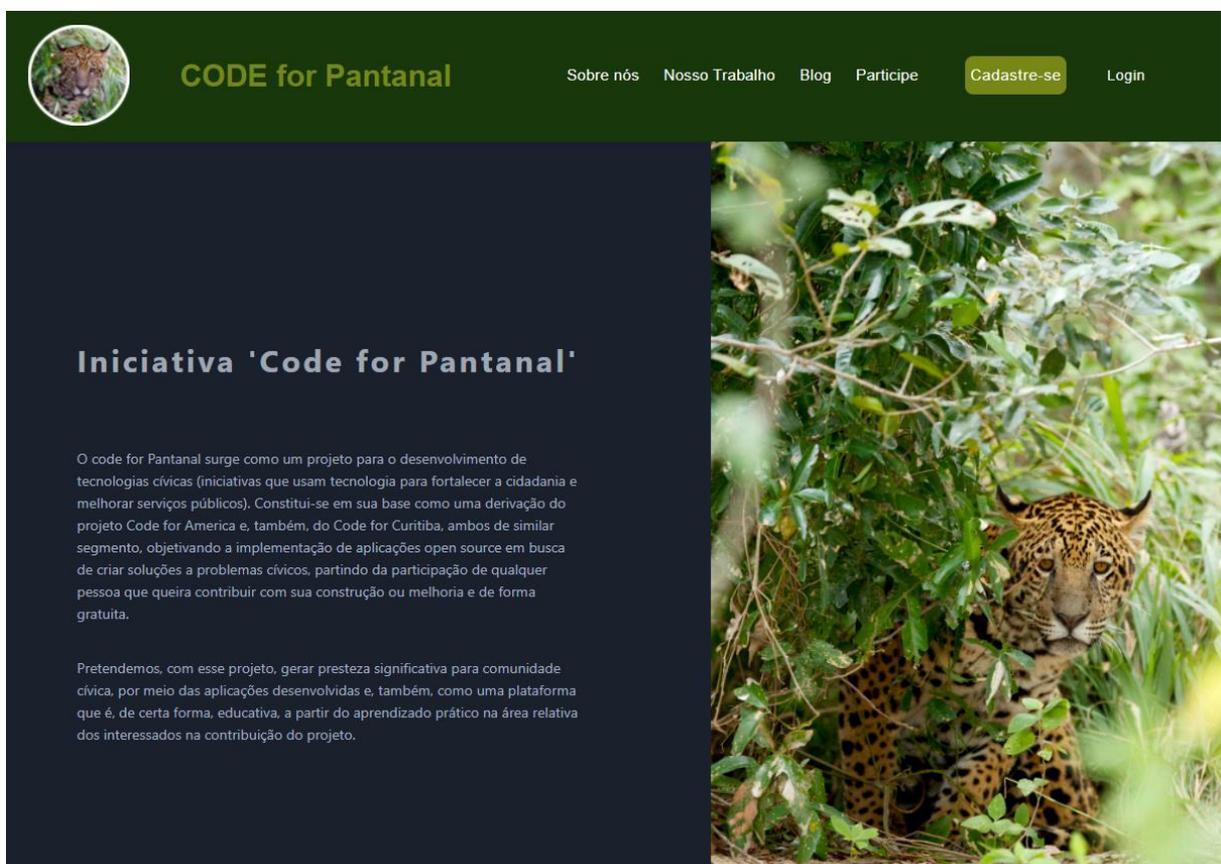


Figura 5 – Tela inicial da plataforma

4.2. Tela de Login

Caso o visitante já seja cadastrado e supondo que tenha proposto uma ideia/sugestão, ele poderá conferir os detalhes e andamento, bastando para isso que faça login, utilizando a rota apresentada na Figura 6 abaixo. Depois de autenticado, poderá ir direto para uma *Dashboard*. Da mesma forma, um colaborador poderá fazer login para verificar, na *Dashboard*, detalhes mais apurados de uma ideia, seu status em relação a ideia, alterar detalhes relativos aos projetos ao qual esteja envolvido, entre outras atividades. Também futuramente, pretende-se atrelar ao sistema a API OAuth do GitHub, para que facilite o *logon*, principalmente de desenvolvedores.

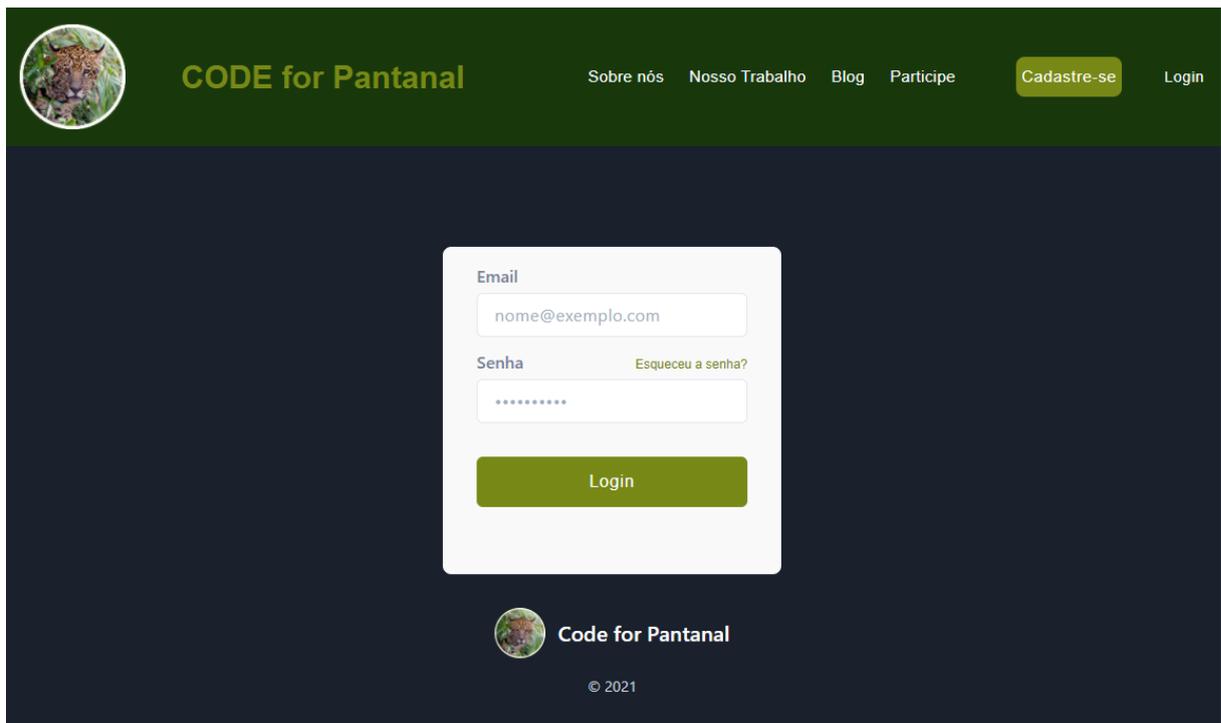


Figura 6 – Tela de Login

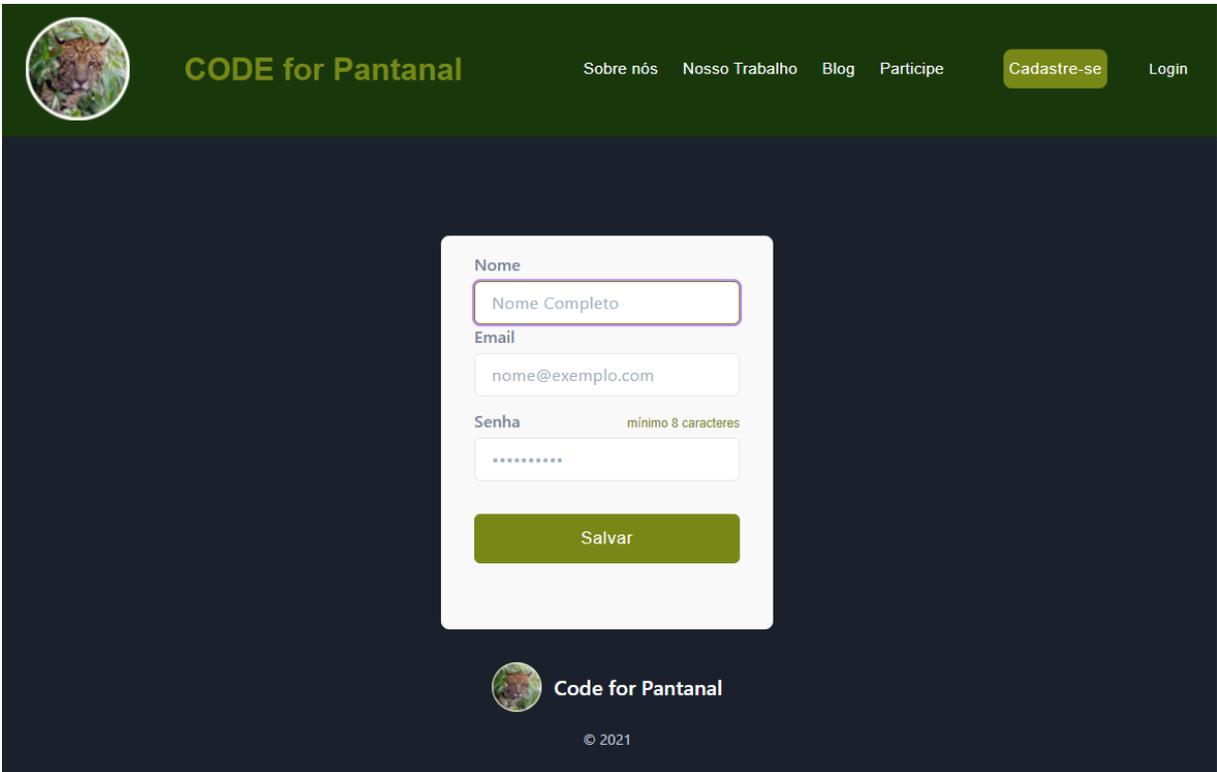
4.3. Tela de Cadastro de Pessoas

Mas caso o visitante não tenha conta na plataforma e queira propor uma ideia ou então se tornar um colaborador, poderá utilizar a rota apresentada na imagem a seguir (Figura 7) para fazer seu cadastro. Da mesma forma que dito anteriormente, iremos associar tal rota a API

OAuth do GitHub, para o caso de o visitante interessado já ter conta no GitHub. Ao finalizar o cadastro e se seus dados estiverem corretos, será encaminhado um e-mail para a caixa de entrada cadastrada contendo os dados preenchidos e a senha.

Um usuário que queira se tornar Colaborador Desenvolvedor ou Colaborador de Teste, poderá executar essa ação na *Dashboard* em campo específico. Para se tornar Colaborador, o usuário deverá:

- i. Ou acessar usando o OAuth do GitHub;
- ii. Ou informar sua conta do GitHub, para que o sistema faça a busca pelo usuário e indexe seus dados;
- iii. Ou iniciar o cadastro de uma conta nova no Git a partir de um link disponibilizado (neste caso o sistema irá redirecioná-lo para o cadastro e quando retornar irá fazer uma requisição dos dados preenchidos);
- iv. E, deverá aceitar as condições de colaborador que, de modo geral, versa sobre disponibilizar de forma livre qualquer código que tenha feito ou lógica que tenha usado, não cobrar ou reter por qualquer parte do código, não utilizar de código malicioso, respeitar as regras os códigos sociais comuns, entre outros detalhes.



A imagem mostra a interface de usuário para o cadastro no site "CODE for Pantanal". No topo, há um menu de navegação com links para "Sobre nós", "Nosso Trabalho", "Blog", "Participe", "Cadastre-se" e "Login". O formulário de cadastro centralizado contém os seguintes campos:

- Nome:** Campo de texto com o placeholder "Nome Completo".
- Email:** Campo de texto com o placeholder "nome@exemplo.com".
- Senha:** Campo de texto com caracteres ocultos por pontos e uma dica de "mínimo 8 caracteres".

Abaixo dos campos, há um botão verde "Salvar". Na base da página, há o logotipo "Code for Pantanal" e o ano "© 2021".

Figura 7 – Tela de cadastro de usuário/colaborador.

4.4. Tela de cadastro de Sugestão

A imagem a seguir (Figura 8) apresenta a tela onde o ator Usuário poderá fazer o cadastro de sua ideia. Ao clicar em salvar a ideia irá para o banco de dados com notação de sugestão, o sistema irá determinar um “Colaborador Gerente”, baseado em alguns fatores que serão determinados posteriormente. Assim que a ideia for aprovada pelo Colaborador Gerente, o sistema irá notificar todos os colaboradores na *Dashboard* e por e-mail e, também, irá indexar este usuário como “Colaborador Criador”. O Colaborador Gerente poderá preencher detalhes iniciais sobre a proposta e, à medida que a ideia for sendo implementada, este ficará responsável por alimentar detalhes mais refinados referentes ao escopo do que agora se tornou um projeto de software. Este colaborador será um dos responsáveis por finalizar o projeto relativo à proposta inicial juntamente com o “Colaborador Criador”, após a confirmação de que a aplicação resultante atende aos requisitos propostos e resolve o problema que fora descrito na ideia inicial e/ou acata aos objetivos declarados. Fica claro assim que serão sempre duas pessoas que deverão finalizar um projeto.

The image shows a web interface for submitting a suggestion. At the top, there is a dark green header with the logo of CODE for Pantanal on the left and navigation links: 'Sobre nós', 'Nosso Trabalho', 'Blog', 'Participe', 'Cadastre-se', and 'Login'. The main content area is dark blue. In the center, there is a white form with the following fields:

- Título da Sugestão**: A text input field with the placeholder 'Digite um titulo'.
- Descrição da Sugestão**: A larger text input field with the placeholder 'Digite uma descrição para sua Sugestão'.
- Objetivo**: A text input field with the placeholder 'Digite um objetivo' and a plus sign icon to its right.

At the bottom of the form is a green button labeled 'Salvar'. Below the form, there is a footer with the CODE for Pantanal logo and the text '© 2021'.

Figura 8 – Tela de cadastro de Sugestão

4.5. Tela de acompanhamento do status da Sugestão

A imagem a seguir (Figura 9) apresenta um protótipo inicial da *Dashboard*, relativa ao Usuário, após ele salvar uma sugestão (no caso o “Aplicativo para notificar ruas escuras”). Observe que há a possibilidade de outras ideias/projetos também já estarem sendo disponibilizadas para este usuário e este poderá entrar em detalhes para ver mais sobre a ideia. Enfatizamos que este é um protótipo e está em fase de implantação, já que percebemos a sua necessidade depois de iniciado este trabalho.



Figura 9 – Protótipo de *Dashboard* do usuário.

4.6. Proposta de módulo de cursos e videos-tutoriais de projetos

Ficou muito claro para nós que o processo de desenvolvimento demanda bastante conhecimento, e isso reforça a necessidade de plataformas como essa, já que é nesse tipo de situação em que um sujeito poderá aprender e/ou treinar suas habilidades em programação.

Um questionamento que surgiu no decorrer da implementação é: “Quais as

dificuldades que o programador pode ter ao contribuir em um projeto *open source*?”. Entre as respostas iniciais, verificamos a exemplo: i) falta de um escopo do projeto, ii) ausência de documentação, iii) falta de informações sobre compartilhamento de código, iv) insegurança em programação, entre outros, que podem vir a desmotivar eventuais colaboradores na participação nestes projetos, como já foi mencionado acima.

Partindo da noção dessas possíveis dificuldades, nos atentamos à necessidade de adicionar ao projeto um módulo de aprendizado (em desenvolvimento), com vislumbre da tentativa de poder fornecer usabilidade para fomentar o interesse de colaboradores ao projeto. A Figura 10 a seguir ilustra o cenário participativo ao contribuir com o *Code for Pantanal*. Portanto, em dado momento, a atenção se voltou para o aprimoramento da experiência do usuário da plataforma. Lembrando que a intenção deste trabalho também é disponibilizar um meio de que entidades, inclusive governamentais, ou qualquer pessoa interessada, possa propor uma ideia/sugestão. Sendo assim, quem estiver interessado em colaborar terá que ter todo um aparato relativo à sugestão de projeto, tais como: descrição detalhada da ideia, objetivos gerais e específicos, geração de requisitos, geração de artefatos como diagramas e protótipos, além de relatórios gerais e de teste, entre outros detalhes que venham a ser relevantes para a devida conclusão de um projeto proposto na plataforma.

A necessidade de todos esses detalhes se dá por conta de que, em geral, qualquer projeto de software deve ter um escopo bem determinado, caso contrário o desenvolvedor poderá ter tempo e trabalho desperdiçado ou ainda, pode-se nem chegar à conclusão do projeto proposto, enfatizando que a proposta é a construção de projetos para a comunidade como um todo. Também por conta disso, acreditamos que a plataforma deverá ser mais independente de qualquer um dos seus criadores (nós, que entraremos como Administradores do sistema) e/ou de seus colaboradores, sendo assim, pretendemos estabelecer uma regra para que colaboradores possam também ser administradores da ideia proposta sem que precise de uma interferência hierárquica superior na plataforma. Surgiu então a necessidade de um *Stage* de aprovação de sugestões, ao qual não é escopo deste TCC, mas que irá compor a plataforma futuramente. Neste *Stage*, que será implementada como uma seção da *Dashboard* do colaborador que o sistema irá determinar como Colaborador Gerente, haverá a possibilidade de ser verificada a ideia, ver dados do proponente, solicitar mais detalhes relativos a sugestão, dar o aceite e liberar para a criação do repositório no Git (onde será feita as adições de códigos), mas também haverá a possibilidade desse colaborador de não aceitar ser o Colaborador Gerente ou de recusar totalmente a ideia (agindo como moderador). A *Dashboard* mencionada, será uma área disponível na plataforma em que todos que tem conta terão acesso. O Usuário utilizará para ver

o andamento de sua proposta, ou de outras, já os colaboradores poderão acessar detalhes sobre sua conta e colaboração, bem como sua graduação no sistema.

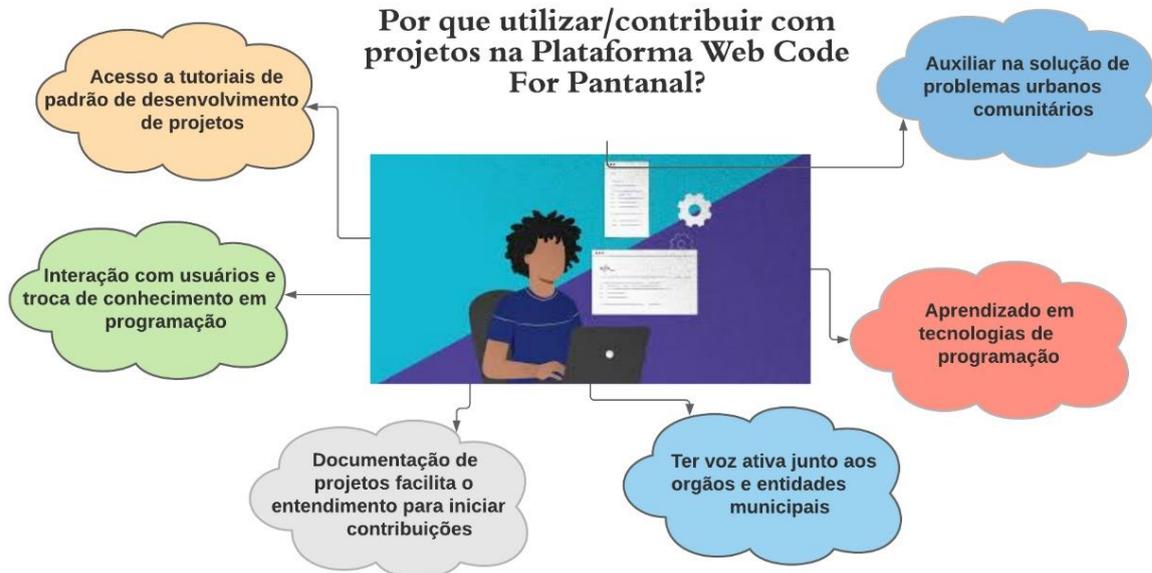


Figura 10 – Motivos para utilizar o *Code for Pantanal*

Com o projeto em evolução, nossas perspectivas para essa plataforma é que de fato seja benéfica tanto para o aprendizado dos colaboradores, permitindo que estes testem suas habilidades na prática, use sua tecnologia preferida, se engaje na comunidade como um todo, mas principalmente que seus conhecimentos sejam úteis à essa comunidade, quanto também para que cidadãos comuns façam reivindicações para problemas sociais que possam ser resolvidos através da tecnologia e que entidades governamentais também estejam engajadas no processo.

5. CONSIDERAÇÕES FINAIS

Em relação a plataforma *Code for Pantanal*, podemos afirmar que surge em um momento de grande necessidade de tecnologias que atendam a comunidade local de Dourados/MS principalmente, mas, como carrega o nome de uma região toda: o Pantanal, poderá ser útil para os problemas que são comuns nas cidades dessa região, ou até para o Centro-Oeste como um todo. Esta plataforma, assim, poderá contribuir na busca de melhorias de

problemas cívicos, ao mesmo tempo que contribui com a participação da comunidade local e, como boa consequência disso, inserirá um novo formato de ensino de programação, já que por Dourados se tratar de uma cidade universitária que contém um grande centro universitário com duas universidades públicas (UFGD e UEMS), um instituto federal de ensino (IFMS), além de universidades privadas, com cursos de tecnologia, além de outras instituições da cidade ou do estado que também contam com curso de tecnologia, poderá utilizar a plataforma como um meio de ensino prático.

Poderia ser questionado a real necessidade de uma plataforma deste tipo, que indexa o projeto em si, já que existe o próprio GitHub que já faz algo similar, contudo é importante salientar que quanto mais próximo da sociedade ao qual está servindo uma tecnologia o for, mas ela será significativa. Sem o direcionamento correto, o GitHub, ou qualquer outra plataforma do mesmo sentido, será só um repositório de códigos, sem a devida destinação. O *Code for Pantanal*, que está sendo proposto acima pretende atender a este anseio.

Assim, podemos concluir que além de contribuir com a busca de soluções para problemas urbanos, pode ser utilizado como uma ferramenta de apoio à didática em ambiente de programação, ao mesmo tempo que poderá fornecer “voz” a comunidade e as entidades governamentais ou que tenha interesse em projetos sociais. E, em relação ao aproveitamento acadêmico dos autores deste trabalho, a construção dessa plataforma abrange várias influências positivas, já que saem agora com mais uma experiência de construção de solução tecnológica, pensada desde o início e levando em conta os diversos percalços pelo caminho. Foram postos em práticas na construção dessa plataforma alguns conhecimentos adquiridos durante o curso: desde lógica de programação, banco de dados, gerência de projetos, até IHC, tudo compondo a engenharia de software da plataforma, conforme foi ensinado durante o curso, que é o que se espera ao concluir um curso de graduação.

Como um dos princípios do projeto é ser *open source* e aberto a atualizações, aqui registramos possíveis melhorias que podem ser abordadas junto à comunidade, considerando ao estágio atual da versão desenvolvida e descrita neste trabalho, dentre as sugestões, citamos:

- Adicionar tratamento de erros, de modo a interpretar mensagens recebidas do servidor, convertendo as mensagens de texto, exibidas para o usuário de forma que ele tenha total condição de identificar o problema ocorrido;
- Concluir a *Dashboard* mencionada acima;
- Implementar um sistema de *To Do* para o projeto, dentro da *Dashboard* e atrelá-lo ao progresso da ideia;

- Adicionar tela de relatórios e quantificação do número de acesso dos projetos existentes.
- Verificar propostas de aprovação de sugestões por meio de votação ou critério comunitário, a fim de que o sistema se torne cada vez mais independente do administrador.
- Implementar uma rota dentro em que o visitante possa verificar estatísticas de problemas locais, conforme as ideias que forem cadastradas. Sendo essa rota composta por um sistema de localização em mapas, identificando locais pontuais de problemas urbano, facilitando inclusive a identificação pelas entidades municipais (este poderá ser uma proposta de sistema dentro da própria plataforma);
- Desenvolver tela administrativa para utilização dos órgãos públicos municipais, de modo que consigam acompanhar e gerenciar projetos e utilizar as tecnologias cívicas desenvolvidas em prol da população. Esta página (*front-end*) seria capaz de conectar com a API (*back-end*) desenvolvida neste trabalho;
- Adicionar opção para sugestões de melhorias na plataforma, de modo similar ao módulo que propõe ideias de projetos.
- Além destas citadas, quaisquer funcionalidades que retornem em melhorias e reduzam a independência dos administradores, devem ser analisadas para implementação da equipe de colaboradores do projeto.

Enfim, espera-se que a partir deste movimento, e com a plataforma de divulgação de projetos, novas sugestões sejam incluídas a serviço da comunidade e que a plataforma de fato sirva ao seu propósito.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDULRAHMAN, I. (2020). SimCOVID: An open-source simulation program for the COVID-19 outbreak. **MEDRXIV**.
- ARAÚJO, L. M. et al., (2018). **POSTe-IP: Aplicativo web para otimização do sistema e das tecnologias de iluminação pública**. In: Encontro Nacional de computação dos institutos federais (ENCOMPIF) – 2018.
- ASAD, M.; DANTEC, C. A. (2017) Tap the "Make This Public" Button: A Design-Based Inquiry into Issue Advocacy and Digital Civics Share on. **Proceedings CHI 17**. P. 6304-6316.
- BARACHO, R. M. A. (2020) Representação e gestão do conhecimento: Aplicações em cidades inteligentes - smart cities. **Perspectivas em Ciência da informação**, v. 25, p. 252-279.
- BOELLA, G. et al. (2019) FirstLife: Combining Social Networking and VGI to Create an Urban Coordination and Collaboration Platform. **IEEE**, v. 7.
- CALANDRE, A. G. et al. (2018). **Aplicativo de mobilidade urbana focado na cidade de Curitiba**. Disponível em: <<http://www.etic.ifc-camboriu.edu.br/2018/pdf/3.pdf>>. Acesso em: 28 nov. 2020.
- CANTELON et al., (2017) **NodeJS in Action**. Manning Publications. 2 ed. 392p.
- CHO, A. (2020) **Digital Civic engagement by young people**. Disponível em: <<https://www.unicef.org/globalinsight/media/706/file/UNICEF-Global-Insight-digital-civic-engagement-2020.pdf>>. Acesso em: 14 nov. 2020.
- CIO. (2019) **Estudo revela importância do Open Source na transformação digital**. Disponível em: <<http://www.cio.pt/2019/09/17/estudo-revela-importancia-do-open-source-na-transformacao-digital/>>. Acesso em: 22 jun. 2020.
- CLEMENTE, A. C. et al. (2018) Gerenciamento de iluminação pública. **Revista Científica Multidisciplinar núcleo do conhecimento: São Paulo**, v. 05, n. 11, p. 107-147. Disponível em: <<https://www.nucleodoconhecimento.com.br/engenharia-eletrica/iluminacao-publica>>. Acesso em: 13 dez. 2020.
- CODE FOR AMERICA. (2020) **How we do It**. Disponível em: <<https://www.codeforamerica.org/how>>. Acesso em: 24 Jun, 2020.
- CODE FOR CURITIBA. (2020). **Onibus-io**. Disponível em: <<https://github.com/CodeForCuritiba/onibus-io>>. Acesso em: 18 mai. 2020.
- CORBETT, E. (2018) Trust and community engagement in Digital Civics: Exploring opportunities for design. **Proceedings DIS 18**. p. 367-370.
- CUNHA, F. et al. (2017) **Sistemas de transporte inteligentes: Conceitos, aplicações desafios**, 76p.

CUNHA, M. A. et al. (2016) **Smart Cities: Transformação Digital de Cidades**. 1. ed. São Paulo: Programa Gestão Pública e Cidadania, 161 p.

DEMIR, U.; SHARMA, G. (2020) SigPrep: open source Web-based prework for signals and systems [SP Education]. **IEEE**. v. 37, n. 6, p. 184-191.

DUAN, C.; LEE, K. G. (2020) The adoption of free and open source software in teaching and learning: An empirical study of general education in Chu Hai college of higher education. **Learning Environment and Design**, p. 127-140.

FELLER, J. (2020) **Understanding open source software development**. Disponível em: <<http://informationr.net/ir/reviews/revs046.html>>. Acesso em: 12 dez. 2020.

GASTIL, J.; MEINRATH, S. D. (2018) Bringing citizens and policymakers together online: Imagining the possibilities and taking stock of privacy and transparency Hazards. **IEEE**, v. 51, n. 6, p. 30-40.

JAMES, D. (2009). **Crafting digital media**. p.91-122. Disponível em: <<https://link.springer.com/book/10.1007%2F978-1-4302-1888-3>>. Acesso em: 19 nov. 2020.

JAVAHERI, T. et al. (2020) CovidCTNet: An open-source deep learning approach to Identify Covid-19 using CT image. **Electrical Engineering and Systems Science**, v. 3.

JUNIOR, C. P.; SPITZ, R. (2016) **Plataformas digitais para participação cívica: Inclusão digital e inovação social digital**. In: Congresso Brasileiro de Pesquisa e desenvolvimento em design. Disponível em <<http://pdf.blucher.com.br.s3-sa-east-1.amazonaws.com/designproceedings/ped2016/0268.pdf>>. Acesso em: Dez. 2020.

KHOMH, F. (2012) **Do faster releases improve software quality? An empirical case study of Mozilla Firefox**. In: 9th IEEE Working Conference on Mining Software Repositories (MSR)– 2012, p. 179-188.

KNUTAS et al. (2019) **Software engineering in civic tech a case study about Code for Ireland**. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). Disponível em: <<https://ieeexplore.ieee.org/document/8797637>>. Acesso em: 13 set. 2020.

KUMAR, A.; SINGH, R. K. Comparative analysis of AngularJs and ReactJS. *International journal of latest trends in engineering and technology*. v. 7, n. 4, p. 225-227.

MACHADO, et al. (2017) **SEMDENGUE aplicativo colaborativo para identificar foco de dengue**. Disponível em: <https://play.google.com/store/apps/details?hl=pt_BR&id=colab.semdengue>. Acesso em: 20 nov. 2020.

MACIEL, D. L.; PIAIA, T. C. (2018) A inclusão de pessoas com deficiência em cidades inteligentes. **Revista Direitos Humanos & Sociedade**, v. 1. n. 2. Disponível em: <<http://periodicos.unesc.net/dirhumanos/article/view/5025/4543>>. Acesso em: 27 nov. 2020.

MASSONI, L. F. H. et al. (2017) As narrativas da cidade no aplicativo Porto Alegre Guide. **Informação e Sociedade**: Joao Pessoa, v. 27, n. 1, p; 147-160. Disponível em: <<https://periodicos.ufpb.br/index.php/ies/article/download/31229/17419/>>. Acesso em: 17 nov. 2020.

MEDLOCK, S. et al. (2020). Implementation of an Open-Source Electronic Health Record for Decision-Support Education in Medical Informatics. **Studies in Health Technology and Informatics**, v. 270, p. 981-985.

MOHAMADOU, Y. et al. (2020) A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. **Applied intelligence**. v. 50, p. 3913-3925.

NATAHOUSE. (2020) **Open source: o que é, qual a importância e o atual cenário**. Disponível em: <<https://natahouse.com/pt/open-source-o-que-e-qual-a-importancia-e-o-atual-cenario/>>. Acesso em: 12 dez. 2020.

NETO, J. A.A.; AMORIM, N. G. (2019). **Desenvolvimento de um modelo de aplicativo para celular do transporte urbano coletivo da cidade de São José dos Campos – São Paulo**. In: VI Congress of Industrial Management and Aeronautical Technology. Disponível em: <<https://publi.cacao.cimatech.com.br/index.php/cimatech/article/view/262/67>>. Acesso em: 09 Dez. 2020.

OLIVIER, P.; WRIGHT, P. (2018) **Digital civics: takin a local turn**. **Interactions**, v. 22, n. 4.

ONU. Organizações das Nações Unidas. (2019) **Revision of World Population Prospects**. Disponível em: <<https://population.un.org/wpp/>>. Acesso em: 14 Jun. 2020.

OPUS. Opus Software. (2021). Disponível em: <<https://www.opus-software.com.br/node-js/>>. Acesso em: 03 fev. 2021.

PACHECO, E. et al. (2019) **Saúde pública e educação ambiental: os casos dos agentes de endemias no combate à dengue em Campo Grande - MS no ano de 2016**. Educação Ambiental em ação: Pelotas, v. 2, n. 68, p. 29-37. Disponível em <<http://www.revistaea.org/artigo.php?idartigo=3729>>. Acesso em 11 Dez. 2020.

PEQUENO, R.; ELIAS, D. (2020) Estruturação urbana e questão de moradia nas cidades de agronegócio. **GeoTextos**: Bahia, v. 16, n. 1, p. 109-136. Disponível em: <<https://www.rbspa.ufba.br/index.php/geotextos/article/view/37591/21432>>. Acesso em: 05 dez. 2020.

PERENS, B. (1999) **The open source definition**. Disponível em: <<https://www.oreilly.com/openbook/opensources/book/perens.html>>. Acesso em: 03 dez. 2020.

PUBLIC LAB. (2020) **Projects**. Disponível em: <<https://publiclab.org/wiki/projects>>. Acesso em: 14 Jul. 2020.

RAELOVICH, S. A. et al. (2020) Some didactic opportunities of application of mobile Technologies for improvement in the education process. **Journal of critical Reviews**. v. 7, n. 11.

REDHAT. (2017) **O que é o open source?**. Disponível em <<https://www.redhat.com/pt-br/topics/open-source/what-is-open-source>>. Acesso em: 22 de jun. 2020.

ROCHA et a. (2019) Índice de Popularidade das Linguagens de Programação e Frameworks Front-end e Back-end nas Fábricas de Software da Região de Belo Horizonte. v. 1, n. 1. Disponível em: <<http://revista.fumec.br/index.php/computacaoesociedade/article/view/7308>>. Acesso em: 27 nov. 2020.

ROSA, R. M. (2014) Tecnologias cívicas y participation ciudadana. **Revista de estudios de juventude**: Madrid, v. 105, n. 1, p. 53-70. Disponível em: <<https://e-archivo.uc3m.es/handle/10016/25852>>. Acesso em: 02 dez. 2020.

SATTAROV, O. et al. (2020) Recommending cryptocurrency trading points with deep reinforcement learning approach. *Application Science*. v. 10, n. 4.