



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

GeoBlocos: uma aplicação descentralizada de incentivo a informações geográficas voluntárias

Erik Bernardo Brito

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador
Prof. Dr. Jorge Henrique Cabral Fernandes

Brasília
2025



GeoBlocos: uma aplicação descentralizada de incentivo a informações geográficas voluntárias

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof. Dr. Jorge Henrique Cabral Fernandes (Orientador)
CIC/UnB

Prof. Dr. Edison Ishikawa	Prof. Dr. Jonathan Rosa Moreira
CIC/UnB	CIC/UnB

Prof. Dr. Jorge Henrique Cabral Fernandes
Coordenador do Curso de Computação — Licenciatura

Brasília, 11 de fevereiro de 2025

Dedicatória

Dedico este trabalho a todos que construíram as bases de conhecimento que fundamentaram seu desenvolvimento.

Agradecimentos

Agradeço aos meus irmãos André, Gabriel e Henrique, pela companhia e incentivo em todos os momentos da vida e aos meus pais pela educação e os ensinamentos que me fizeram ser quem sou hoje.

À minha namorada Paula por todo o apoio e companheirismo ao longo do período de desenvolvimento deste trabalho e em todos os outros momentos da vida.

A todos os meus amigos pelas conversas sobre este trabalho e que trouxeram grandes esclarecimentos em seu andamento.

Ao meu orientador, Jorge Henrique Cabral Fernandes, pela orientação, presteza e paciência que tornaram a realização deste trabalho possível. Também a todos os professores do CIC pelos conhecimentos transmitidos ao longo desses anos.

Por fim, agradeço a todos que trabalharam e trabalham em prol de uma educação superior pública e de qualidade, o que possibilitou o meu acesso e o de muitos outros à UnB.

Resumo

O objetivo deste trabalho é explorar o potencial do modelo de aplicações descentralizadas baseadas em *blockchain* para desenvolver uma aplicação que atue no incentivo a contribuições de Informações Geográficas Voluntárias (IGV) no Sistema de Informação Geográfica (SIG) aberto OpenStreetMap, recompensando os usuários por meio da geração de tokens não-fungíveis, os *NFTs*, para contribuições a projetos de mapeamento criados no mesmo sistema. O *software* desenvolvido consiste em uma extensão ao navegador de internet Google Chrome que está integrada à base de dados aberta do OpenStreetMap e utiliza a rede *blockchain* Ethereum para persistência dos seus dados utilizando *smart contracts*. A disponibilidade de dados geográficos é de grande importância para compreensão do espaço geográfico como mediador das relações humanas, com essas informações sendo utilizadas desde o desenvolvimento de políticas públicas à aplicação na Educação promovendo a integração entre a escola e a comunidade em que está inserida. A metodologia utilizada foi o ciclo de desenvolvimento de *software* com a validação sendo feita por testes que compõem o ciclo de desenvolvimento aliada a pesquisa bibliográfica para comparação dos fundamentos teóricos aos resultados obtidos. Para trabalhos futuros, propõem-se a identificação e desenvolvimento de novas funcionalidades pertinentes ao *software*, bem como a possível aplicação em ambiente escolar da aplicação como ferramenta de ensino e avaliação da sua adequação a este fim.

Palavras-chave: informação geográfica, IGV, blockchain, aplicações descentralizadas

Abstract

The objective of this work is to explore the potential of the decentralized applications model based on blockchain to develop an application that promotes contributions of Volunteered Geographic Information (VGI) in the open Geographic Information System, OpenStreetMap, rewarding users through the generation of non-fungible tokens (NFTs) for contributions to mapping projects created within the same system. The developed software consists of a browser extension for Google Chrome that is integrated with the OpenStreetMap open database and uses the Ethereum blockchain network for data persistence through smart contracts. The availability of geographic data is of great importance for understanding geographic space as a mediator of human relationships, with this information being used in everything from the development of public policies to its application in education, promoting integration between the school and the community in which it is located. The methodology used was the software development cycle, with validation carried out through tests that are part of the development cycle, combined with a bibliographical review to compare the theoretical foundations with the results obtained. For future work, it is proposed to identify and develop new relevant functionalities for the software, as well as the possible application of the tool in a school environment as a teaching tool and evaluation of its adequacy for this purpose.

Keywords: geographic information, VGI, blockchain, decentralized apps

Sumário

1	Introdução	1
1.1	Compreensão do espaço geográfico	1
1.2	Informações Geográficas Voluntárias	2
1.3	Aplicações em Blockchain	2
1.4	Pergunta inicial de pesquisa	3
1.5	Justificativa de pesquisa	3
1.6	Objetivo geral	4
1.7	Objetivos específicos	4
2	Metodologia	5
2.1	Pesquisa bibliográfica	5
2.2	Ciclo de desenvolvimento	5
3	Referencial teórico	7
3.1	Sistemas de Informação Geográfica	7
3.2	Informações Geográficas Voluntárias	8
3.3	Docência escolar e território	11
3.4	Blockchain	13
3.4.1	Modelos de consenso	16
3.4.2	Smart contracts	18
3.5	Aplicações descentralizadas	19
4	Tecnologias utilizadas	21
4.1	OpenStreetMap	21
4.2	Extensões do browser Google Chrome	25
4.3	Rede Ethereum	25
4.3.1	Solidity	26
4.4	MetaMask	26
4.5	InterPlanetary File System	27
4.6	Tokens Não-Fungíveis	27

5	Desenvolvimento	28
5.1	Descrição da aplicação	28
5.2	Requisitos da aplicação	28
5.3	Casos de uso	29
5.4	Desenho arquitetural da aplicação	30
5.5	Detalhamento dos componentes	33
5.5.1	Extensão de navegador	33
5.5.2	Persistência de dados	35
5.6	Arquitetura do código	37
5.6.1	Smart contracts	39
5.6.2	Utilização da rede Ethereum	41
5.6.3	Armazenamento de arquivos na rede IPFS	43
5.6.4	Acesso à base de dados do OpenStreetMap	46
5.7	Integração e testes	47
5.7.1	Ferramentas de teste	47
5.7.2	Testes de validação	48
5.7.3	Testes de verificação	49
6	GeoBlocos	50
6.1	Apresentação visual do protótipo	50
6.1.1	Tela Meus Mapeamentos	50
6.1.2	Tela Meus NFTs	52
6.1.3	Tela Projetos Abertos	53
6.1.4	Tela Meus Projetos	54
6.1.5	Tela Criar Projeto	55
6.2	Instalação da aplicação	56
6.3	Guia de uso	60
7	Discussões	65
7.1	Avaliação dos resultados	65
7.2	Trabalhos futuros	66
	Referências	68

Lista de Figuras

3.1	Funcionamento do encadeamento de blocos (Adaptado).	15
3.2	Funcionamento genérico de uma rede <i>blockchain</i> (Adaptado).	16
3.3	Fluxo de funcionamento de um <i>smart contract</i> (Adaptado).	18
3.4	Comparativo entre aplicações centralizadas e descentralizadas (Adaptado).	20
4.1	Tela do OpenStreetMap com o mapa do Distrito Federal e Entorno.	22
4.2	Etiquetas do OpenStreetMap associadas à UnB.	23
4.3	Tela de edição do OpenStreetMap.	24
5.1	Casos de uso da aplicação GeoBlocos.	30
5.2	Arquitetura da aplicação sob a Visão Conceitual.	31
5.3	Arquitetura da aplicação sob a Visão de Módulos.	32
5.4	Organização do padrão de arquitetura MVC (Adaptado).	33
5.5	Arquitetura de componentes da extensão de navegador.	34
5.6	Modelagem dos <i>smart contracts</i> da aplicação.	36
5.7	Estrutura de diretórios da aplicação.	38
5.8	<i>Smart contract</i> de gerenciamento de NFTs.	40
5.9	<i>Smart contract</i> do banco de dados de projetos de mapeamento.	41
5.10	Integração da aplicação com a rede Ethereum.	43
5.11	Armazenamento de arquivos na rede IPFS.	44
5.12	Armazenamento dos metadados dos NFTs.	45
5.13	Integração da aplicação com as APIs do OpenStreetMap.	47
5.14	Ambiente de desenvolvimento Remix Online IDE.	48
6.1	Tela de mapeamentos criados pelo usuário.	51
6.2	Pop-up de solicitação de confirmação de criação de NFT.	52
6.3	Tela de NFTs criados à partir dos mapeamentos do usuário.	53
6.4	Tela de projetos de mapeamento abertos na aplicação.	54
6.5	Tela de consulta dos projetos criados pelo usuário.	55
6.6	Tela de criação de projetos de mapeamento.	56

6.7	1º passo de instalação.	57
6.8	2º passo de instalação.	57
6.9	3º passo de instalação.	58
6.10	4º passo de instalação.	59
6.11	5º passo de instalação.	59
6.12	Instalação do MetaMask.	60
6.13	Tela de autenticação do OpenStreetMap.	61
6.14	Acesso à aplicação GeoBlocos.	61
6.15	Navegação entre as telas do sistema.	62
6.16	Criação de um novo NFT.	63
6.17	Criação de novo projeto.	64
6.18	Preenchimento das informações de um novo projeto.	64

Lista de Abreviaturas e Siglas

API Application Programming Interface.

CAPES Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

CID Content Identifier.

CSS Cascading Style Sheets.

DHT Distributed Hash Tables.

DOM Document Object Model.

DPoS Delegated Proof of Stake.

ERC Ethereum Request for Comments.

ETH Ether.

EVM Ethereum Virtual Machine.

GPS Global Positioning System.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

IDE Integrated Development Environment.

IGV Informações Geográficas Voluntárias.

IoT Internet of Things.

IPFS InterPlanetary File System.

JSON JavaScript Object Notation.

LPoS Leased Proof of Stake.

MVC Model-View-Controller.

NFT Non-Fungible Tokens.

PoA Proof of Activity.

PoB Proof of Burn.

PoET Proof of Elapsed Time.

PoS Proof of Stake.

PoW Proof of Work.

SIG Sistemas de Informação Geográfica.

TCU Tribunal de Contas da União.

TEIA Territórios Escolares Inteligentes e Abertos.

UN-GGIM United Nations Committee of Experts on Global Geospatial Information Management.

UnB Universidade de Brasília.

Capítulo 1

Introdução

1.1 Compreensão do espaço geográfico

A compreensão do espaço geográfico é um fator de extrema importância no desenvolvimento da história e desempenho das atividades de sociedades humanas. Sintetizando diversas abordagens sobre o entendimento do que são territórios, Milton Santos destaca como uma noção ampla que “o território compõe de forma indissociável a reprodução dos grupos sociais, no sentido de que as relações sociais são espacial ou geograficamente mediadas” [1]. Considerando essa relação observada, o autor destaca ainda que tal noção é passível de “ser estendida a qualquer tipo de sociedade, em qualquer momento histórico, e podendo igualmente ser confundida com a noção de espaço geográfico”. Entende-se, portanto, que o território é definido por referência às relações sociais, sendo estas também relações de poder, em determinado espaço geográfico.

Os impactos das relações entre pessoas e os territórios que elas ocupam também tem ganhado atenção para o desempenho da Educação. Em vistas de promover o ensino das pessoas em uma sociedade voltado ao desenvolvimento da sua cidadania plena e de forma democrática, adequada ao nosso tempo mergulhado em tecnologia, surgiram nas últimas décadas, iniciativas [2] que integram a educação curricular tradicional aos conhecimentos que dialogam e compõem o território de vivência dos estudantes. Esta integração permite o desenvolvimento de capacidade de ação e autonomia no espaço e na sociedade em que estão inseridos. Assim cabe aos profissionais da Educação adotarem esse olhar para fora das salas de aula e para a vivência dos seus alunos para buscar elementos das regiões que frequentam a fim de que sejam incorporados nos ensino oferecido, promovendo a troca de conhecimentos de maneira bidirecional.

1.2 Informações Geográficas Voluntárias

Os avanços tecnológicos no desenvolvimento da computação e comunicação digital através da Internet, possibilitando o acesso massivo da população a grandes volumes de informação de forma praticamente instantânea, potencializou o desenvolvimento de diversas bases de dados fundamentadas na difusão do conhecimento de forma colaborativa. Juntando-se a tal contexto o aprimoramento dos Sistemas de Informação Geográfica (SIG) [3], o acesso difundido a imagens de satélite do globo terrestre e a incorporação, em produtos dos mais diversos, do Global Positioning System (GPS), surgiram, nos anos 2000, Sistemas de Informação Geográfica de acesso gratuito, editáveis e com base de dados colaborativos, no modelo de aplicações adotado em larga escala junto à Web 2.0 em que os usuários são a principal fonte de conteúdo.

Plataformas como a Wikimapia [4], Wikiloc [5] e o OpenStreetMap [6], em grande parte o sistema que este trabalho pretende explorar, se tornaram SIGs robustos em suas bases de dados especialmente junto a construção colaborativa dessas bases alimentadas por Informações Geográficas Voluntárias (IGV) [7].

1.3 Aplicações em Blockchain

A adoção massiva de dispositivos digitais pela sociedade trouxe, além dos benefícios advindos do seu uso, desafios acerca do tráfego intenso e uso de dados. Aplicações modernas apresentam um demanda enorme de estruturas de transmissão, processamento e armazenamento de dados, fato que acaba sendo um impeditivo ao desenvolvimento de aplicações robustas que não possuam os recursos materiais, humanos e financeiros com disponibilidade e qualidade necessárias. Nesse cenário surge, no fim da primeira década do século XXI [8], uma nova tecnologia computacional possibilitando a construção de aplicações em uma estrutura física compartilhada a todos os interessados nessa aplicação, a *blockchain*.

Inicialmente, a *blockchain* foi utilizada na implementação em um sistema digital de transações financeiras que propunha um modelo confiável de realizar tais transações entre indivíduos de forma independente de qualquer entidade centralizadora com a rede Bitcoin. Entretanto, ao longo dos anos a percepção de confiabilidade com a crescente adoção ao Bitcoin encorajou o surgimento de novos usos baseadas na tecnologia *blockchain* [9]. Dentre as novas abordagens desenvolvidas com base nas redes descentralizadas, destaca-se a criação dos *smart contracts* [10] que possibilitaram o uso da estrutura de redes *blockchain* para o desenvolvimento de programas computacionais executados usando o poder computacional dos integrantes dessas redes.

A possibilidade do desenvolvimento de programas de computador usando a estrutura disponível pelas redes *blockchain* trouxe uma nova possibilidade de elaborar sistemas de informação com aplicações de uso geral com a organização de aplicações descentralizadas [11]. Desta forma, existe o potencial de explorar uma estrutura de alta disponibilidade e poder de processamento para construções de aplicações rompendo a barreira inicial de dispor de grandes recursos para uma execução prática. Apesar do uso do poder computacional de redes *blockchain* não ser gratuito, exigindo uma remuneração aos participantes da rede, os custos são proporcionais às necessidades de processamento e armazenamento demandados pela aplicação.

1.4 Pergunta inicial de pesquisa

Considerando o objetivo de promover uma educação que integre alunos e professores aos arredores do espaço geográfico em que está inserida a escola, olhando o território como ambiente de mediação das relações sociais para exercício da cidadania plena, este trabalho busca responder a pergunta: como a produção de dados geográficos voluntários pode ser incentivada, com a possibilidade de construção de aplicações descentralizadas, explorando a arquitetura de soluções computacionais baseadas em redes *blockchain*?

1.5 Justificativa de pesquisa

Sistemas de Informação Geográfica são de extrema importância na sociedade de vida digital contemporânea, assim como informações geográficas sempre foram na história, mas com a facilidade de acesso existe também a necessidade de informações confiáveis, atualizadas e de alta disponibilidade. SIGs baseados em bancos de dados colaborativos se mostraram como um meio de popularizar ainda mais o acesso a dados geográficos, mas dependem do engajamento dos seus usuários como mantenedores das informações. Além disso, a participação do autor em uma ação de extensão que visava instruir e promover o uso do OpenStreetMap, o projeto Territórios Escolares Inteligentes e Abertos (TEIA) [12], foi um fator motivador ao desenvolvimento deste trabalho como um forma de construir uma ferramenta a mais para incentivar esse uso.

Esta pesquisa pretende abordar o potencial trazido pelo modelos de aplicações descentralizadas como um meio de incentivar o fornecimento de Informações Geográficas Voluntárias. As novas possibilidades de construção de aplicações utilizando as estruturas da Web3 [13], dentre elas as aplicações descentralizadas baseadas em *blockchains*, permitem um novo leque de opções de construção de ferramentas no âmbito educacional, como a proposta neste trabalho, com a finalidade de promover uma educação mais integrada

entre a escola como instituição e a comunidade a que ela pertence enquanto o espaço geográfico em que a escola está inserida, sendo influenciada por ele e também exercendo influência sobre este espaço.

1.6 Objetivo geral

Construir uma aplicação de incentivo a IGV que seja independente de uma entidade ou indivíduo mantenedor quanto a disponibilidade do sistema e a persistência de dados. Além disso que tal aplicação demande pouco poder computacional do usuário, de acesso simples e com baixos custos no uso da rede *blockchain*, além de ser de uso amigável para todos os usuários envolvidos, visando um possível uso em ambiente educacional.

1.7 Objetivos específicos

Como resultado da pesquisa e do desenvolvimento de uma aplicação, pretende-se atingir os seguintes objetivos:

- Definir a arquitetura geral da aplicação, enfatizando a independência de serviços centralizados e a sua eficiência computacional.
- Construir um protótipo de aplicação que funcione de forma acessível a um grande número de pessoas, enquanto dependa de poucos recursos computacionais dos seus usuários.
- Elaborar um modelo, dentro da aplicação, de incentivo ao fornecimento de Informações Geográficas Voluntárias (IGV).
- Criar um guia de uso da aplicação de forma a acessibilizar e difundir o seu uso.
- Realizar a avaliação do protótipo da aplicação desenvolvida.

Capítulo 2

Metodologia

O desenvolvimento deste trabalho seguirá uma abordagem mista, combinando a pesquisa bibliográfica ao desenvolvimento prático da aplicação proposta, seguindo um ciclo de desenvolvimento de *software* tradicional em cascata.

2.1 Pesquisa bibliográfica

A pesquisa bibliográfica foi escolhida para este trabalho como forma de fundamentar a justificativa de existência da aplicação proposta à partir de uma necessidade social real identificada e também para embasar as escolhas das tecnologias utilizadas na construção dessa aplicação [14]. O método de pesquisa utilizado foi desenvolvido a partir da busca por materiais já elaborados e publicados, sendo principalmente artigos científicos, mas também livros e documentações diversas como especificações de tecnologias já existentes. Dentre as fontes desse material de referência cabe destacar a busca de publicações no Portal de Periódicos da CAPES e a bibliografia utilizada ao longo do curso de Licenciatura em Computação na Universidade de Brasília.

2.2 Ciclo de desenvolvimento

Devido ao uso de tecnologias ainda não totalmente consolidadas para o desenvolvimento da aplicação proposta neste trabalho, foi escolhido um método de desenvolvimento de *software* tradicional em cascata como forma de proporcionar maior robustez ao processo de desenvolvimento com um método de uso bem estabelecido na indústria de *software*. O ciclo de vida de *software* utilizado no desenvolvimento deste trabalho é composto pelas seguintes etapas [15]:

- levantamento de requisitos;

- identificação dos casos de uso;
- análise e projeto da arquitetura da aplicação;
- implementação dos componentes do sistema;
- integração dos componentes;
- criação de um protótipo;
- realização de testes; e
- avaliação dos resultados obtidos.

Serão analisados casos de integração de informações geográficas em sistemas com dados alimentados por Informações Geográficas Voluntárias (IGV), com ênfase na utilização do OpenStreetMap. A implementação prática será realizada por meio da construção de uma aplicação de criação de projetos de mapeamento voluntário, acoplada ao OpenStreetMap, para incentivar a contribuição voluntária por meio da geração de Non-Fungible Tokens (NFT), garantindo a integridade e imutabilidade das informações acerca dos projetos e contribuições utilizando a rede *blockchain* Ethereum. A escolha desta rede *blockchain* foi por conta da maior robustez das ferramentas de desenvolvimento disponíveis para ela em comparação a outras redes que também suportam a implementação de *smart contracts*.

Capítulo 3

Referencial teórico

Este capítulo tem como objetivo apresentar o referencial teórico que fundamentou os rumos do desenvolvimento seguido por este trabalho em direção à construção da aplicação proposta. Inicialmente contextualiza-se a evolução e as aplicações dos Sistemas de Informação Geográfica, explorando em seguida a sua relação com as contribuições de Informações Geográficas Voluntárias, além de discutir sobre a relação entre a prática docente e território em outra seção. Por fim, é apresentada a tecnologia *blockchain* e seus usos, especialmente sobre a construção de aplicações descentralizadas utilizando *smart contracts* em uma seção dedicada.

3.1 Sistemas de Informação Geográfica

Os Sistemas de Informação Geográfica (SIG) constituem sistemas computacionais que integram dados geográficos, possibilitando o armazenamento, análise e manipulação de dados relacionados a informações espaciais em um sistema automatizado. Sua aplicação compreende diversas áreas de conhecimento e atuações profissionais, uma vez que informações geográficas são necessárias em diversos campos de utilização pela sociedade como elaboração de políticas públicas, planejamento urbano, controle de epidemias por instituições de saúde, gestão de tráfego, otimização de atividades logísticas e gestão de recursos naturais. A vida contemporânea como um todo é notavelmente interdependente da qualidade da informação contida nos Sistemas de Informação Geográfica.

Estes sistemas apresentam numerosas estruturas de componentes, cada uma adequada à finalidade que o SIG é proposto a atender, resultando assim em características e organização de componentes bastante distintas entre si. Entretanto, pode-se observar, em uma visão generalista, os seguintes componentes como recorrentes a um Sistemas de Informação Geográfica: interface com o usuário; entrada e integração de dados; funções de processa-

mento desses dados; visualização e localização de espaços geográficos; e armazenamento e recuperação de dados [16].

O advento e evolução dos Sistemas de Informação Geográfica junto ao crescimento do poder computacional trouxe avanços significativos ao poder de análise e compreensão do espaço geográfico por parte dos agentes interessados nessas informações, tanto no seu uso por acadêmicos das áreas das Geociências quanto por suas aplicações nas demais esferas sociais, políticas e econômicas [3]. Desta forma resultando em sistemas de informação naturalmente multidisciplinares, também transcendendo seu uso técnico-científico à aplicação na vida cotidiana de pessoas comuns com a difusão e acessibilidade dos recursos digitais.

Dentre os fatores que contribuíram para o aprimoramento desses sistemas de informação cabe destacar o avanço das tecnologias de processamento de dados e da computação gráfica, a adoção por diversos dispositivos do Global Positioning System (GPS) e a obtenção de imagens por satélites iniciadas durante a década de 1970. Inicialmente, durante os anos 70, a utilização era ainda restrita a entidades governamentais até passar por sua difusão e popularização nos meios acadêmicos, industriais e comerciais nas décadas de 1980 e 1990. Durante os anos 2000, com a acessibilidade crescente da computação pessoal e da Internet que se atinge a massificação do uso dos Sistemas de Informação Geográfica [3].

3.2 Informações Geográficas Voluntárias

Com a popularização da Internet durante os anos 2000 e crescente aprimoramento dos Sistemas de Informação Geográfica, um fenômeno passou a ser observado com o uso de serviços com o Google Earth permitindo o acesso a imagens de satélite de todo o globo terrestre por cidadãos comuns. Tal fenômeno ficou conhecido como Informações Geográficas Voluntárias (IGV), onde a população em geral passou a construir bases de dados geográficos por iniciativa própria, atuando em uma área que por séculos ficou restrita a agente estatais no levantamento dessas informações [7]. Serviços como a Wikimapia [4], o OpenStreetMap [6] e o Wikiloc [5] forneceram a bases tecnológicas para seus usuários atuarem como geradores de conteúdo sobre as localidades em que vivem e conhecem.

Olhando para os avanços nos sistemas de informação que viabilizaram a popularização dos Sistemas de Informação Geográfica alimentados por Informações Geográficas Voluntárias cabe destacar os seguintes pontos [17]:

Web 2.0 a difusão de *softwares* possibilitando o envio de informações, por meio dos navegadores de Internet, por seus usuários para serem armazenadas em suas bases de dados, além de permitir o acesso de visualização e até mesmo edição por usuários diferentes dos criadores originais do conteúdo, viabilizou o modelo de Internet com

conteúdo criado pelos próprios usuários. Este modelo é facilmente observado em páginas Wiki e nas redes sociais.

Georreferenciamento adoção de padrões de referências ao posicionamento de localizações no globo terrestre. Desta forma facilitando a interoperabilidade entre sistemas diversos, sendo as medidas de latitude e longitude os mais difundidos.

Geotags um código padronizado que pode ser atribuído a conjuntos de dados de forma a referenciar uma localização geográfica com precisão à qual aquelas informações se referem. Sendo essas *geotags* utilizadas também fora de Sistemas de Informação Geográfica como em redes sociais como o X (antigamente Twitter), Instagram e Facebook, além de ser frequente em fotografias digitais permitindo a categorização das fotos pela localidade em que foram geradas, como nos aplicativos de álbuns de fotos em *smartphones*.

GPS a vasta aplicação do Global Positioning System (GPS) em produtos diversos como carros e *smartphones* junto à sua robustez, confiabilidade e agilidade em mensurar a posição na Terra.

Computação gráfica avanços da computação gráfica possibilitaram a visualização a imagens de alta resolução obtidas por satélite em dispositivos digitais dos mais diversos, até mesmo a visualização em três dimensões de alguns espaços.

Internet de banda larga informações geográficas junto a imagens sendo utilizadas nos sistemas por um volume massivo de pessoas demandam o tráfego de um grande volume de dados, ter a estrutura necessária e disponível é um grande diferencial em viabilizar o uso de qualquer aplicação baseada na *web*.

Um meio de fornecimento de Informações Geográficas Voluntárias que merece atenção são as submetidas por usuários de aplicações não focadas em informações geográficas por meio das *geotags*, os diversos blogs, Wikis e redes sociais se mostram como uma grande fonte de dados geográficos. Aplicativos e sites de redes sociais por exemplo, permitem que seus usuários realizem uma forma de *check in* nos lugares que visitam, fornecendo novas informações sobre esses lugares ou mesmo atualizando informações preexistentes. O fluxo de informações geográficas como essas geradas por usuários de redes sociais encontrou um uso comercial por parte de empresas de diversos segmentos como fontes de dados para elaboração de estratégias de negócios. Por outro lado a adoção de *geotags*, especialmente por redes sociais, levanta questionamentos acerca da garantia de privacidade dos usuários dessa aplicações, uma vez que nem sempre a escolha de compartilhar uma localização está facilmente visível durante o uso, como os metadados de localização de fotografias que por

vezes sequer é acessível por estes usuários, apesar de estarem embutidas nos arquivos de imagem que trafegam pela rede [18].

Apesar das preocupações acerca da privacidade com a monetização das informações coletadas de seus usuários pelas redes sociais, o relatório *Future Trends in geospatial information management* [19], produzido pela United Nations Committee of Experts on Global Geospatial Information Management (UN-GGIM), destaca também a importância das Informações Geográficas Voluntárias no sucesso da participação cidadã em projetos de mapeamento. O trabalho voluntário ou subsidiado por recursos oriundos de financiamento coletivo pode beneficiar em grande medida instituições governamentais responsáveis pela administração do território, especialmente no caso de países subdesenvolvidos que enfrentam escassez dos recursos necessários às atividades de mapeamento de forma oficial dessas localidades.

A incorporação do uso de dados obtidos através de IGV por entidades estatais pode ser benéfica também em situações onde a existência de informações oficiais é consolidada, mas encontra desafios na alocação dos recursos disponíveis para atualização tempestiva dessas informações para assegurar um desempenho de governança que atenda o interesse público [20]. Além da obtenção de dados por IGV, a adoção por governos de SIG baseados na Web 2.0 encontra aplicações nas áreas de informações em tempo real sobre tráfego e trânsito, a prestação de serviços públicos adequada à demografia das regiões e a participação dos cidadãos nos processos de planejamento e tomada de decisão promovendo a participação democrática dos cidadãos [21].

Entretanto, outro aspecto a ser considerado ao utilizar dados geográficos obtidos através de Informações Geográficas Voluntárias é a característica de que essas informações oriundas de contribuições voluntárias não possuem a garantia de qualidade. Esta qualidade da informação geográfica compreende as dimensões de precisão posicional, precisão de atributos, consistência lógica, integridade e linhagem. Frente aos desafios de assegurar esses atributos de qualidade dos dados obtidos por IGV, Goodchild [22] relata três abordagens para aperfeiçoar a qualidade dessas contribuições:

Abordagem crowd-sourcing busca assegurar a melhoria na qualidade dos dados geográficos coletados baseando-se nas ideias de que múltiplas pessoas atuando em um problema tendem a convergir em soluções de erros individuais e que ter essas fontes diversas e numerosas também ajudam a convergência em dados factuais e verdadeiros. Vale destacar que o termo *crowd-sourcing* nesse contexto de garantia de qualidade diz respeito a esses significados e não à ideia de financiamento coletivo com que o termo é frequentemente associado.

Abordagem social consiste em permitir um grau superior de poder de edição junto a poderes de moderação do conteúdo da plataforma aos seus contribuidores mais produtivos, considerando o fenômeno observado em aplicações de conhecimento colaborativo em que um grande volume dos dados partem da contribuição de uma pequena parcela dos usuários que teriam maior domínio sobre as informações tratadas.

Abordagem geográfica busca validar a qualidade de novas informações com base na consistência entre os novos dados com os dados previamente existentes e já consolidados sobre uma determinada região, assim incongruências muito divergentes entre os dados seriam indicativos de informações incorretas.

3.3 Docência escolar e território

No âmbito dos debates sobre os rumos a serem seguidos pela educação no Brasil, uma das bandeiras de grande destaque é a defesa da adoção da educação integral como uma das formas de promover a melhoria da qualidade da educação no país. A educação integral, no entanto, não consiste apenas no aumento da carga horária dos estudantes, mas também integral no sentido de englobar os demais aspectos do aprendizado humano além do conhecimento das disciplinas tradicionais como Língua Portuguesa, Matemática e as Ciências Naturais e Sociais. O aprendizado social pelos alunos durante os anos escolares são vistos como tão importantes quanto as disciplinas que compõem a grade curricular formal para a construção de uma educação realmente emancipadora para os indivíduos, fornecendo as ferramentas necessárias para compreensão e atuação plena da cidadania por indivíduos em uma sociedade verdadeiramente democrática [23].

O ponto de partida do desenvolvimento deste trabalho foi a participação do autor em uma ação de extensão realizada na Universidade de Brasília no âmbito do projeto Territórios Escolares Inteligentes e Abertos (TEIA) [12]. A ação realizada consistiu em ministrar oficinas de instrução e incentivo ao uso do OpenStreetMap para alunos de escolas públicas do Distrito Federal buscando avaliar se o seu uso permitiria a leitura de questões relativas aos territórios em que estão inseridos alunos, professores e membros da comunidade onde ficam as escolas. Neste contexto surgiu a ideia inicial de desenvolver uma aplicação que pudesse ser uma ferramenta adicional de incentivo ao uso do OpenStreetMap.

Uma das estratégias de promoção da cidadania pela educação é a união do currículo escolar tradicional ao conhecimento sobre a localidade e comunidade onde o estudante vive e como o saber científico ajuda a explicar esse mundo ao seu redor. O economista Ladislau Dowbor destaca a capacidade da educação nesta abordagem, inclusive a capacidade de promover o desenvolvimento local, uma vez que pessoas com a devida capacidade

de compreensão e formação podem ser atores em iniciativas capazes de transformar determinadas localidades, gerando dinâmicas construtivas [24]. Ainda sobre essa abordagem o autor destaca o seguinte:

Há uma dimensão pedagógica importante neste enfoque. Ao estudarem de forma científica e organizada a realidade que conhecem por vivência, mas de forma fragmentada, as crianças tendem a assimilar melhor os próprios conceitos científicos, pois é a realidade delas que passa a adquirir sentido. Ao estudarem, por exemplo, as dinâmicas migratórias que constituíram a própria cidade onde vivem, as crianças tendem a encontrar cada uma a sua origem, segmentos de sua identidade, e passam a ver a ciência como instrumento de compreensão da sua própria vida, da vida da sua família. A ciência passa a ser apropriada, e não mais apenas uma obrigação escolar.

Seguindo a ideia de trabalhar a educação em uma abordagem que una a educação de forma integrada à comunidade onde o estudante tem o seu espaço de vivência encontramos o modelo do Bairro-Escola [25] que visa a integração do ambiente escolar ao seu entorno por meio de parcerias com os diversos atores que integram um espaço geográfico. A ampliação do ambiente de aprendizado para além do espaço da escola possibilita aos alunos uma nova visão sobre o que é o conhecimento, criando uma troca entre o que é ensinado na escola para a realidade da comunidade próxima e também trazendo conhecimentos dessa comunidade para o ambiente escolar. O Bairro-Escola compreende um sistema que une escolas, famílias e comunidades para compor um território educativo, podendo apresentar diferentes formas de organização, mas visando “a garantia do desenvolvimento físico, intelectual, afetivo e social das pessoas” como afirma Helena Singer [26], que foi Diretora da Cidade Escola Aprendiz, organização que promove a tecnologia do Bairro-Escola em diversos estados do Brasil.

Apesar da denominação Bairro-Escola, a estratégia educacional não se limita a englobar necessariamente um bairro, mas sim abranger um espaço geográfico que seja relacionado aos objetivos pedagógicos pela instituição escolar. O desenho do território educativo busca ser definido com base na experiência dos alunos, familiares, professores e atores públicos e privados que estão envolvidos em um determinado espaço seja ele um bairro, uma comunidade, uma cidade ou qualquer arranjo espacial em que esses envolvidos estejam relacionados. Tal característica confere ao modelo do Bairro-Escola uma flexibilidade no seu desenho e adoção que não possibilita uma delimitação prévia do território referenciado, mas que encontramos princípios que podem ser adotados para essa delimitação [27]:

- priorizar o olhar para o microterritório;
- focar as condições para o desenvolvimento integral de crianças e jovens;
- tomar a escola como elemento central do território;

- considerar crianças, adolescentes e jovens que estudam e crianças, adolescentes e jovens que moram nos territórios; e
- assegurar o princípio da replicabilidade.

3.4 Blockchain

Com o objetivo de viabilizar a realização de transações financeiras entre duas partes interessadas sem a dependência de uma terceira parte considerada como confiável pelas partes anteriores, o Bitcoin é proposto em 2008 como um ativo financeiro digital e negociável [8]. O embasamento da viabilidade da proposta foi através a aplicação de uma nova tecnologia baseada em redes *peer-to-peer*, posteriormente denominada *blockchain*. Desta forma, possibilitando uma nova forma de realizar transações sem a necessidade de envolver uma terceira parte confiável, utilizando um sistema baseado em comprovação criptográfica como garantidor da segurança das transações com o ativo. A aplicação computacional proposta com a tecnologia *blockchain*, apesar de inicialmente pensada em usos financeiros, apresentou também formas de armazenamento de dados persistentes, imutáveis e descentralizados enquanto assegurava redundância através da rede *peer-to-peer*.

As qualidades relacionadas aos dados que a aplicação da *blockchain* proporciona entrou aplicações em diversos segmentos produtivos onde os sistemas utilizados dependem de alta disponibilidade de informações, bem como a sua integralidade, rastreabilidade e segurança. A evolução de redes *blockchain* além do Bitcoin, inicialmente dedicadas a aplicações financeiras, possibilitou encontrar usos posterior em setores como [9]:

- o rastreamento de cadeias de produção no setor alimentício;
- em sistemas de cibersegurança, com o fator da confiança tornando-se independente do fator humano e seus possíveis erros e brechas, a imutabilidade dos dados e a descentralização do sistema evitando a concentração em um único possível ponto de ataque;
- em sistemas de votação eletrônica;
- em serviços cartoriais de registros de propriedade;
- para serviços de saúde que podem se beneficiar das características de confiabilidade dos dados na *blockchain* junto ao uso de *smart contracts* para automação em prol da redução da burocracia dos processos administrativos;
- uma possível solução para as preocupações com privacidade da indústria de Internet of Things (IoT);

- o rastreamento e transparência nos registros envolvendo a indústria automobilística; e
- no aperfeiçoamento da eficiência de cadeias logísticas, registrando em tempo real o trânsito e mudança de posse de mercadorias diversas e assegurando a autenticidade das transações.

Essencialmente a *blockchain* funciona como um livro-razão digital replicado em todos os nós participantes, organizando as suas transações em blocos que posteriormente são ligados uns aos outros de forma encadeada, com o bloco atualmente em uso referenciando diretamente o bloco imediatamente anterior, de forma atrelada ao conteúdo contido neste último, e assim sucessivamente desde o bloco inicial da cadeia. A segurança dos registros da *blockchain* é salvaguardada pela aplicação de algoritmos de *hash* no dados contidos nas transações efetivadas, com o encadeamento entre todos os blocos servindo como a garantia que um dado previamente registrado não possa ser alterado sem que se altere todo o histórico dos registros. Outra característica que reassegura a integridade das informações é a publicidade do histórico dos registros, permitindo assim sua auditoria por qualquer ator interessado. Em um Sumário Executivo [28] sobre a tecnologia *blockchain*, o Tribunal de Contas da União (TCU) descreve a tecnologia como o trecho a seguir, ilustrando o seu funcionamento na Figura 3.1:

uma blockchain é uma estrutura de dados que armazena transações organizadas em blocos, os quais são encadeados sequencialmente, servindo como um sistema de registros distribuído. Cada bloco é dividido em duas partes: cabeçalho e dados. O cabeçalho inclui metadados como um número único que referencia o bloco, o horário de criação do bloco e um apontador para o hash do bloco anterior, além do hash próprio do bloco. Os dados geralmente incluem uma lista de transações válidas e os endereços das partes, de modo que é possível associar uma transação às partes envolvidas (origem e destino).

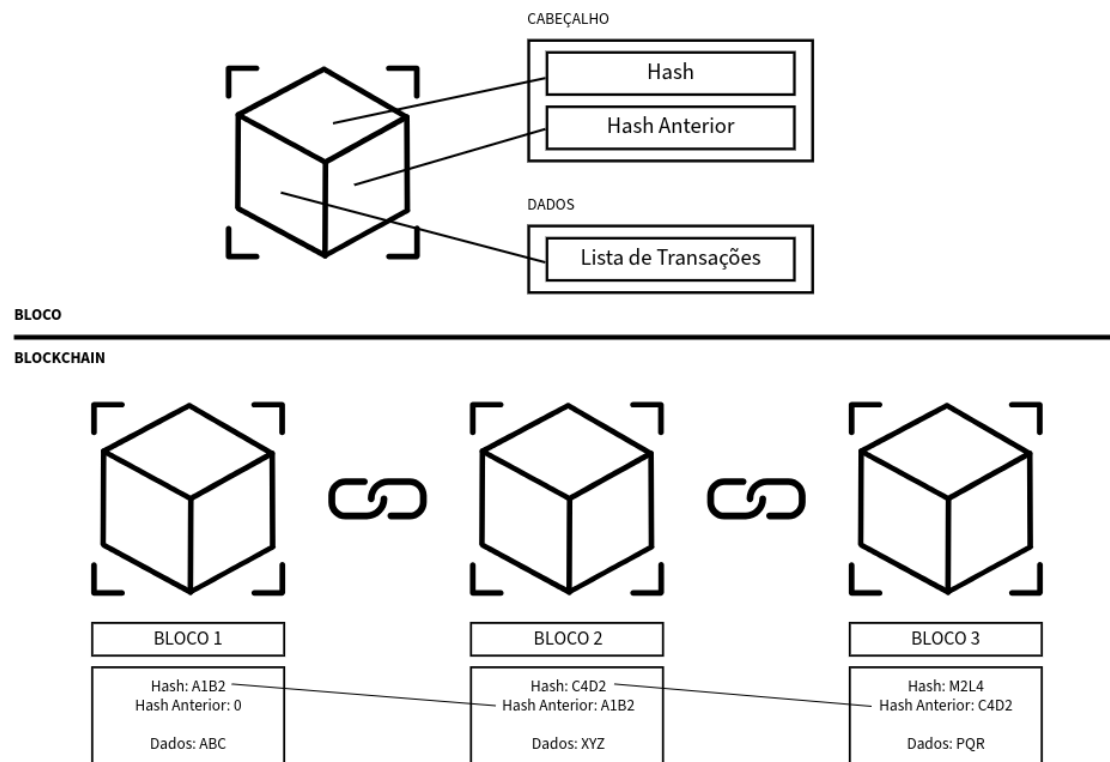


Figura 3.1: Funcionamento do encadeamento de blocos (Adaptado) (Fonte: [28]).

A caracterização da *blockchain* como um banco de dados descentralizado se deve ao armazenamento dos registros em cada um terminais conectados à sua rede *peer-to-peer* que esteja executando a aplicação, sendo chamados nós da rede [29]. A integridade da base de dados é proporcionada pela igualdade do conteúdo armazenado em cada um desses nós conectados e cada novo registro sendo integrado à cadeia somente através do consenso da maioria dos nós conectados através de mecanismos que variam a cada implementação de uma rede *blockchain*. Essa organização arquitetural da rede é um dos mecanismos que reforçam a inviolabilidade dos registros contidos na cadeia de blocos, uma vez que um atacante que deseje falsificar os dados ali contidos precisaria controlar mais da metade dos nós para obter sucesso em um ataque.

Uma síntese do funcionamento geral de uma rede *blockchain*, em etapas, é descrito no esquema representado pela Figura 3.2.

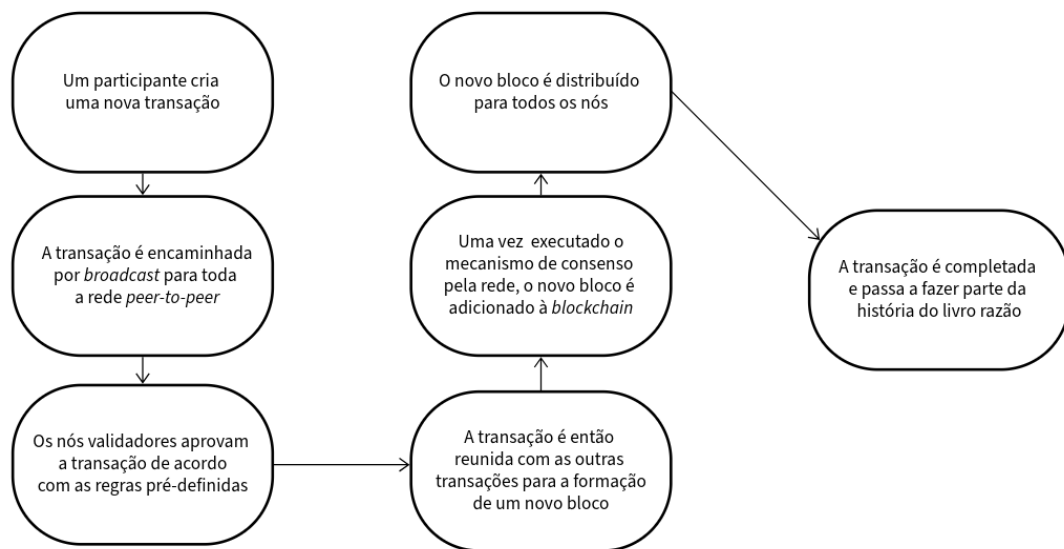


Figura 3.2: Funcionamento genérico de uma rede *blockchain* (Adaptado) (Fonte: [28]).

3.4.1 Modelos de consenso

Um fator importante na caracterização de cada implementação de uma rede *blockchain* é o seu modelo de consenso, que consiste nas regras adotadas pela rede de como novos registros de transações são validados e considerados como aceitos pela rede para integrar a base de dados na *blockchain*. Além disso, os modelos de consenso integram parte importante dos mecanismos de segurança da rede, levando em consideração formas de mitigar o sucesso de atacantes ou fraudadores. Alguns dos modelos relevantes já implementados são descritos a seguir [30]:

Proof of Work (PoW) este é o modelo inicialmente proposto e ainda utilizado pela rede Bitcoin. Uma nova solicitação de transação na rede precisa ser validada por um dos nós dessa rede que fica como o responsável por registrar na *blockchain* a nova informação. Um extenso e complexo processo de cálculos de validação do *hash* do cabeçalho do novo bloco é iniciado pelos nós candidatos a efetivamente validar a transação e adicionar o novo bloco à cadeia, sendo estes cálculos realizados pelo uso do poder computacional chamados trabalho. Dentre os nós candidatos a nó validador, é selecionado o que primeiro conseguir calcular o valor adequado ao cabeçalho do novo bloco, processo que ficou conhecido como mineração, pois o nó selecionado é recompensado por desempenhar tal tarefa.

Proof of Stake (PoS) um dos problemas apresentados na adoção do modelo Proof of Work é a sua ineficiência no consumo de energia, uma vez que vários nós estão usando um grande poder computacional para apenas um ser selecionado como validador, decorrendo assim no desperdício da energia usada pelos demais nós. Como alternativa a esse problema o Proof of Stake é proposto com o critério de seleção do validador baseado na quantidade do ativo digital em posse dos nós candidatos a validadores. Entretanto, a adoção deste modelo causa um novo problema à rede, porque concentra o poder de decisão nos participantes com mais recursos financeiros.

Leased Proof of Stake (LPoS) é uma variação do Proof of Stake proposta para mitigar a concentração do poder sobre a rede inerente ao modelo. As regras de seleção incorporam em seus algoritmos critérios para proporcionar a chance de atuar como validadores para nós em posse de poucos ativos.

Delegated Proof of Stake (DPoS) outra alternativa proposta ante os efeitos negativos do Proof of Stake. Este modelo de consenso adota a eleição, entre os nós participantes, de um pequeno grupo de nós geradores e validadores de novos blocos. Permite também a fácil remoção dos status de validador pelos demais participantes.

Forma Híbrida de PoW e PoS utilizam uma forma combinada entre Proof of Work e Proof of Stake, exigindo a prova de trabalho por cálculos computacionais, mas considerando os valores em posse do nó candidato a validador como fator de aumento de chances de sucesso.

Proof of Elapsed Time (PoET) consistem em um modelo em que para cada novo nó que ingresse na rede é atribuído um tempo de espera aleatório para que seja concedido o direito de publicar um novo bloco à cadeia armazenada na rede. A adoção desse sistema de seleção depende de execução em um ambiente seguro para evitar fraudes no tempo de espera efetivamente decorrido desde a conexão de cada nó [31].

Proof of Activity (PoA) é uma extensão do modelo Proof of Work combinada ao Proof of Stake. Apenas os cabeçalhos dos blocos são gerados por cálculos de *hash* com seu conteúdo especificando *stakeholders* aleatórios. As transações são efetivadas após o primeiro desses *stakeholders* selecionados encontrar o cabeçalho e juntar às demais informações para adicionar o novo bloco à rede. De forma que a quantidade de ativos possuídos por um nó impactando em maiores chances de aprovação de blocos [32].

Proof of Burn (PoB) o critério de seleção neste modelo é baseado na “queima” de criptomoedas da rede ao enviar parte dos ativos em posse de um nó para um endereço

onde ele fica inutilizável, mas não inexistente. Desta maneira o poder de escrita de novos blocos é condicionado à capacidade de um nó minerador de abrir mão de seus recursos em troca da geração de novos blocos.

3.4.2 Smart contracts

Expandindo as possibilidades de uso do modelo de computação descentralizada iniciado na rede Bitcoin e a implementação da *blockchain*, surgiram novas redes com a funcionalidade de executar os *smart contracts*. O seu funcionamento consiste no armazenamento de códigos em linguagens de programação que podem ser replicados e executados usando o poder computacional disponível na rede *peer-to-peer* através, e sob supervisão, de seus nós. Quando uma determinada condição previamente estabelecida no contrato é atendida, o código escrito e armazenado na rede do contrato é executado de fato [10]. O fluxo de como um *smart contract* funciona é descrito na Figura 3.3.

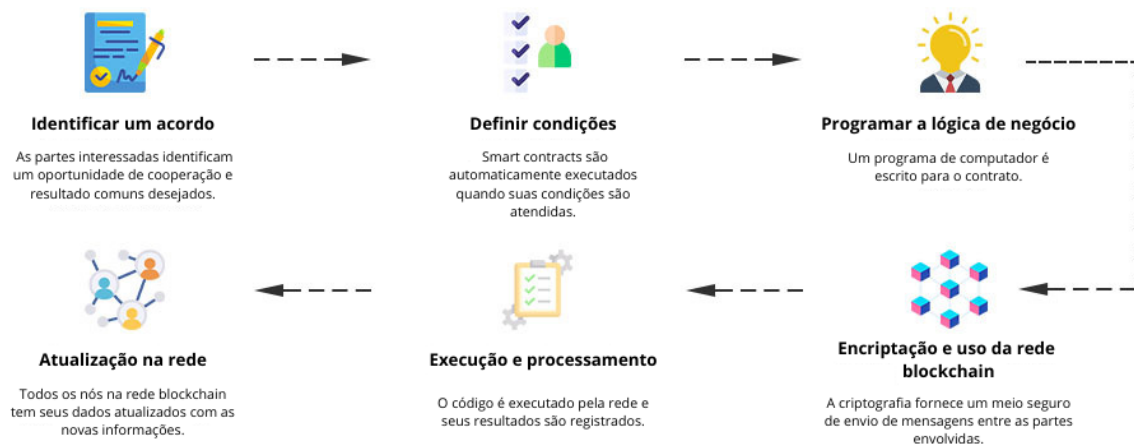


Figura 3.3: Fluxo de funcionamento de um *smart contract* (Adaptado) (Fonte: [33]).

Os *smart contracts* mantêm os princípios comuns a redes *blockchain* assegurando um meio seguro, transparente e auditável de realizar transações sem a dependência de uma entidade centralizadora, porém com a flexibilidade de criação de inúmeras regras e condições adequadas a cada caso proporcionada pelo poder de linguagens de programação de uso geral. A rede Ethereum, por exemplo, surge afirmando em sua proposta o intuito de oferecer uma rede que inclui uma linguagem de programação Turing-completa [34],

permitindo a qualquer interessado escrever *smart contracts* e aplicações descentralizadas criando seu próprio conjunto arbitrário de regras de propriedade, formatos de transação e funções de transição de estado.

3.5 Aplicações descentralizadas

Na criação de aplicações *web* tradicionais, que adotam o modelo de comunicação, transferência e processamento de dados através do modelo cliente-servidor, um notável desafio é conseguir lidar com a tendência do crescimento no volume de dados e a capacidade da estrutura executando a aplicação do lado do servidor de atender a sua demanda de acessos. O crescimento vertical é limitado ao avanço tecnológico no poder computacional da época, portanto a alternativa para conseguir atender a demanda do fluxo de informação é adotar o crescimento horizontal na estrutura responsável pela aplicação, adotando o que se define como uma aplicação distribuída.

Com o surgimento e consolidação das capacidades das redes *blockchain*, considerando fatores como a capacidade de lidar com grandes volumes de dados após a sua difusão e adoção massiva, o grande poder computacional resultante da junção dos inúmeros nós conectados que compõem redes consolidadas como o Ethereum, a Solana e a Cardano, além da flexibilidade de uso inovadora com a implementação da capacidade de escrever programas computacionais de uso geral através dos *smart contracts*, surge o conceito de aplicações descentralizadas, também chamadas *DApps* por conta da abreviação do termo em inglês *decentralized apps*. Vale destacar que existem diferenças do que se considera como aplicações distribuídas e aplicações descentralizadas, com o primeiro modelo utilizando da comunicação em rede para viabilizar a execução da aplicação em nós diversos da rede, escalando a sua capacidade de atender solicitações de usuários e aumentar sua capacidade computacional. Já no segundo modelo citado, a grande diferença se encontra na descentralização do poder de decisão por meio do modelo característico de consenso entres os nós encontrado em redes *blockchain*, ante a decisão centralizada no modelo de aplicações distribuídas, apesar da distribuição das responsabilidades de processamento e armazenamento [11].

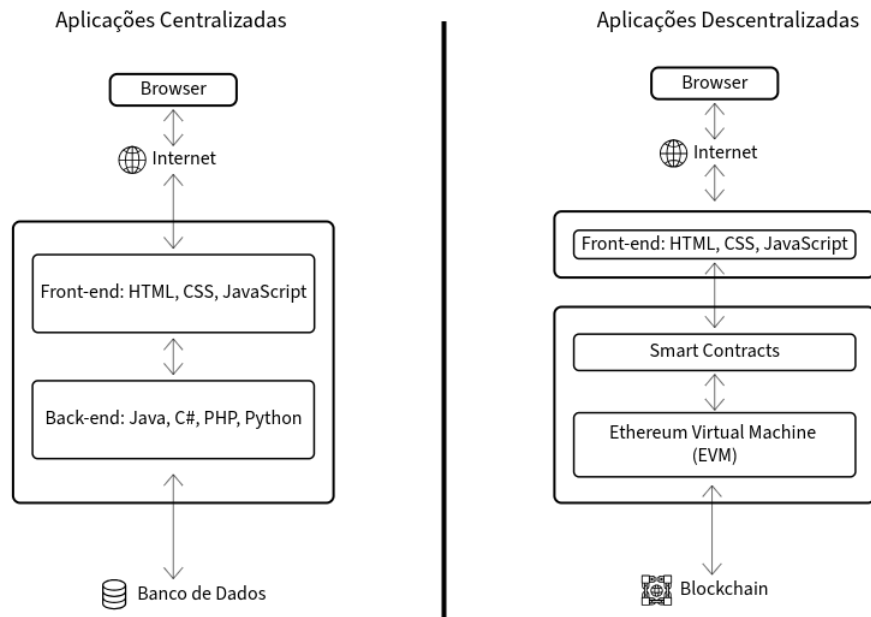


Figura 3.4: Comparativo entre aplicações centralizadas e descentralizadas (Adaptado) (Fonte: [11]).

Do ponto de vista da arquitetura das aplicações, as aplicações centralizadas ou distribuídas que utilizam o modelo cliente-servidor separam as aplicações no *front-end*, a parte que executa no dispositivos do usuário, tradicionalmente para aplicações *web* usando as linguagens HTML, CSS e JavaScript, e no *back-end*, aplicação executando nos servidores executando o armazenamento e os processamentos dos dados necessários escrito em linguagens como Java, C#, PHP ou Python. As aplicações descentralizadas adotam um modelo semelhante de *front-end* quando acessíveis por um navegador *web*, no entanto a implementação tradicional de *back-end* é substituída pelo uso de *smart contracts* que são executados em uma rede *peer-to-peer* do tipo *blockchain*, utilizando também essa estrutura para persistência dos dados. A Figura 3.4 ilustra as diferenças entre as arquiteturas de aplicações centralizadas e descentralizadas.

Capítulo 4

Tecnologias utilizadas

Este capítulo tem como objetivo descrever as tecnologias utilizadas no desenvolvimento da aplicação proposta, de forma a fundamentar as escolhas na sua organização de componentes e a arquitetura do *software*.

4.1 OpenStreetMap

O OpenStreetMap é uma aplicação do tipo SIG que contém um mapa gratuito de todo o planeta Terra, é um mapa editável por seus usuários apresentando uma base de dados geográficos alimentada majoritariamente por voluntários com uma licença de conteúdo aberto [6]. A plataforma foi criada como a intenção de oferecer uma alternativa gratuita de acesso a mapas por todo o mundo que tradicionalmente exigiam algum pagamento pelo seu acesso e restrições rígidas sobre seu compartilhamento e alterações, mesmo que para correção ou atualização de informações. Com a acessibilidade crescente a dispositivos com Global Positioning System (GPS) aliada às imagens áreas disponibilizadas pelo OpenStreetMap, diversos colaboradores da plataforma obtiveram um meio para a alimentação de mapas com grande precisão. A Figura 4.1 contém uma captura de tela da página do OpenStreetMap com o mapa de parte da região do Distrito Federal e Entorno.

Relations correspondem a agrupamentos de *nodes* e/ou *ways* para descrever elementos mais complexos no mapa. Exemplos de elementos que usam *relations* são: rotas de transporte que englobam diversas vias terrestres; delimitação de países, estados e municípios; áreas pertencentes a um mesmo elemento, mas que não sejam contíguas.

Tags todos os tipos básicos de elementos, *nodes*, *ways* e *relations*, podem ser associados a *tags* para descrever características particulares desses elementos.

O sistema de etiquetas é a base de classificação dos dados armazenados pela aplicação, sendo aberta para os usuários a criação livre de novos valores correspondentes às chaves do par de dados, apesar de ser encorajada a padronização para melhor organização das informações. As etiquetas são utilizadas pelo OpenStreetMap para classificar elementos como infraestruturas de acordo com a sua finalidade e função social, barreiras de locomoção, fronteiras administrativas, edifícios de construção humana, localidades de manufatura, serviços de emergência, composição geológica de uma área, uso do solo e vias terrestres e áreas. A Figura 4.2 exemplifica o sistema de etiquetas com as associadas à Universidade de Brasília (UnB) [36].

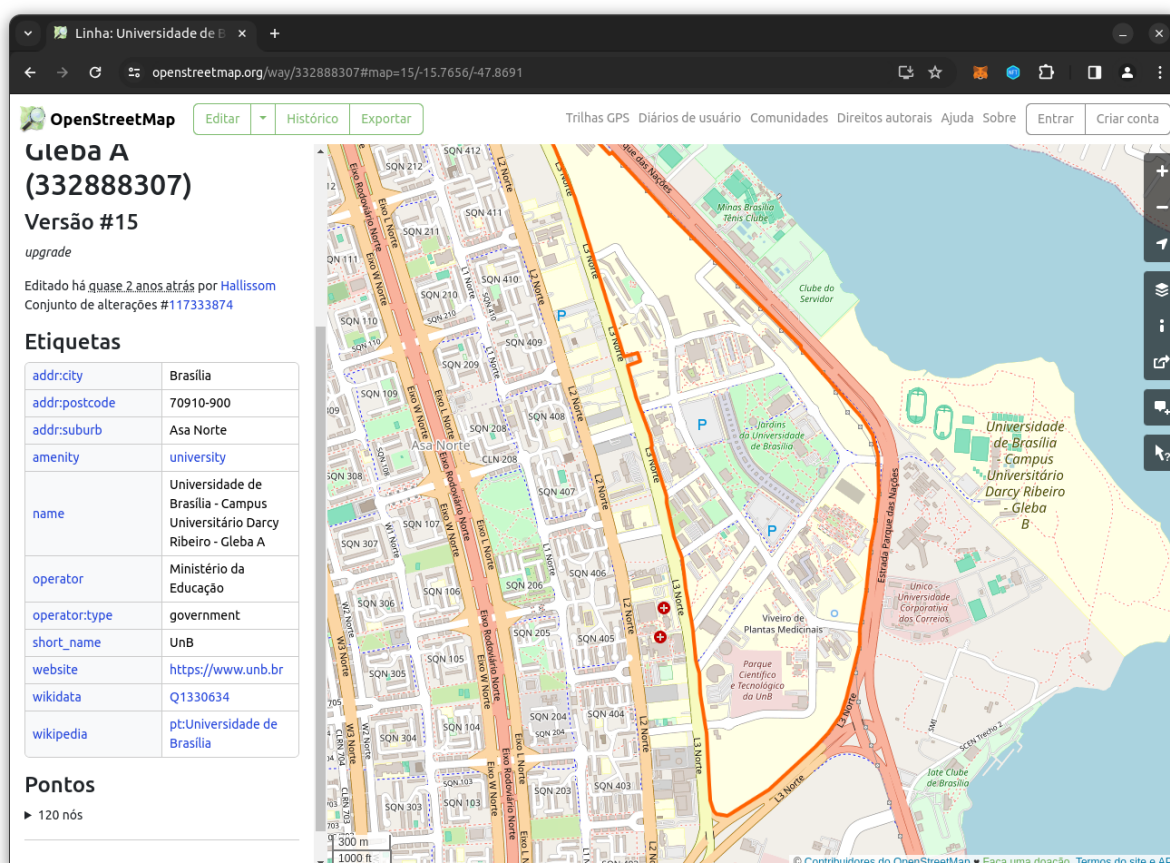


Figura 4.2: Etiquetas do OpenStreetMap associadas à UnB.

Uma das características mais notáveis do OpenStreetMap é a possibilidade de edição dos mapas contidos na sua base de dados por parte de usuários comuns. Com este objetivo, a própria página oficial do OpenStreetMap disponibiliza uma interface, representada na Figura 4.3 para criação, alteração e exclusão de elementos contidos nos mapas.

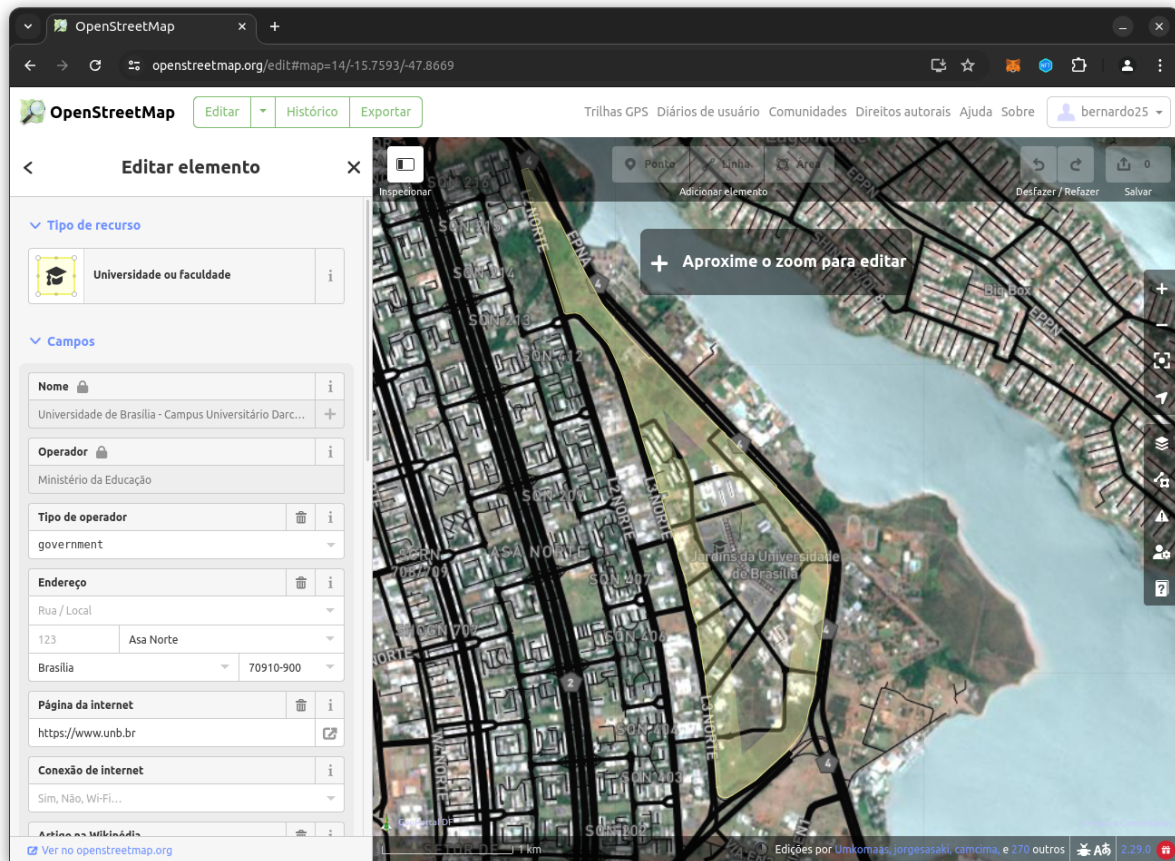


Figura 4.3: Tela de edição do OpenStreetMap.

Além do acesso livre para qualquer pessoa por meio do seu *website*, o OpenStreetMap disponibiliza acesso a sua base de dados por meio de diversos serviços de Application Programming Interface (API) [37] para o uso no desenvolvimento de outros *softwares* de forma gratuita. Dentre as APIs disponibilizadas cabe destacar duas delas, a API principal [38], sendo voltada a aplicações para editores do mapa, permitindo o acesso a funcionalidades voltadas à edição e inserção de informações na base de dados do OpenStreetMap e a Overpass API [39], que é uma API criada apenas para leitura de dados, desenvolvida como foco em performance para transferência de grandes volumes de informação utilizando uma linguagem de consulta específica.

4.2 Extensões do browser Google Chrome

O navegador de Internet Google Chrome permite a criação de extensões, que são pequenas aplicações que executam no ambiente do *browser* para adicionar novas funcionalidades durante a navegação na Web [40]. Dentre as capacidades que as extensões fornecem estão a personalização da interface do usuário, modificação do conteúdo em *websites* específicos, *scripts* de monitoração de eventos capazes de executar determinadas tarefas em reação a eles e até mesmo a criação de aplicações completas e independentes do endereço acessado na Web.

O desenvolvimento de extensões utiliza as mesmas linguagens para criação de páginas *web*, o HTML, o CSS e o JavaScript. Extensões de navegador tem acesso a APIs do navegador Google Chrome para uso de diversos recursos oferecidos como fazer requisições HTTP, acessar dispositivos de *hardware* do aparelho do usuário, ler e alterar elementos contidos no DOM da página acessada, ter comunicação e troca de dados com outras extensões.

Extensões que contenham um interface gráfica podem ser apresentadas na forma de janelas *pop-ups*, conteúdo no painel lateral nativo do navegador e também novas páginas inteiras como se fossem um *website*. Outros tipos de extensões são as que funcionam inteiramente em *background*, ou seja sem nenhuma exibição direta ao usuário, como extensões que adicionam novas funcionalidades a *websites* ou executam atividades com base no seu conteúdo. De forma resumida, extensões possuem três componentes capazes de executar funcionalidades:

Manifesto o único arquivo obrigatório e que declara os metadados da extensão, os recursos usados por ela, quais permissões são necessárias do navegador e quais componentes são executados em segundo plano.

Service workers são executados em segundo plano processando eventos do navegador, apesar de não possuírem acesso ao DOM da página visualizada.

Scripts de conteúdo executam códigos JavaScript no contexto de uma página *web*, podendo acessar e manipular o DOM da página em uso.

4.3 Rede Ethereum

A Ethereum é uma rede *blockchain* de segunda geração baseada no modelo de consenso Proof of Stake (PoS), apesar de inicialmente ter utilizado o modelo Proof of Work. Tem como uma das suas finalidade ser uma rede de transações financeiras utilizado a sua moeda digital Ether (ETH) [41]. Entretanto, a rede Ethereum também trás a proposta

de ser uma rede *blockchain* capaz de operar como um sistema computacional programável Turing-completo com seu suporte à execução de *smart contracts*.

A execução de códigos programáveis na rede acontece por meio da Ethereum Virtual Machine (EVM), um computador virtual que estado atual está copiado em todos os participantes da Ethereum, adicionalmente aos dados armazenados na sua *blockchain*. Qualquer um desses participantes da rede pode enviar uma solicitação por *broadcast* para acionar a EVM e executar alguma computação de dados. Os demais participantes recebem essa mensagem, verificam suas instruções, validam e posteriormente executam as instruções solicitadas [34].

Os programas escritos para ter sua execução pela Ethereum Virtual Machine são armazenados na *blockchain* da rede Ethereum na forma de *smart contracts* e como qualquer dado na *blockchain* é imutável, além de ter seu conjunto de instruções transparentes e auditáveis, mas na forma de *bytecode* que pode ser interpretado pela EVM em um ambiente isolado dos demais dados contidos na cadeia de blocos [42]. Esses *smart contracts* podem ser acionados a qualquer tempo mediante o pagamento de taxas usando a moeda Ether em caso de programas em que há a alteração de estado de forma permanente na *blockchain* ou necessite alguma computação de dados, mas gratuita para o acionamento de funcionalidades de mera consulta a informações já armazenadas.

4.3.1 Solidity

Efetivamente a Ethereum Virtual Machine (EVM) utiliza um conjunto de instruções próprio em *bytecode* para realizar computações, mas também existe uma linguagem de alto nível disponível para escrita de *smart contracts* na rede Ethereum, a Solidity. Esta é uma linguagem orientada a objetos, chamados como *Contracts* no contexto da linguagem, com tipagem estática, suporte a herança, modularização em bibliotecas e tipos complexos definidos pelo programador. É uma linguagem de uso geral para desenvolvimento de aplicações que podem ser executadas pela EVM. Sua sintaxe é inspirada por linguagens como C++, Python e JavaScript, adotando o padrão de construção de blocos de código delimitados por chaves.

4.4 MetaMask

O MetaMask é uma aplicação que funciona como carteira digital para o armazenamento de chaves privadas de contas na rede Ethereum, estando disponível em aplicações para dispositivos móveis e por meio de extensões de navegadores de internet [43]. Além da rede principal permite a conexão com redes de teste voltadas a desenvolvimento, redes

privadas e *sidechains*, redes *blockchain* separadas da rede principal, compatíveis com o Ethereum.

Por meio do MetaMask é possibilitado o acesso a realizar operações com a criptomoeda Ether e interação com *smart contracts* disponíveis na rede. Além do acesso direto pelo usuário, também estão disponíveis APIs [44] para acesso às contas do usuário por outras aplicações e páginas *web*, facilitando a interação com execução de contratos e leitura dos dados armazenados na *blockchain*.

4.5 InterPlanetary File System

O InterPlanetary File System (IPFS) é um sistema de arquivos descentralizado por uma rede *peer-to-peer*, utilizando diversos protocolos para endereçamento, roteamento e transferência de dados endereçados por conteúdo. A proposta desse sistema de protocolos é interligar os diversos sistemas conectados a um mesmo sistema de arquivos [45].

O conjunto de protocolos que são utilizados pelo IPFS foi inspirado em grande medida pelo sistema de distribuição de arquivos BitTorrent, também baseado em redes *peer-to-peer*, e o sistema de controle de versões Git. O endereçamento dos arquivos distribuídos na rede é realizado com base no seu conteúdo e organizado em Distributed Hash Tables (DHT) e acessíveis por meio de hiperlinks [46].

4.6 Tokens Não-Fungíveis

Dentre os avanços na utilização aliado ao desenvolvimento de novas funcionalidades disponibilizadas pelas redes *blockchain*, o Non-Fungible Tokens (NFT) surge como uma proposta de ser um formato de token indivisível em uma determinada rede que é utilizado como certificado de posse de determinado ativo, seja ele digital ou mesmo uma referência a um ativo físico.

A ideia dos NFTs é utilizar a imutabilidade de dados e registros na *blockchain* para oferecer um formato de posse digital de um bem único e não reproduzível com precisão dentro daquela determinada rede, assegurando sua autenticidade. Na rede Ethereum, o primeiro protocolo de Non-Fungible Tokens implementado foi o especificado pela ERC-721 [47].

Capítulo 5

Desenvolvimento

O presente capítulo tem por objetivo detalhar o processo de desenvolvimento do *software* proposto como solução ao problema abordado neste trabalho, seguindo a divisão em seções das seguintes etapas do processo de desenvolvimento de *software*: levantamento de requisitos, identificação dos casos de uso, análise e projeto da arquitetura da aplicação, modelagem dos componentes do sistema, integração dos componentes, seus testes e validações.

5.1 Descrição da aplicação

O GeoBlocos é uma aplicação descentralizada para criação e colaboração em projetos de mapeamento no OpenStreetMap. Qualquer usuário pode iniciar um novo projeto, definindo áreas e estruturas de interesse para atualização e aprimoramento dos dados geográficos. Os participantes que contribuem com edições no OpenStreetMap são incentivados por meio da geração de tokens do tipo Non-Fungible Tokens (NFT). Esses NFTs desempenham um papel de prova de trabalho, garantindo a autenticidade e imutabilidade das contribuições registradas na *blockchain* através de *smart contracts*.

5.2 Requisitos da aplicação

A aplicação tem como objetivos oferecer uma plataforma descentralizada de criação de projetos de mapeamento no OpenStreetMap. Qualquer usuário interessado pode criar um novo projeto de mapeamento de uma determinada região e definir quais estruturas de interesse naquela área atendem aos objetivos do projeto. Usuários mapeadores são incentivados a participar através da geração de tokens, do tipo Non-Fungible Tokens (NFT), com base nos dados contidos nos seus conjuntos de alterações enviados ao OpenStreetMap.

A criação desses tokens tem como objetivo providenciar aos criadores de projeto um meio de recompensar os participantes por suas contribuições na atualização de dados geográficos de suas regiões de interesse. A opção pelo uso de NFTs se justifica como prova de trabalho pela imutabilidade dos registros que são adicionados à *blockchain* utilizando *smart contracts*, além da posse por um único usuário de cada token, desta forma impedindo uma dupla recompensa por um mesmo trabalho após a transferência do usuário gerador da informação a um usuário criador de projetos de mapeamento.

5.3 Casos de uso

A modelagem de casos de uso é uma técnica amplamente utilizada no desenvolvimento de *software* para ilustrar os requisitos da aplicação a partir da representação da relação dos atores externos com as funcionalidades que a aplicação disponibiliza. Sua forma mais simples representa esses atores externos, ou usuários, como figuras-palito ligados a elipses que descrevem cada caso de uso [48]. Para esta aplicação foram considerados os atores do tipo mapeador e do tipo criador de projeto de forma que o desempenho de um papel não restrinja o desempenho do outro.

Os usuários mapeadores podem listar as suas contribuições no OpenStreetMap, criar tokens do tipo Non-Fungible Tokens a partir das suas contribuições, listar os seu tokens NFT criados e visualizar os projetos de mapeamento abertos. Usuários criadores de projetos podem usar o sistema para listar os projetos de mapeamento abertos no sistema, listar os projetos de mapeamento criados por ele e criar novos projetos. A representação de ambos os casos de uso está na Figura 5.1.

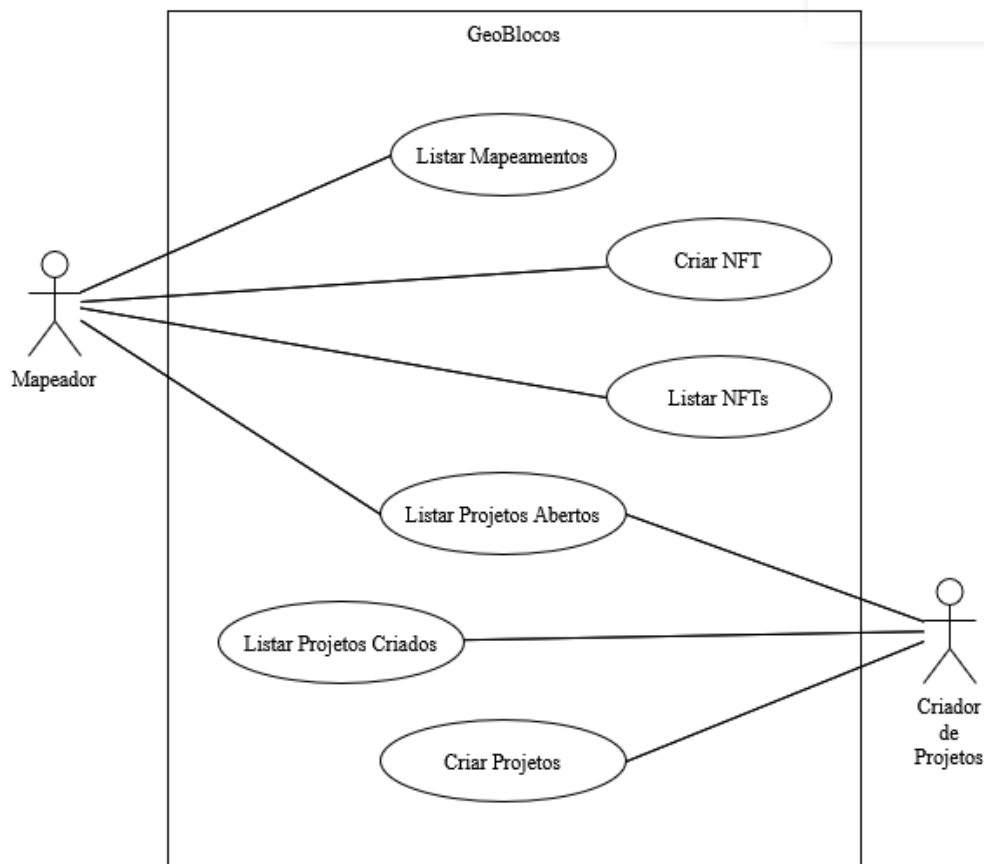


Figura 5.1: Casos de uso da aplicação GeoBlocos.

5.4 Desenho arquitetural da aplicação

Esta seção descreve a arquitetura da aplicação GeoBlocos em representações inspiradas pelas diretrizes e técnicas propostas por Hofmeister, Nord e Soni [49]. Entretanto para este trabalho foram criadas apenas as representações de acordo com a Visão Conceitual e a Visão de Módulos.

Visão Conceitual

De forma geral, a aplicação desenvolvida neste trabalho consiste em uma aplicação executada no computador do usuário e que se comunica com mais três elementos externos. O primeiro deles é a base de mapas e dados geográficos do OpenStreetMap, acessível por uma página *web* e diversas APIs. Os demais elementos são redes distribuídas em diversos computadores que compreendem a rede *blockchain* Ethereum, capaz de executar *smart*

contracts, e a rede IPFS, destinada ao armazenamento de arquivos. A Figura 5.2 contém um diagrama representando essa Visão Conceitual da aplicação.

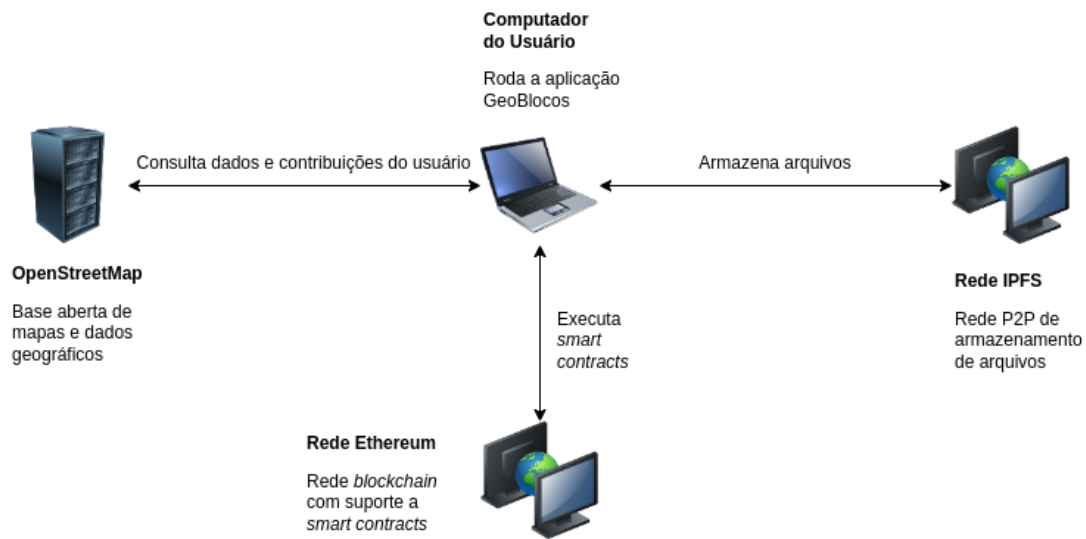


Figura 5.2: Arquitetura da aplicação sob a Visão Conceitual.

Visão de Módulos

As interações do usuário com a aplicação estão centralizadas em uma extensão para o *browser* Google Chrome ao acessar a página do OpenStreetMap. A extensão verifica as informações do usuário após o *login* no OpenStreetMap, utilizando as APIs oficiais da plataforma [37]. Para leitura de dados das contribuições desse usuário é utilizada a API de edição e para validação dos critérios de contribuições em relação aos projetos de mapeamento existentes é utilizada a API voltada a consulta de grandes volumes de dados, a *Overpass API*.

A persistência dos dados da aplicação acontece por meio de *smart contracts* na *blockchain* da rede Ethereum, com a interação do usuário com a rede acontecendo por meio da carteira digital de contas MetaMask. Ao acontecer alguma solicitação do usuário de interação com a *blockchain*, uma comunicação é aberta pela extensão com o MetaMask pedindo ao usuário para confirmar a transação utilizando sua carteira e respectivos fundos na rede.

A posse dos NFTs utilizado pela aplicação é gerenciada em um *smart contract* chamado MappingNFT seguindo o padrão ERC-721 [47]. Os metadados de cada token, bem como os dados brutos contidos no conjunto de alterações enviadas ao OpenStreetMap pelo usuário mapeador, são registrados na rede IPFS em um arquivo JSON por meio do provedor NFT.Storage [50].

Os projetos registrados na plataforma são armazenados em um *smart contract* que atua como banco de dados chamado ProjectDatabase. Cada projeto armazenado pelo contrato contém informações sobre o usuário do OpenStreetMap criador do projeto, seu nome descritivo, a delimitação da região a ser mapeada e as validações das classificações de dados geográficos aceitas por aquele projeto. Um diagrama da organização dos módulos da aplicação e suas interações com outros recursos está representado na Figura 5.3.

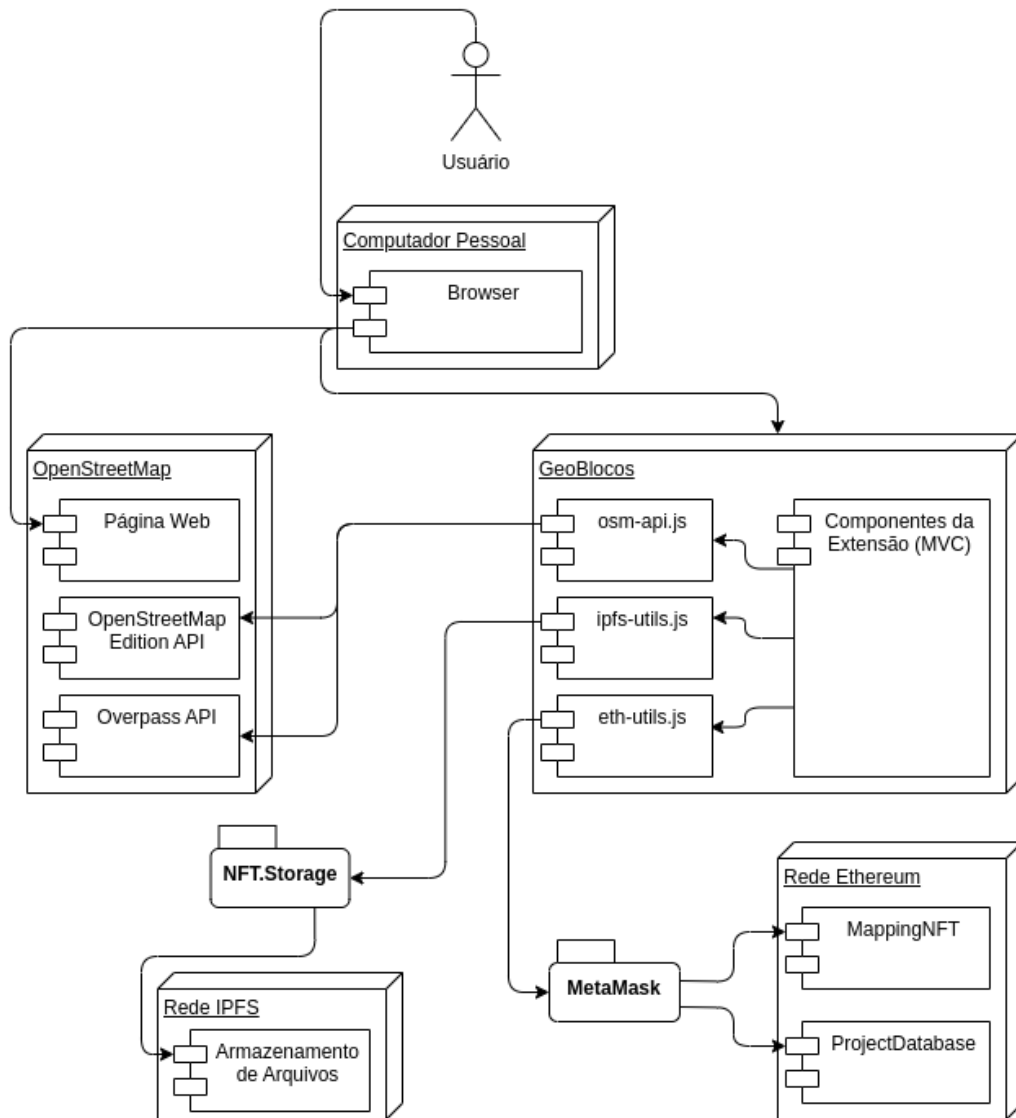


Figura 5.3: Arquitetura da aplicação sob a Visão de Módulos.

5.5 Detalhamento dos componentes

Esta seção descreve os detalhes dos componentes da aplicação através do diagrama de classe de cada componente nas subseções seguintes.

5.5.1 Extensão de navegador

Esta é a parte da aplicação em que há a interação com o usuário, consistindo em uma extensão para o navegador Google Chrome que deve ser instalada para uso do sistema. A organização dos componentes seguiu o padrão de arquitetura Model-View-Controller (MVC), esta arquitetura visa separar os componentes do sistema de forma que se permitam alterações de forma independente em camadas dos tipos Modelos, Visões e Controladores [51]. Uma representação gráfica da organização do modelo MVC está na Figura 5.4.

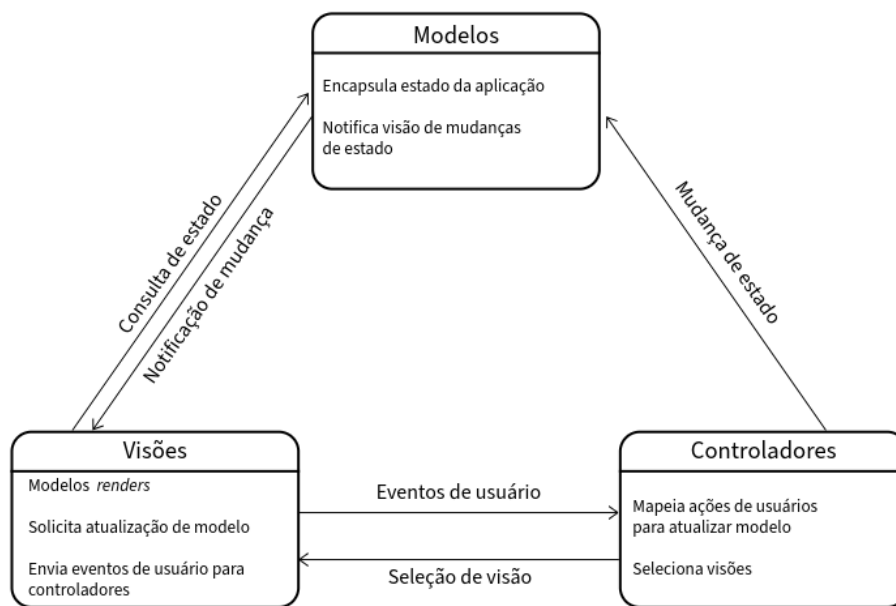


Figura 5.4: Organização do padrão de arquitetura MVC (Adaptado) (Fonte: [51]).

A Figura 5.5 detalha os componentes Model-View-Controller que integram a extensão de navegador para a aplicação.

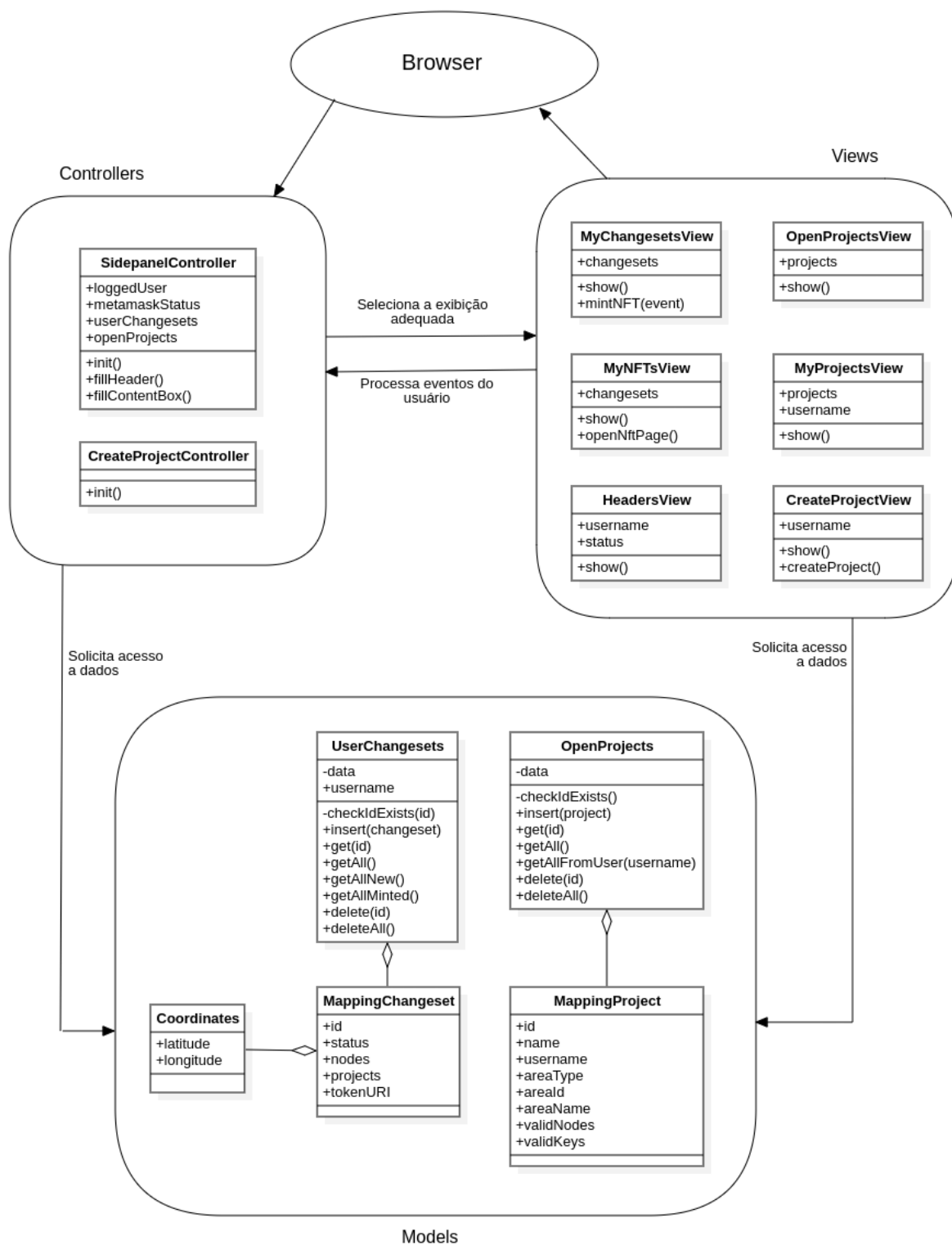


Figura 5.5: Arquitetura de componentes da extensão de navegador.

Abaixo estão as descrições textuais dos componentes representados na Figura 5.5.

SidepanelController controla o conteúdo a ser exibido na tela principal da aplicação de acordo com a seleção do usuário e acessa os dados necessários para aplicação.

CreateProjectController controla a exibição de conteúdo para a tela de criação de projetos de mapeamento.

HeadersView adiciona dinamicamente os elementos do cabeçalho na página principal da aplicação.

MyChangesetsView adiciona dinamicamente os elementos para exibição da tela Meus Mapeamentos.

MyNFTsView adiciona dinamicamente os elementos para exibição da tela Meus NFTs.

OpenProjectsView adiciona dinamicamente os elementos para exibição da tela Projetos Abertos.

MyProjectsView adiciona dinamicamente os elementos para exibição da tela Meus Projetos.

CreateProjectView adiciona dinamicamente os elementos para exibição da tela Criar Projeto.

Coordinates classe que representa uma coordenada geográfica pelos seus valores de latitude e longitude.

MappingChangeset classe que representa o *changeset* correspondente a uma alteração enviada pelo usuário para o OpenStreetMap e demais informações correlatas para uso da aplicação.

MappingProject classe que representa um projeto de mapeamento criado para a aplicação.

UserChangesets classe que representa uma coleção de objetos do tipo *MappingChangeset* para o usuário ativo na aplicação.

OpenProjects classe que representa uma coleção de objetos do tipo *MappingProject* que estejam ativos na aplicação.

5.5.2 Persistência de dados

Considerando o objetivo de construir uma aplicação que não dependa de qualquer estrutura computacional centralizada, a persistência dos dados do sistema é feita através de

smart contracts na *blockchain* da rede Ethereum. A aplicação utilizará dois *smart contracts*, sendo um para geração e gerenciamento da posse dos tokens NFTs criados pelos usuários e outro que atuará como um banco de dados para os projetos de mapeamento criados. A Figura 5.6 representa a modelagem para a criação desses contratos.

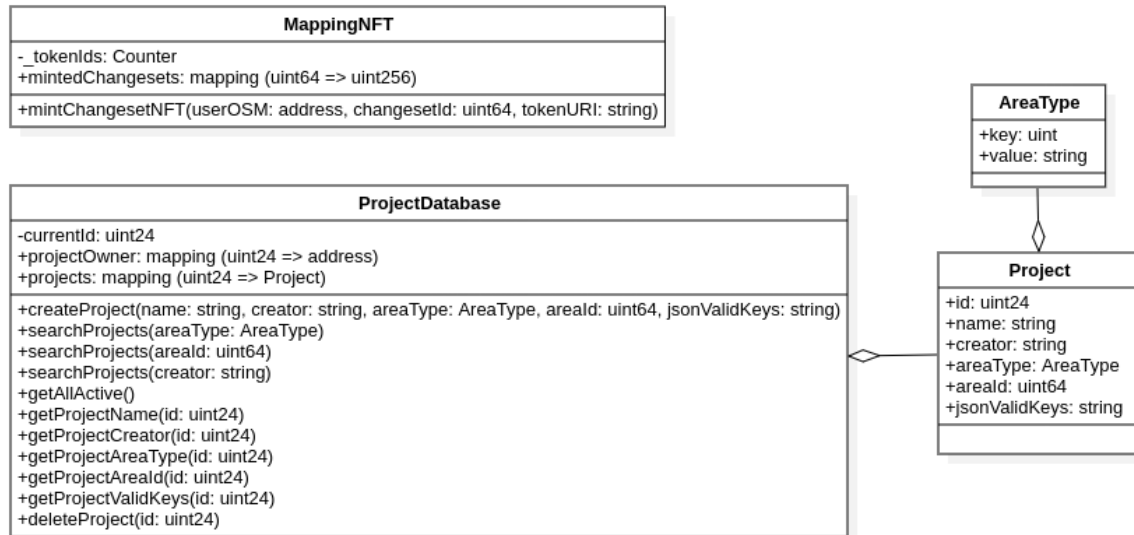


Figura 5.6: Modelagem dos *smart contracts* da aplicação.

Ambos os *smart contracts* utilizaram a linguagem de programação Solidity para implementação desses contratos na rede Ethereum e disponibilizar sua execução pela Ethereum Virtual Machine. A seguir temos mais detalhes sobre a implementação dos dois *smart contracts*:

MappingNFT contrato que gerencia os NFTs criados e utilizados pela aplicação, sendo uma implementação do padrão ERC-721 [47] baseada na versão disponibilizada pela biblioteca OpenZeppelin [52]. Armazena as informações sobre cada token NFT gerado junto a referência dos dados do token armazenado na rede IPFS e o endereço da carteira do proprietário na rede Ethereum. Todas as transações envolvendo a posse desses tokens são realizadas por esse contrato.

ProjectDatabase atua como um banco de dados para os projetos de mapeamento criados para a aplicação, armazenando informações sobre o nome do projeto, o nome de usuário do criador do projeto no OpenStreetMap, o tipo da área a ser mapeada, o código no OpenStreetMap identificador da área a ser mapeada e um campo flexível do tipo JSON de critérios de validação de contribuições no projeto.

5.6 Arquitetura do código

Nesta seção é apresentada a estrutura do código que compõe o *software* desenvolvido, a descrição dessa estrutura, além de incluir alguns dos trechos mais relevantes do código desenvolvido. A extensão de navegador que constitui a aplicação é organizada seguindo a estrutura de diretórios e arquivos listada na Figura 5.7.

```
app/
├── manifest.json
├── buttons
│   └── *.png
├── icons
│   └── *.png
├── scripts
│   ├── content-script.js
│   ├── service-worker.js
│   ├── controllers
│   │   ├── create-project.js
│   │   └── sidepanel.js
│   ├── models
│   │   ├── coordinates.js
│   │   ├── enums.js
│   │   ├── mapping-changeset.js
│   │   ├── mapping-project.js
│   │   ├── open-projects.js
│   │   ├── osm-attributes.js
│   │   └── user-changesets.js
│   ├── utils
│   │   ├── constants.js
│   │   ├── eth-utils.js
│   │   ├── ipfs-utils.js
│   │   └── osm-api.js
│   └── views
│       ├── common.js
│       ├── create-project.js
│       ├── header.js
│       ├── my-changesets.js
│       ├── my-nfts.js
│       ├── my-projects.js
│       └── open-projects.js
└── templates
    ├── color-theme.css
    ├── create-project.css
    ├── create-project.html
    └── not-logged.html
```

```
├── sidepanel.css
└── sidepanel.html
```

Figura 5.7: Estrutura de diretórios da aplicação.

Abaixo está a descrição dos componentes listados na árvore de diretórios apresentada anteriormente.

manifest.json arquivo de declaração dos metadados da extensão, permissões necessárias do navegador para a correta execução do programa e identificação dos *scripts* executados em segundo plano.

buttons/ diretório que agrupa os arquivos de imagem usados para botões da aplicação.

icons/ diretório que agrupa os arquivos de imagem usados para ícones da aplicação.

scripts/content-script.js *script* que acessa o DOM da página do OpenStreetMap para obter informações de identificação do usuário conectado.

scripts/service-worker.js este *script* é executado em segundo plano, monitorando eventos do navegador para controlar a ativação da extensão apenas na página do OpenStreetMap. Também é responsável por solicitar informações sobre o DOM da página *web* acessada para o **content-script.js**.

scripts/controllers/ diretório que concentra os componentes do tipo *Controller* usados pela aplicação.

scripts/models/ diretório que concentra os componentes do tipo *Model* usados pela aplicação.

scripts/utils/ diretório que contém *scripts* que implementam funcionalidades adicionais necessárias para a aplicação: configuração de parâmetros, utilização de recursos da rede Ethereum, armazenamento de arquivos na rede IPFS e acesso às APIs do OpenStreetMap.

scripts/views/ diretório que concentra os componentes do tipo *View* usados pela aplicação.

templates/ diretório que agrupa os arquivos HTML e CSS que define o *layout* base e estilização dos elementos da aplicação.

5.6.1 Smart contracts

O primeiro *smart contract* implementado para a aplicação é o responsável pela criação e armazenamentos de todos os dados necessários para os tokens NFTs. Sua implementação deve seguir as normas estabelecidas na ERC-721 [47], que determina as funcionalidades e metadados necessários para o uso de um NFT válido. Cada token armazenado consiste em um valor único referenciando o endereço de um arquivo de metadados armazenado na rede IPFS e que a posse esteja associada a um único endereço de carteira de ativos na rede Ethereum. O trecho de código na Figura 5.8 é a implementação utilizada pela aplicação, sendo baseada na implementação disponível pela biblioteca OpenZeppelin [52] adicionada de uma função para a criação de um novo NFT.

```
1  pragma solidity ^0.8.0;
2
3  import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
4  import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
5  import "@openzeppelin/contracts/utils/Counters.sol";
6  import "@openzeppelin/contracts/access/Ownable.sol";
7
8  contract MappingNFT is ERC721URIStorage, Ownable {
9      using Counters for Counters.Counter;
10     Counters.Counter private _tokenIds;
11     mapping(uint64 => uint256) public mintedChangesets;
12
13     constructor(address initialOwner)
14         ERC721("MappingNFT", "CST")
15         Ownable(initialOwner)
16     {}
17
18     function mintChangesetNFT(address userOSM, uint64 changesetId, string memory tokenURI)
19         public
20         returns (uint256)
21     {
22         require(mintedChangesets[changesetId] == 0, "Changeset already minted!");
23         _tokenIds.increment();
24
25         uint256 newTokenId = _tokenIds.current();
26         _mint(userOSM, newTokenId);
27         _setTokenURI(newTokenId, tokenURI);
28         mintedChangesets[changesetId] = newTokenId;
29
30         return newTokenId;
31     }
32 }
```

Figura 5.8: *Smart contract* de gerenciamento de NFTs.

O outro contrato implementado tem a finalidade de atuar com um banco de dados para os projetos de mapeamento criados no sistema. A implementação é uma coleção de uma estrutura de dados personalizada que representa um projeto de mapeamento e que contém as informações de nome do projeto, nome de usuário no OpenStreetMap do criador do projeto, o tipo de área que delimita a região a ser mapeada, o código identificador dessa área no OpenStreetMap e um campo JSON flexível contendo parâmetros de validação de contribuições ao projeto. O trecho de código reproduzido na Figura 5.9 ilustra as estruturas de dados necessárias para este *smart contract* e as funcionalidades de criação e exclusão de projetos.

```
1  pragma solidity ^0.8.0;
2
3  contract ProjectDatabase {
4      uint24 internal currentId = 0;
5      mapping (uint24 => address) internal projectOwner;
6      mapping (uint24 => Project) public projects;
7
8      enum AreaType {
9          Relation,
10         Way
11     }
12
13     struct Project {
14         uint24 id;
15         string name;
16         string creator;
17         AreaType areaType;
18         uint64 areaId;
19         string jsonValidKeys;
20     }
21
22     function createProject(string memory name,
23                           string memory creator,
24                           AreaType areaType,
25                           uint64 areaId,
26                           string memory jsonValidKeys)
27         public
28         returns (uint24)
29     {
30         require(bytes(name).length > 0, "Invalid project name.");
```

```

31     require(bytes(creator).length > 0, "Invalid creator name.");
32     require(areaId != 0, "Invalid area ID.");
33     require(bytes(jsonValidKeys).length > 1, "Invalid validation keys.");
34
35     uint24 id = getNextId();
36     projects[id] = Project(id, name, creator, areaType, areaId, jsonValidKeys);
37     projectOwner[id] = msg.sender;
38
39     return id;
40 }
41
42 function deleteProject(uint24 id)
43     public
44 {
45     require(id < currentId, "Project ID doesn't exist");
46     require(projectOwner[id] == msg.sender, "Sender isn't the creator of the project.");
47     Project memory oldProject = projects[id];
48
49     if (bytes(oldProject.name).length > 0) {
50         delete projects[id];
51     }
52 }
53 }

```

Figura 5.9: *Smart contract* do banco de dados de projetos de mapeamento.

5.6.2 Utilização da rede Ethereum

A comunicação da aplicação com a rede *blockchain* Ethereum ocorre por intermédio da carteira digital MetaMask através a solicitação de transações utilizando o endereço ativo no *browser* do usuário e interagindo com os endereços na rede Ethereum em que estão armazenados os *smart contracts*. O trecho de código na Figura 5.10 representa a solicitação de acessos à carteira MetaMask do usuário e a funcionalidade de criação de um novo token NFT.

```

1  const Web3 = require('web3');
2  const { createExternalExtensionProvider } = require('@metamask/providers');
3  const { storeChangesetNFT } = require('../ipfs-utils');
4  const NFT_CONTRACT = require('../data/artifacts/contracts/MappingNFT.sol/MappingNFT.json');
5  const { NFT_CONTRACT_ADRESS, ETHEREUM_NETWORK } = require('../app/scripts/utlis/constants');
6

```

```

7  let metamask;
8  let web3;
9
10 try {
11     metamask = createExternalExtensionProvider();
12     web3 = new Web3(metamask);
13 } catch (error) {
14     console.log(error);
15     web3 = undefined;
16     metamask = undefined;
17 }
18
19 async function isChangesetMinted(changesetId) {
20     try {
21         const mappingContract = new web3.eth.Contract(
22             NFT_CONTRACT.abi,
23             NFT_CONTRACT_ADRESS
24         );
25         const resp = await mappingContract.methods
26             .mintedChangesets(changesetId)
27             .call();
28
29         return resp !== 0n;
30     } catch (error) {
31         return false;
32     }
33 }
34
35 async function mintChangeset(changesetId) {
36     if (await isChangesetMinted(changesetId)) return false;
37
38     let isMinted = false;
39     const mappingContract = new web3.eth.Contract(
40         NFT_CONTRACT.abi,
41         NFT_CONTRACT_ADRESS
42     );
43     const ipfsChangesetNFT = await storeChangesetNFT(changesetId);
44
45     try {
46         await metamask.request({ method: 'eth_requestAccounts' });
47
48         if (metamask.selectedAddress !== null) {
49             const mintMethod = mappingContract.methods.mintChangesetNFT(
50                 metamask.selectedAddress,
51                 changesetId,
52                 ipfsChangesetNFT

```

```

53     );
54     const gasPrice = await web3.eth.getGasPrice();
55     const estimatedGas = await mintMethod.estimateGas();
56     const rawTx = {
57         from: metamask.selectedAddress,
58         to: NFT_CONTRACT_ADRESS,
59         gas: estimatedGas,
60         gasPrice: gasPrice,
61         data: mintMethod.encodeABI(),
62         chain: ETHEREUM_NETWORK,
63     };
64
65     await web3.eth
66         .sendTransaction(rawTx)
67         .on('transactionHash', (hash) => {
68             console.log(`Hash da transação: ${hash}`);
69         })
70         .on('receipt', (receipt) => {
71             console.log(receipt);
72             isMinted = receipt !== undefined;
73         })
74         .on('error', (error, receipt) => {
75             console.log(`Erro ao realizar transação: ${error}`);
76         });
77     }
78 } catch (error) {
79     if (error.code === 4001) {
80         console.log('Solicitação ao MetaMask não autorizada pelo usuário!');
81     }
82
83     console.log(`Erro ao tokenizar changeset (${changesetId}): ${error}`);
84 }
85
86 return isMinted;
87 }

```

Figura 5.10: Integração da aplicação com a rede Ethereum.

5.6.3 Armazenamento de arquivos na rede IPFS

O armazenamento de arquivo na rede IPFS acontece por meio do provedor NFT.Storage [50], sendo utilizado para armazenar o dados de um conjunto de alterações no OpenStreetMap

enviado pelo usuário e também os metadados que serão associados a um NFT. Cada arquivo enviado para a rede IPFS fica associado a um endereço único denominado Content Identifier (CID) que é gerado com base no conteúdo do arquivo, tal característica funciona com um mecanismo para evitar o armazenamento duplicado de arquivo, pois diferentes arquivos com conteúdos iguais resultam em um mesmo CID. O trecho de código na Figura 5.11 apresenta o funcionamento do armazenamento de um arquivo qualquer na rede IPFS.

```
1  const {
2    NFT_STORAGE_API_KEY,
3    OSM_API_URL,
4  } = require('../app/scripts/utils/constants');
5
6  async function storeBlobIPFS(blob) {
7    const response = await fetch('https://api.nft.storage/upload', {
8      method: 'POST',
9      headers: {
10        Authorization: `Bearer ${NFT_STORAGE_API_KEY}`,
11      },
12      body: blob,
13    })
14    .then((response) => response.json())
15    .catch(console.error);
16
17    if (response.ok)
18      return response.value.cid;
19    console.log(`Falha ao enviar arquivo: ${response}`);
20  }
```

Figura 5.11: Armazenamento de arquivos na rede IPFS.

Uma característica importante dos tokens NFT é ter um conjunto de metadados associados a cada token para melhor representar as suas características sem armazenar um volume de dados muito grande na rede *blockchain* Ethereum, onde o custo de armazenamento pode ser elevado, e utilizar a rede IPFS para a guarda desses metadados. A implementação do armazenamento de metadados utilizada pela aplicação está no trecho de código da Figura 5.12.

```
1  async function storeChangesetNFT(changesetID) {
2    const changesetOriginUrl = `${OSM_API_URL}/changeset/${changesetID}.json`;
```

```

3  const changesetFile = await getChangesetFile(changesetOriginUrl);
4  const changesetIPFS = await storeChangesetIPFS(changesetFile);
5  const user = changesetIPFS.user;
6  const userID = changesetIPFS.userID;
7
8  const nftMetadata = {
9    title: 'Mapping NFT Metadata',
10   type: 'object',
11   properties: {
12     name: {
13       type: 'string',
14       description: `OpenStreetMap changeset: ${changesetID}.`,
15     },
16     description: {
17       type: 'string',
18       description: `Changeset from user (uid): ${user} (${userID}).`,
19     },
20     image: {
21       type: 'string',
22       description:
23         `ipfs://bafkreie4cqltlwljtmccztypjm2wrw6zg6rpxk3dgna4gcxwjiaqvgya`,
24     },
25     content: {
26       type: 'string',
27       description: `ipfs://${changesetIPFS.cid}`,
28     },
29     origin: {
30       type: 'string',
31       description: `${OSM_API_URL}/changeset/${changesetID}.json`,
32     },
33   },
34 };
35
36 const nft = new Blob([JSON.stringify(nftMetadata)], {
37   type: 'application/json',
38 });
39 const nftCID = await storeBlobIPFS(nft);
40
41 return `ipfs://${nftCID}`;
42 }

```

Figura 5.12: Armazenamento dos metadados dos NFTs.

5.6.4 Acesso à base de dados do OpenStreetMap

O banco de dados que contém as informações geográficas utilizadas pelo OpenStreetMap tem seu acesso livre por meio de diversas APIs. Duas delas são utilizadas para acesso aos dados necessários para a aplicação, a API v0.6 [38], que disponibiliza funcionalidades para aplicações de editores de mapas, e a Overpass API [39], destinada apenas a consultas e otimizada para trafegar grandes volumes de dados. O uso dessas APIs é feito por meio de requisições HTTP como exemplificado no recorte do código da aplicação na Figura 5.13 que é utilizado para verificar os conjuntos de alterações enviados pelo usuário e verificando se já existe um NFT criado para aquele conjunto.

```
1 import { OSM_API_URL } from '../utils/constants.js';
2 import { ChangetsetStatus } from '../models/enums.js';
3 import { MappingChangeset } from '../models/mapping-changeset.js';
4 import { UserChangesets } from '../models/user-changesets.js';
5
6 export async function fetchUserChangesets(username, projects) {
7   const userChangesets = new UserChangesets(username);
8   const changesetsQueryURL = `${OSM_API_URL}/changesets?display_name=${username}`;
9   const response = await fetch(changesetsQueryURL, {
10     method: 'GET',
11     headers: {
12       Accept: 'application/json',
13     },
14   });
15
16   try {
17     const responseJSON = await response.json();
18
19     userChangesets.deleteAll();
20
21     for (const changeset of responseJSON.changesets) {
22       const nodes = await fetchChangesetNodes(changeset.id);
23       const acceptedProjects = listAcceptedProjects(nodes, projects);
24
25       if (acceptedProjects.length > 0) {
26         const mappingChangeset = new MappingChangeset(
27           changeset.id,
28           (await ethutils.isChangesetMinted(changeset.id))
29             ? ChangetsetStatus.Minted
30             : ChangetsetStatus.New,
31           nodes,
32           acceptedProjects
33         );
34
```

```

35         userChangesets.insert(mappingChangeset);
36     }
37 }
38
39     return userChangesets;
40 } catch (error) {
41     console.log(`erro ao consultar changesets: ${error}`);
42 }
43 }

```

Figura 5.13: Integração da aplicação com as APIs do OpenStreetMap.

5.7 Integração e testes

O processo de testes em um *software* objetiva verificar se um programa desenvolvido consegue atender ao que foi proposto em seus requisitos e também verificar a existência de erros e comportamentos inesperados durante o uso do sistema, sendo este um processo em que ocorre a entrada de dados fictícios no sistema e avaliação dos dados de saída gerados após o processamento. Pode-se classificar os testes baseados em seus objetivos em duas categorias distintas: os testes de validação, onde é verificado se o *software* implementado atende aos requisitos especificados no início do desenvolvimento, e os testes de verificação, onde se busca a existência de comportamentos inesperados como gerar resultados incorretos ou diferentes do especificado para o projeto [53].

5.7.1 Ferramentas de teste

Para os testes da parte do sistema em que há a interação com o usuário, ou seja, a extensão para o navegador Google Chrome onde está a interface gráfica de uso do *software*, foi utilizado o “Modo de desenvolvedor” disponível no próprio navegador para instalação da extensão em desenvolvimento e seu uso para testes. O uso dessa ferramenta foi útil tanto para a realização dos testes de validação de requisitos e visualização dos elementos gráficos da aplicação quanto para os testes de verificação de erros através de saída de dados pelo console JavaScript embutido no navegador.

Inicialmente, os *smart contracts* utilizados na aplicação foram escritos e testados em uma IDE própria para o desenvolvimento de contratos na rede *blockchain* Ethereum, a Remix Online IDE [54], reproduzida na Figura 5.14. Essa IDE foi escolhida para o desenvolvimento dos contratos da aplicação por sua possibilidade de testes em rede

blockchain local e diferentes redes de testes da Ethereum [55]. Após a definição do código e realização dos testes necessários em rede local, foi utilizada a rede de testes Sepolia [56] para a implementação e integração com os demais componentes do sistema.

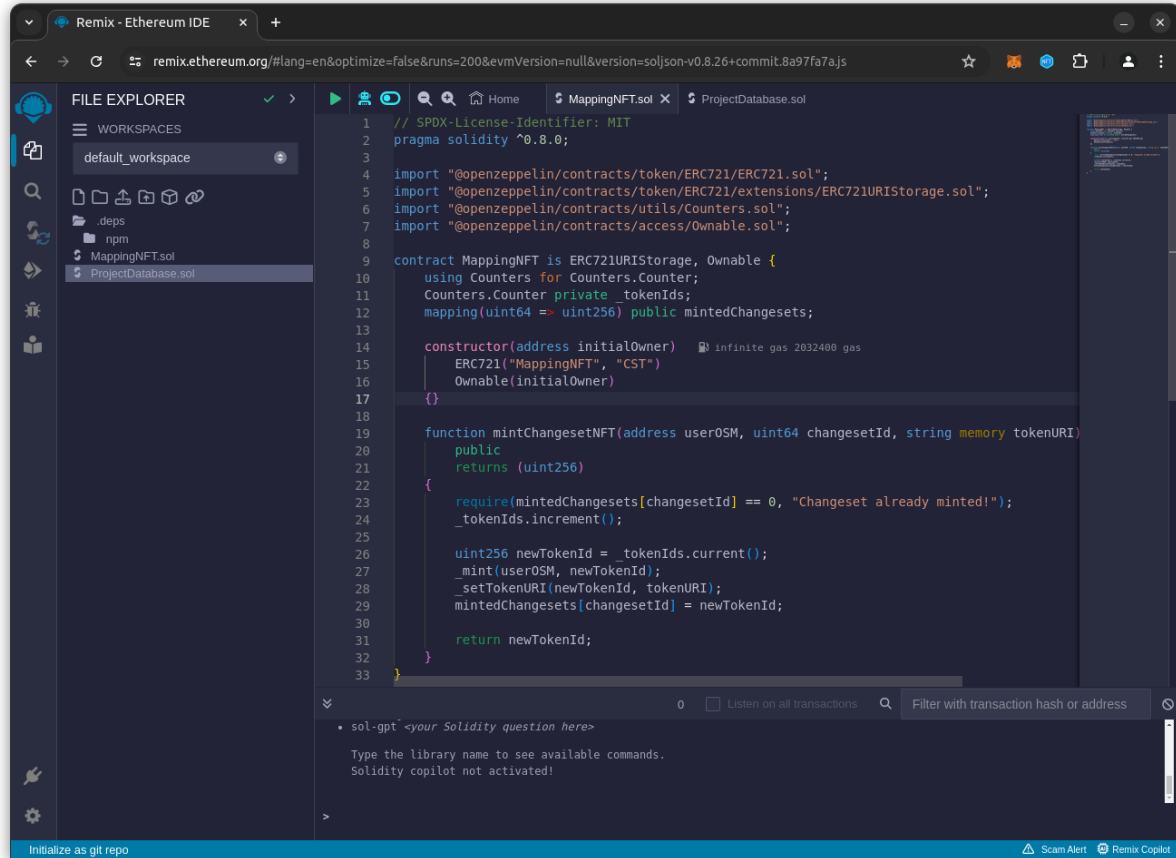


Figura 5.14: Ambiente de desenvolvimento Remix Online IDE (Fonte: [54]).

5.7.2 Testes de validação

Durante os testes de validação foram realizadas inspeções das funcionalidades definidas nos casos de uso do projeto para ambos os atores externos definidos, o usuário *Mapeador* e o usuário *Criador de Projeto*.

Para os casos de uso *Listar Mapeamentos* e *Listar NFTs* foi verificada se as contribuições do usuário *Mapeador* listadas na extensão correspondiam às informadas pelo perfil do usuário no OpenStreetMap, além da separação entre as contribuições em que já ocorreu a criação de um NFT no caso de uso *Listar NFTs* e todas as demais contribuições sendo exibidas apenas na funcionalidade de *Listar Mapeamentos*. No caso de uso *Criar NFT* foi verificada a ocorrência de solicitação para a carteira digital MetaMask de autorização pra efetivação da transação e, se aprovada, ocorria a alteração de estado no *smart contract*

responsável pelo gerenciamento dos tokens NFT. Por fim, foi verificada a equivalência entre os projetos de mapeamento armazenados pelo *smart contract* que atua como banco de dados para esses elementos e os listados na interface de usuário para o caso de uso *Listar Projetos Abertos*.

Para o ator *Criador de Projeto* foi verificada a correta filtragem na listagem de todos os projetos de mapeamento abertos no sistema, para o caso de uso *Listar Projetos Abertos* compartilhado com o usuário *Mapeador*, e apenas os criados pelo usuário autenticado no OpenStreetMap no caso de uso *Listar Projetos Criados*. A validação do caso de uso *Criar Projetos* foi feita por meio da verificação na alteração dos dados do *smart contract* de base de dados para projetos após a solicitação de criação de projeto na interface de usuário e autorização da transação na carteira MetaMask.

5.7.3 Testes de verificação

As verificações de erros na aplicação foram realizados através da entrada de dados fictícios para os componentes desenvolvidos especificamente para o sistema e avaliação dos resultados produzidos. Já para os elementos do sistema que interagem com sistemas de dados externos a abordagem utilizada foi a comparação entre os dados reproduzidos no sistema e os encontrados nesses sistemas externos.

Os primeiros testes realizados foram o uso em rede local dos *smart contracts* desenvolvidos através da IDE Remix [54], avaliando entradas para todas as funções implementadas e os resultados produzidos ou se houve a alteração desejada no estado do contrato. Também nos estados iniciais de desenvolvimento foram realizados testes nos componentes de armazenamento de arquivos na rede IPFS e comparação entre o conteúdo de entrada e o que ficou armazenado na rede de arquivos.

Em seguida os testes necessários foram os de uso da aplicação na extensão de navegador, autenticando no OpenStreetMap e efetuando a comparação dos dados apresentados na extensão com os dados listados pelo OpenStreetMap. Após o *deploy* dos *smart contracts* desenvolvidos para a rede de testes Sepolia [56] foram testados a criação de NFTs, solicitando a autorização das transações na carteira MetaMask e monitorada a alteração de estado dos contratos na rede de testes. Procedimento similar foi realizado para testar a criação de projetos de mapeamento no sistema, verificando se entradas propositalmente indevidas eram identificadas e rejeitadas pelo sistema e também se os dados apresentados para o usuário eram correspondentes aos armazenados na rede *blockchain* pelo contrato responsável.

Capítulo 6

GeoBlocos

6.1 Apresentação visual do protótipo

Aplicação é acessada, após a instalação de sua extensão ao navegador Google Chrome, com o acionamento de um painel lateral que fica disponível apenas no endereço do OpenStreetMap (<https://www.openstreetmap.org/>) e com o *login* de um usuário cadastrado na plataforma. Apresenta um cabeçalho contendo o nome de usuário do OpenStreetMap conectado e o status de conexão com a carteira MetaMask. O conteúdo a ser exibido é acessado por meio de um seletor localizado abaixo do cabeçalho com as informações básicas. As opções de conteúdo disponível são detalhadas nas seções posteriores.

6.1.1 Tela Meus Mapeamentos

Apresentação das contribuições enviadas pelo usuário conectado ao OpenStreetMap. A visualização compreende uma listagem dos conjuntos de alterações enviados pelo usuário que atendam a pelo menos um dos projetos de mapeamento abertos na plataforma. Os cartões de visualização de mapeamentos contém um título, clicável para maior detalhamento, com o ID do conjunto de alterações, um botão de criação de um Non-Fungible Tokens (NFT) a partir dos dados daquela contribuição, uma listagem dos projetos que aquela contribuição atende e a região englobada pelo projeto. A Figura 6.1 representa a tela de visualização das contribuições do usuário.

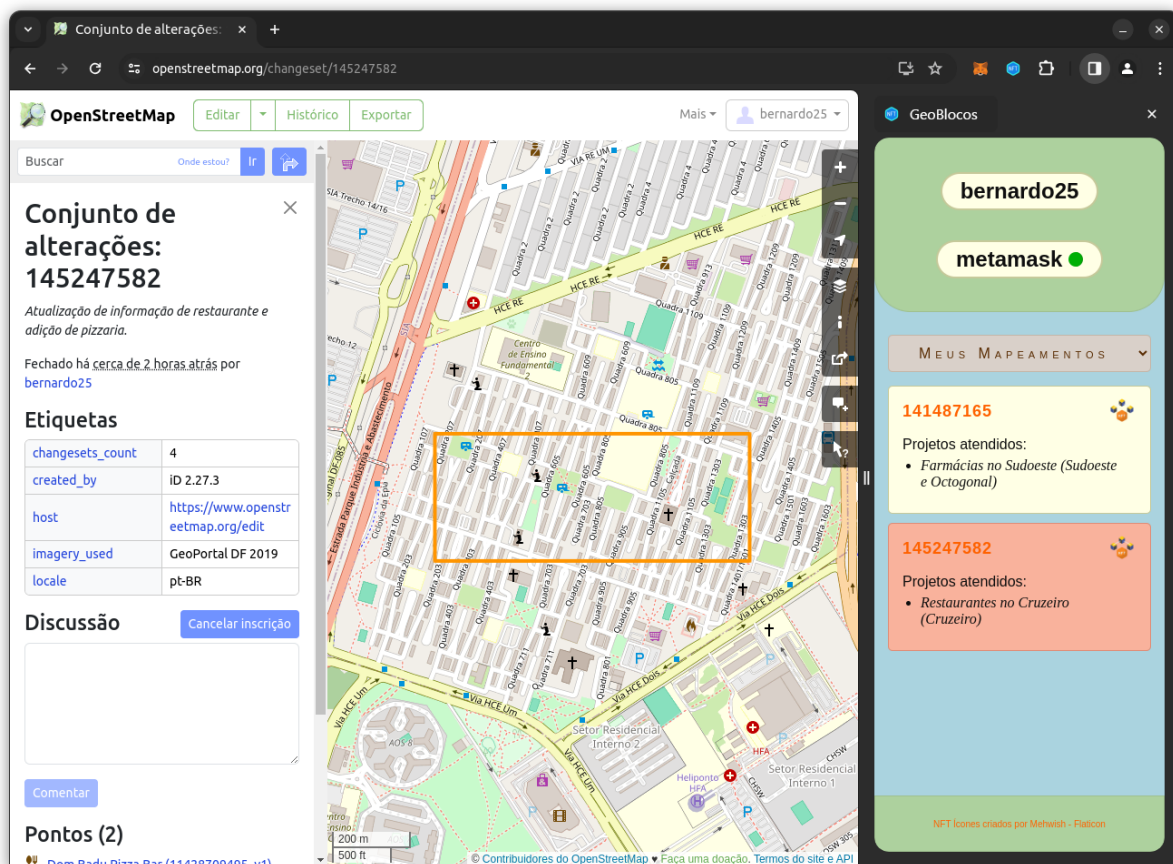


Figura 6.1: Tela de mapeamentos criados pelo usuário.

Ao clicar no botão de criação de NFT, é enviada ao usuário uma solicitação de autorização da transação com o *smart contract* responsável por criar, registrar a posse e armazenar os tokens da aplicação. A Figura 6.2 exemplifica essa interação.

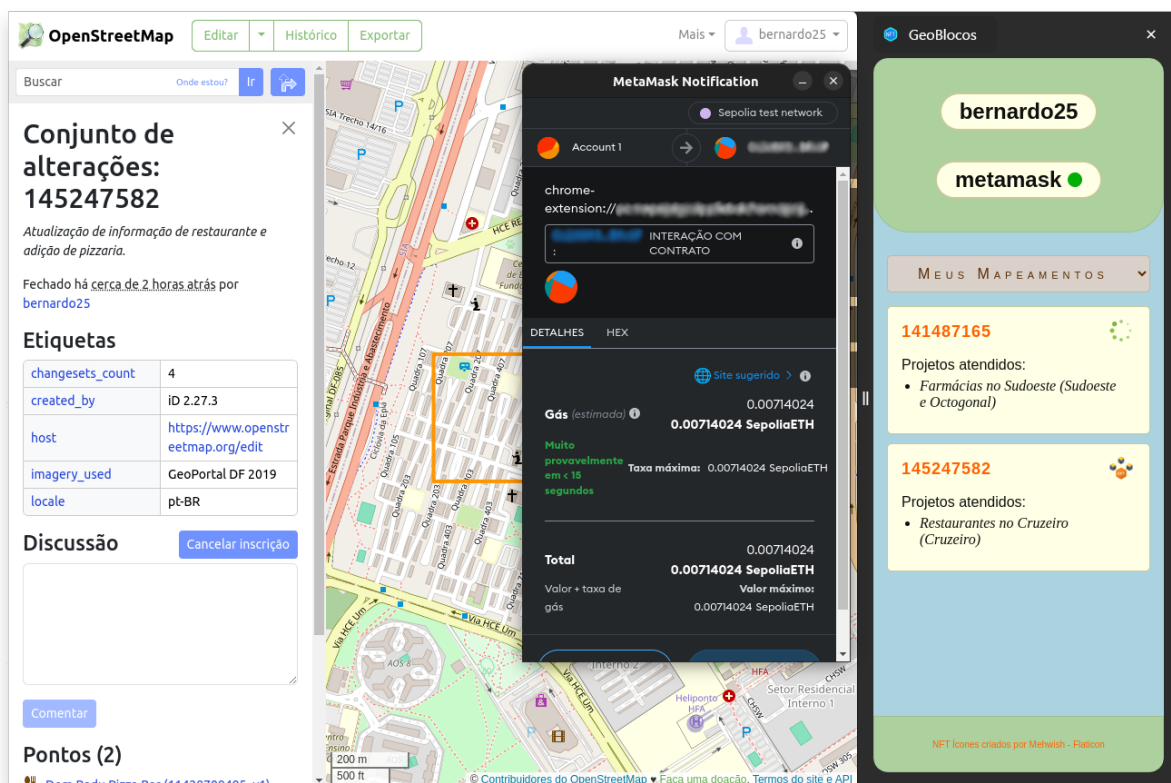


Figura 6.2: Pop-up de solicitação de confirmação de criação de NFT.

6.1.2 Tela Meus NFTs

Tela que representa a listagem dos NFTs criados à partir das contribuições do usuário conectado. Os tokens são listados no formato de cartões de visualização contendo um título, clicável para maior detalhamento, com o ID do conjunto de alterações, um botão de visualização dos dados daquele Non-Fungible Tokens (NFT), uma listagem dos projetos que aquela contribuição atende e a região englobada pelo projeto. A representação da tela está na Figura 6.3.

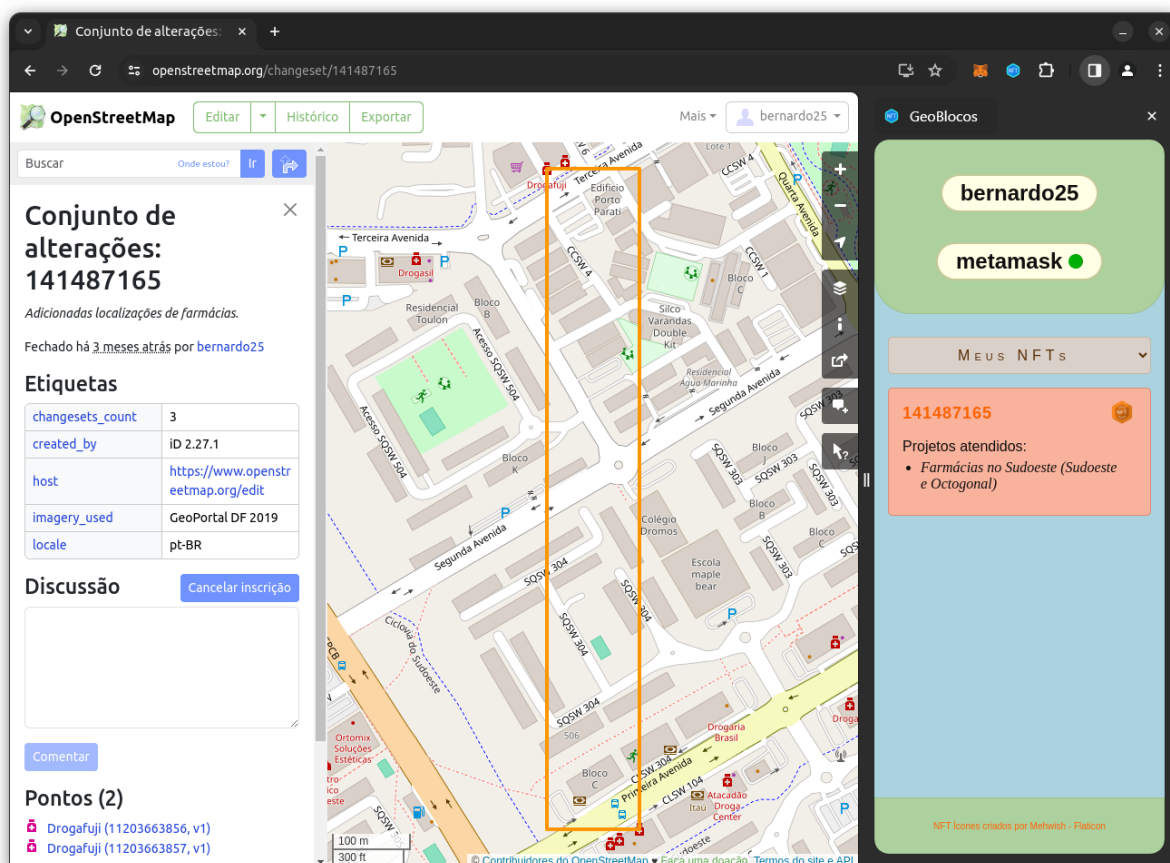


Figura 6.3: Tela de NFTs criados à partir dos mapeamentos do usuário.

6.1.3 Tela Projetos Abertos

Visualização na aplicação dos projetos de mapeamentos que estão abertos para novas contribuições na plataforma. A listagem ocorre no formato de cartões contendo no título o nome descritivo do projeto, a área que delimita as contribuições a serem aceitas, a listagem dos atributos em forma de *tags* do OpenStreetMap que são considerados válidos dentro do projeto e a informação do usuário do OpenStreetMap criador daquele projeto. Também está apresentado um campo de pesquisa de projetos, filtrando de acordo com o nome do projeto ou região a ser mapeada. A Figura 6.4 representa a visualização descrita.

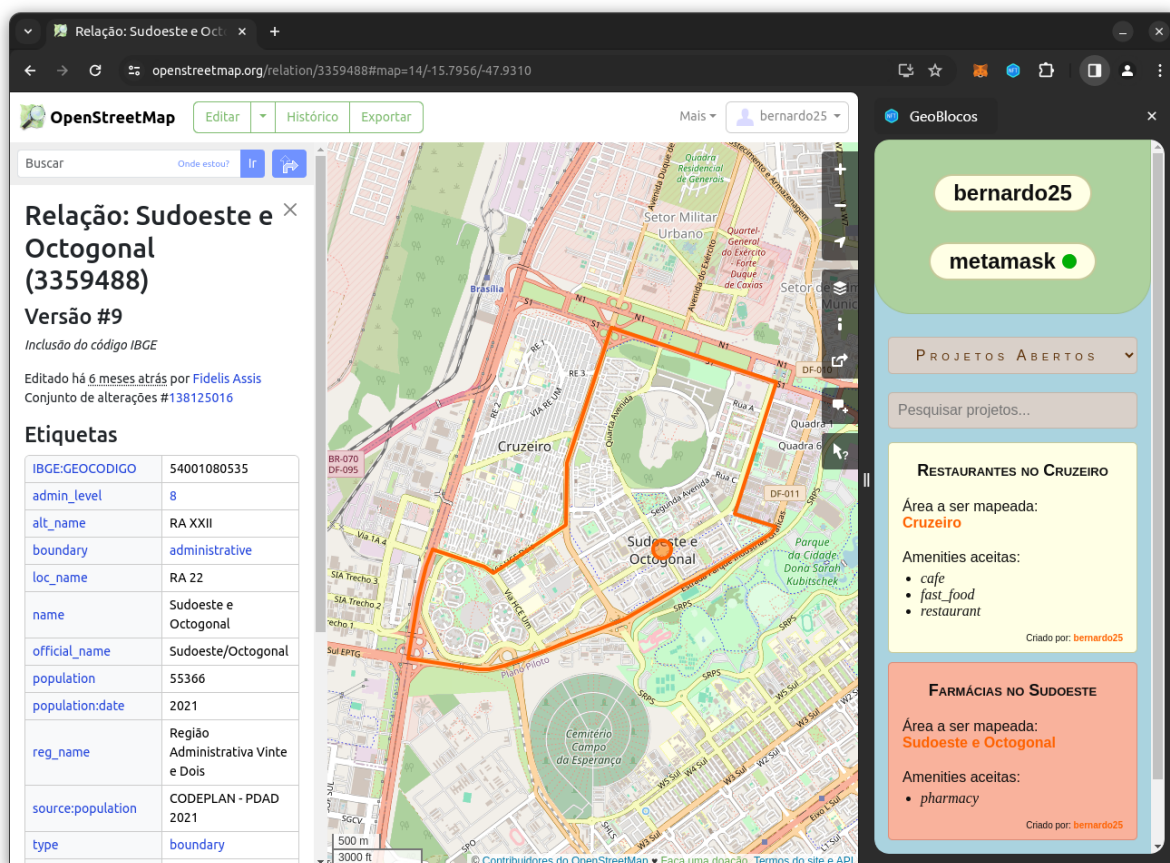


Figura 6.4: Tela de projetos de mapeamento abertos na aplicação.

6.1.4 Tela Meus Projetos

Tela da aplicação correspondente a projetos de mapeamento criados pelo usuário atualmente conectado ao OpenStreetMap. A listagem ocorre no formato de cartões contendo no título o nome descritivo do projeto, a área que delimita as contribuições a serem aceitas, a listagem dos atributos em forma de *tags* do OpenStreetMap que são considerados válidos dentro do projeto. Também estão apresentados um campo de pesquisa de projetos, filtrando de acordo com o nome do projeto ou região a ser mapeada, e um botão de criação de novos projetos. Uma representação da tela da aplicação descrita está na Figura 6.5.

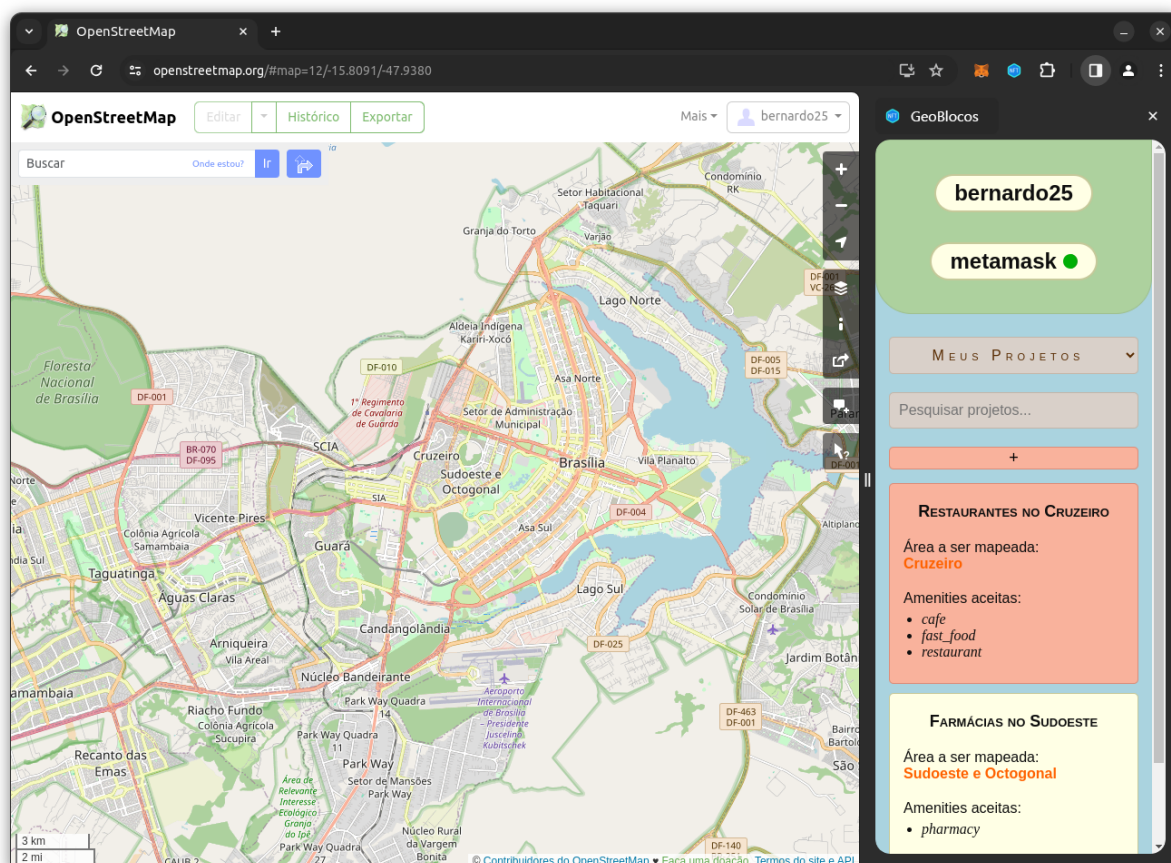


Figura 6.5: Tela de consulta dos projetos criados pelo usuário.

6.1.5 Tela Criar Projeto

Tela de criação de um novo projeto de mapeamento. Composta por um formulário contendo: um campo de definição de um nome descritivo ao projeto; um campo seletor do tipo de área a ser delimitada para os mapeamentos de acordo com o tipo de dados geográficos utilizados pelo OpenStreetMap, sendo aceitos *relations* e *ways*; um campo de inserção do ID utilizado pela área selecionada na base de dados do OpenStreetMap; e campos de seleção de valores considerados válidos para *tags* descritivas dos elementos contidos nas contribuições ao projeto. A tela citada está representada na Figura 6.6.

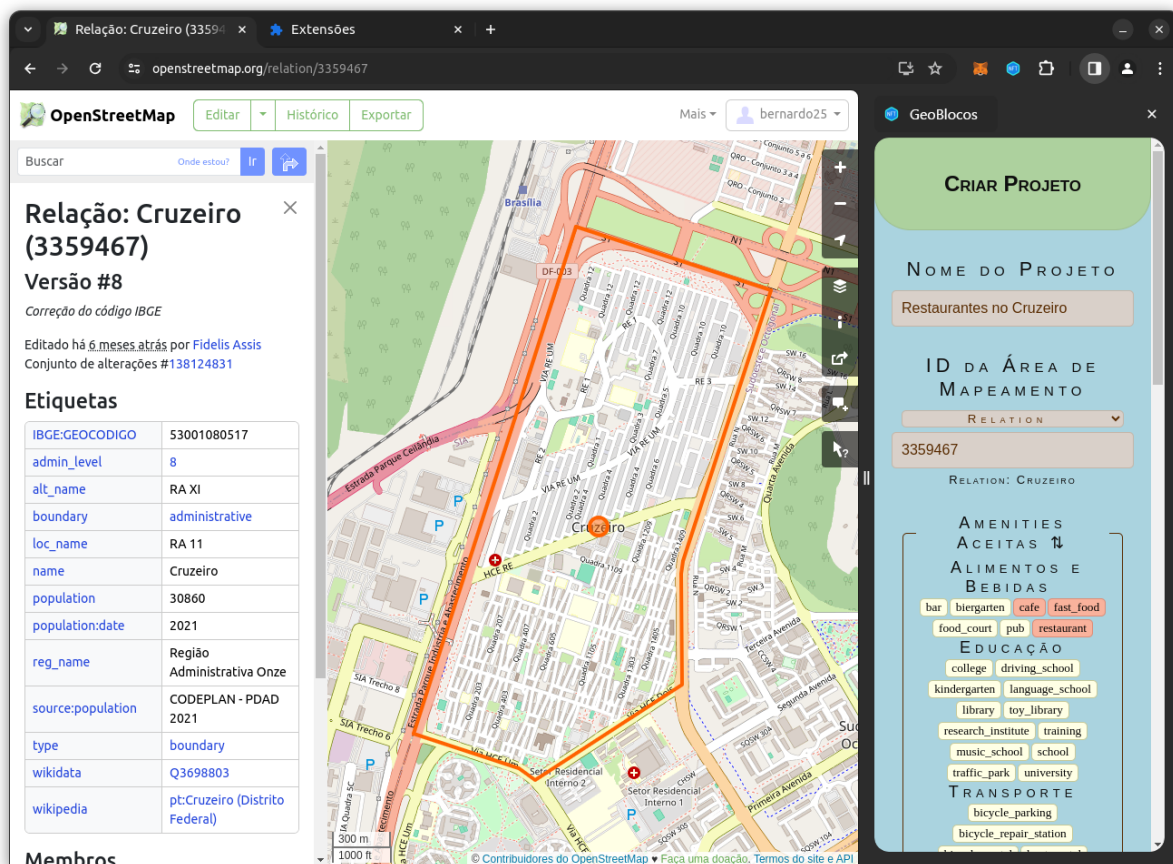


Figura 6.6: Tela de criação de projetos de mapeamento.

6.2 Instalação da aplicação

1. Fazer o download do arquivo de instalação da extensão no endereço: <https://github.com/erikbrto/geoblocos/blob/main/dist/GeoBlocs.crx>.

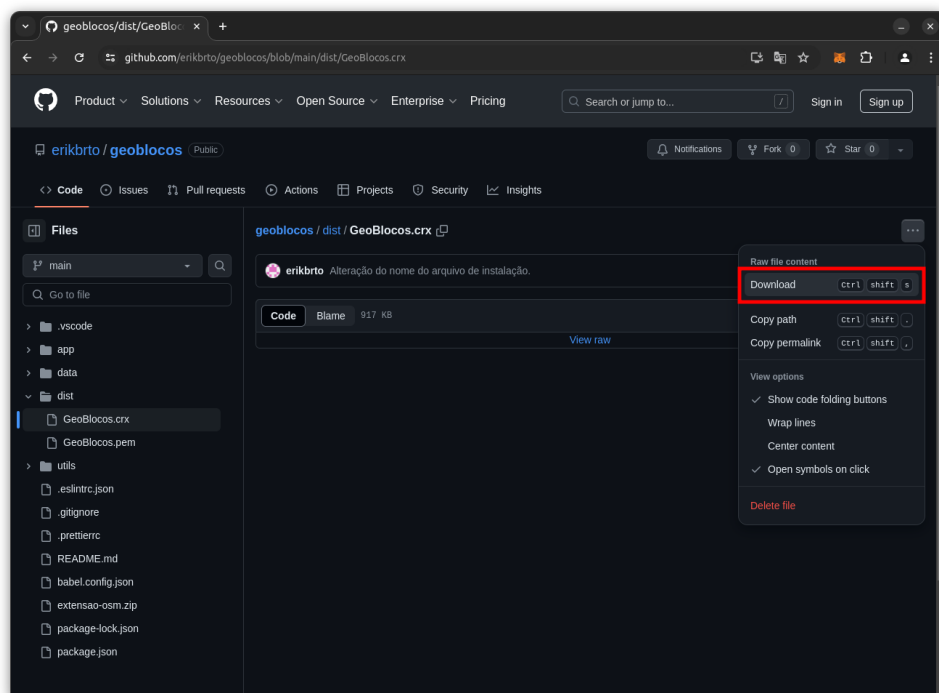


Figura 6.7: 1º passo de instalação.

2. Acessar a funcionalidade **Gerenciar extensões** no navegador Google Chrome.

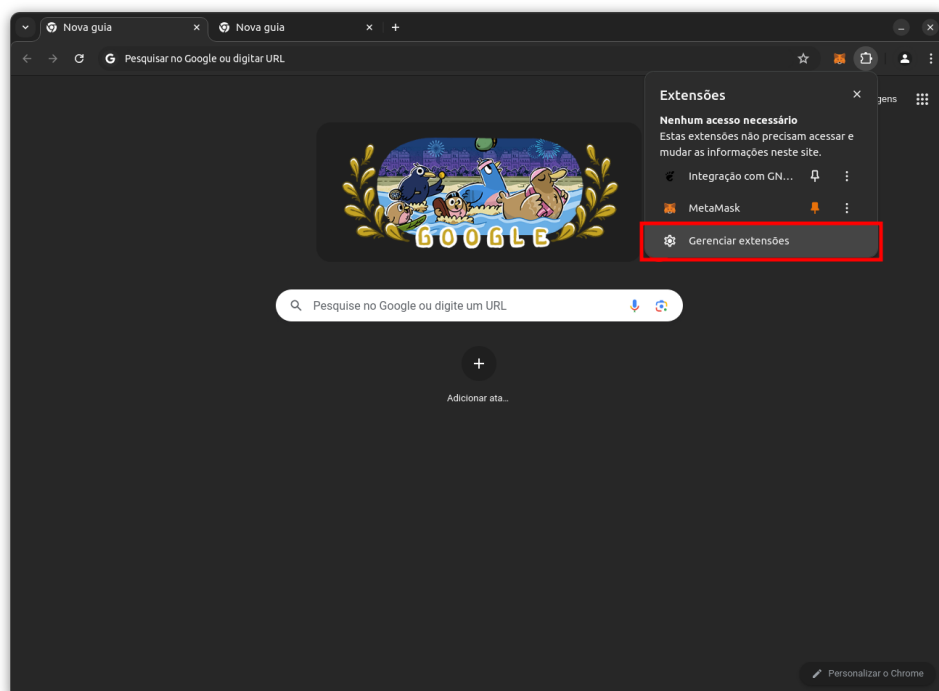


Figura 6.8: 2º passo de instalação.

3. Ativar a opção **Modo do desenvolvedor** no navegador.

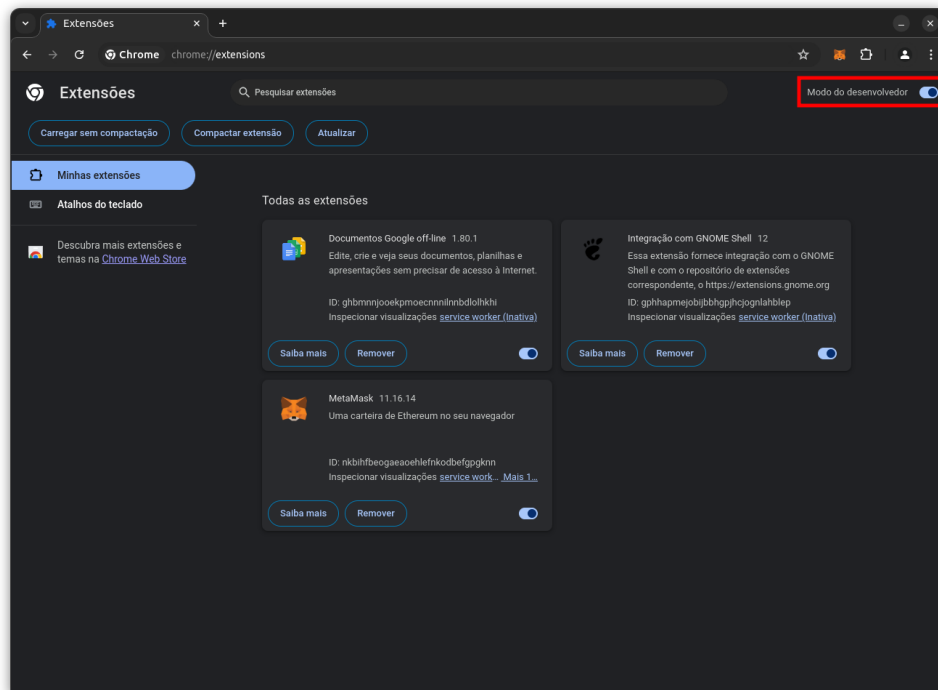


Figura 6.9: 3º passo de instalação.

4. Arrastar o arquivo **GeoBlocos.crx** para a janela do Google Chrome.

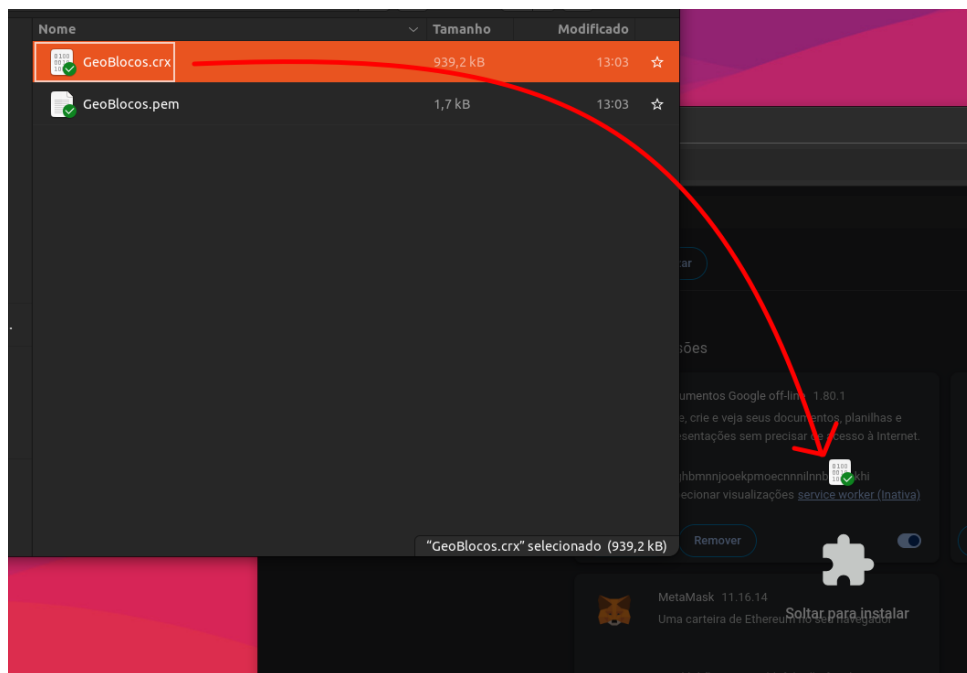


Figura 6.10: 4º passo de instalação.

5. Confirmar a instalação da extensão.

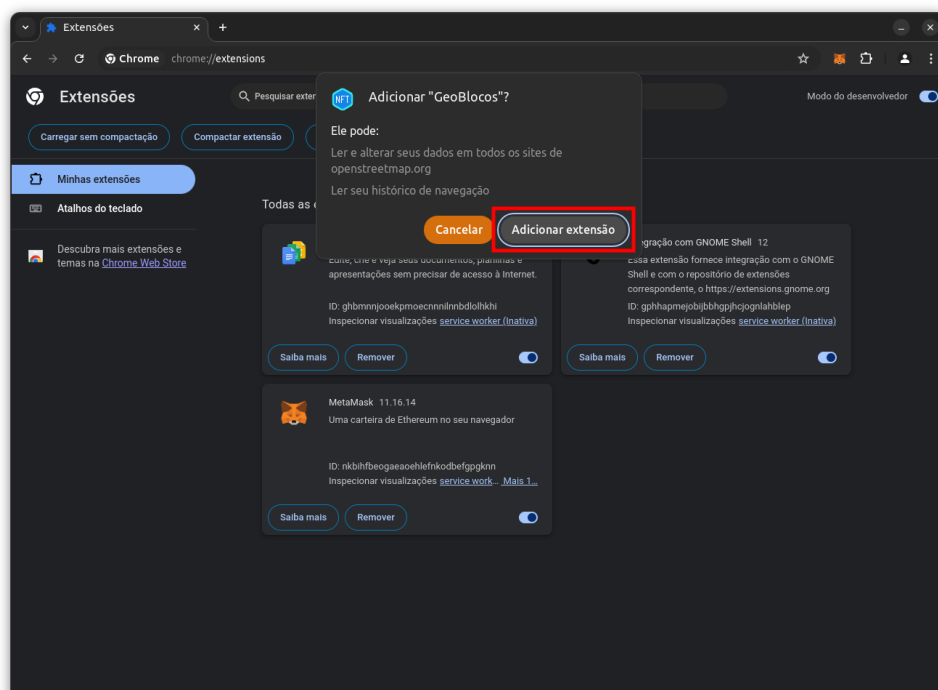


Figura 6.11: 5º passo de instalação.

6.3 Guia de uso

1. Instalar e configurar no Google Chrome a extensão da carteira digital MetaMask, disponível no endereço: <https://metamask.io/download/>.

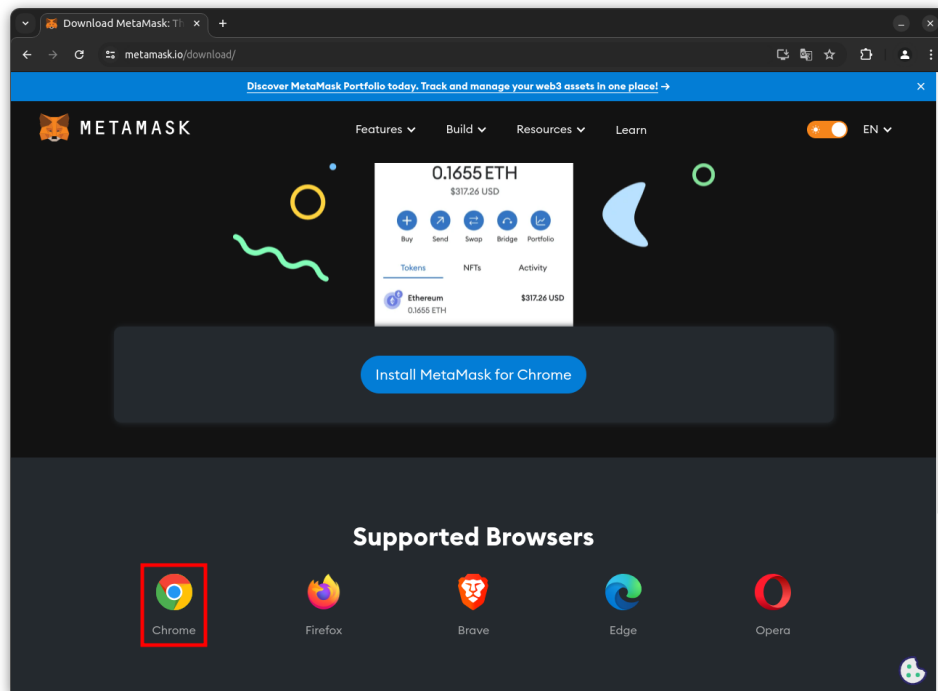


Figura 6.12: Instalação do MetaMask.

2. Conectar com a conta de usuário no OpenStreetMap pelo endereço: <https://www.openstreetmap.org/login>.

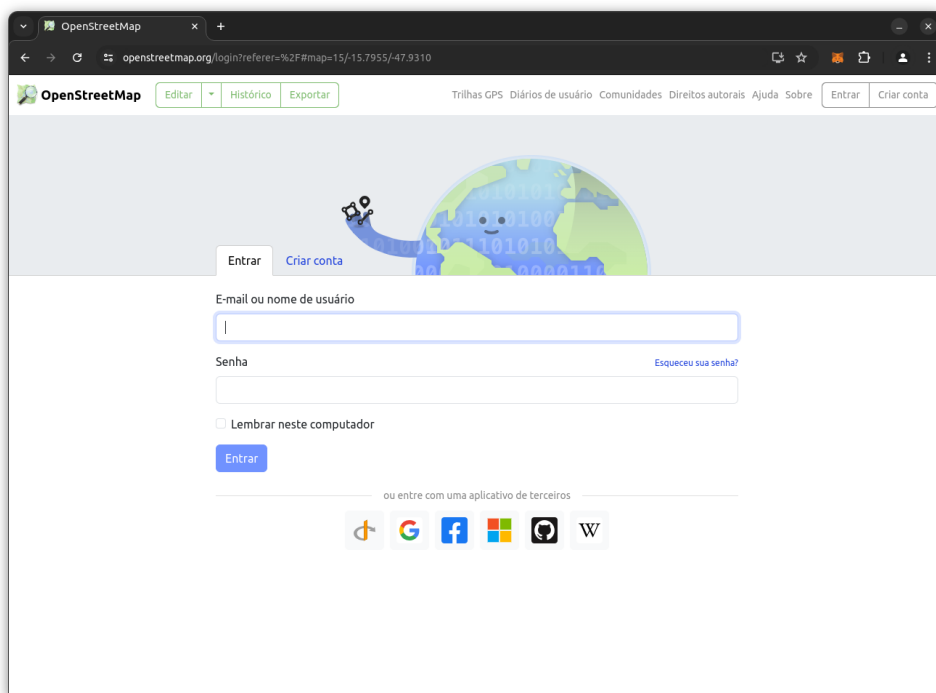


Figura 6.13: Tela de autenticação do OpenStreetMap.

3. Clicar no ícone da extensão GeoBlocs para abrir o painel da aplicação.

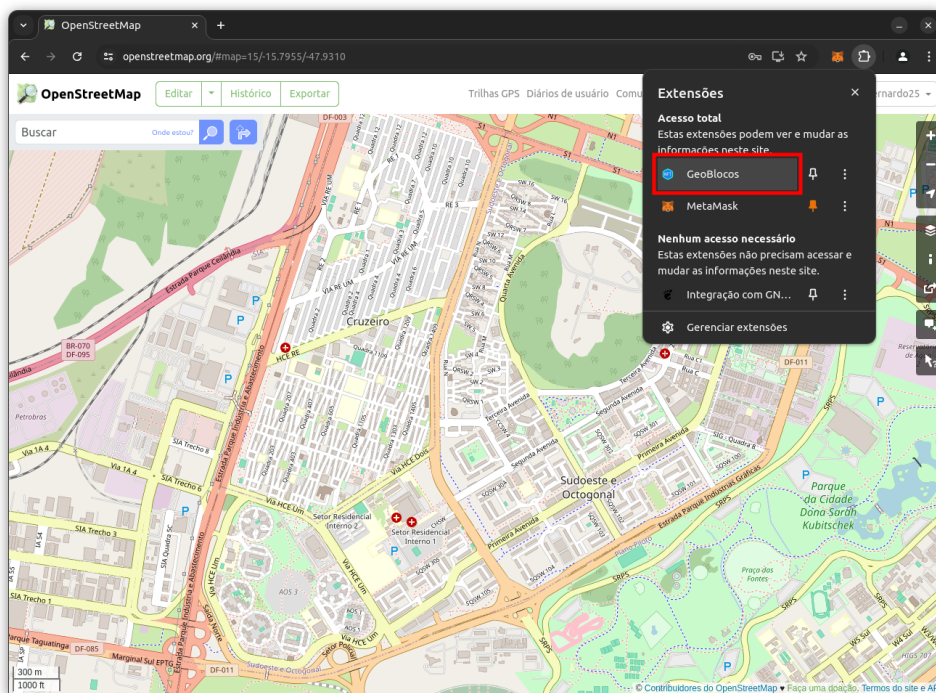


Figura 6.14: Acesso à aplicação GeoBlocs.

4. A navegação pelo sistema é feita pela caixa de seleção destacada na Figura 6.15.

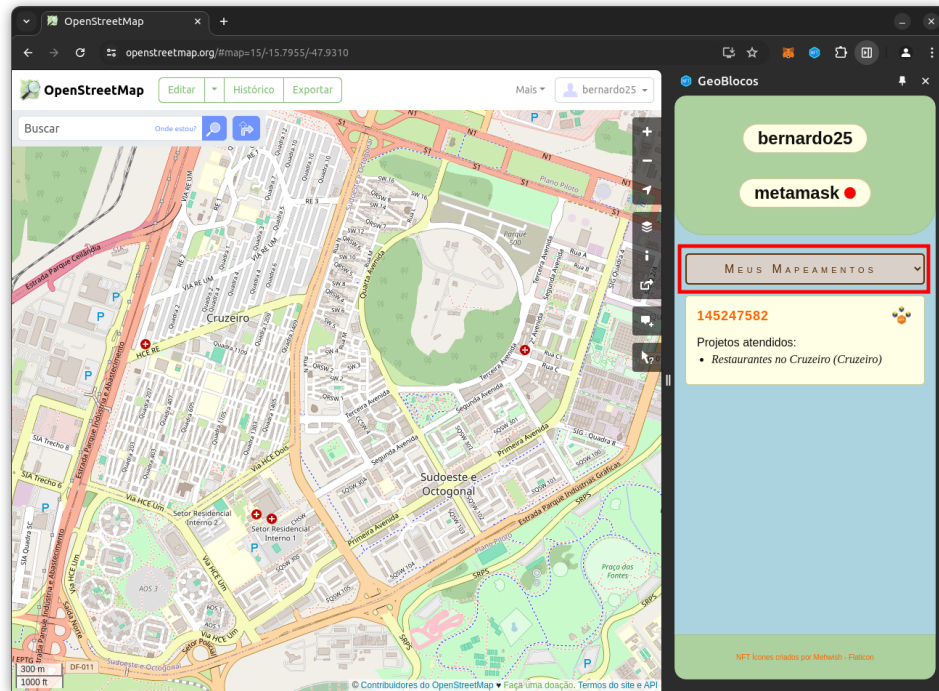


Figura 6.15: Navegação entre as telas do sistema.

5. Para criar um novo NFT deve-se clicar no ícone correspondente ao mapeamento desejado na tela **Meus Mapeamentos**.

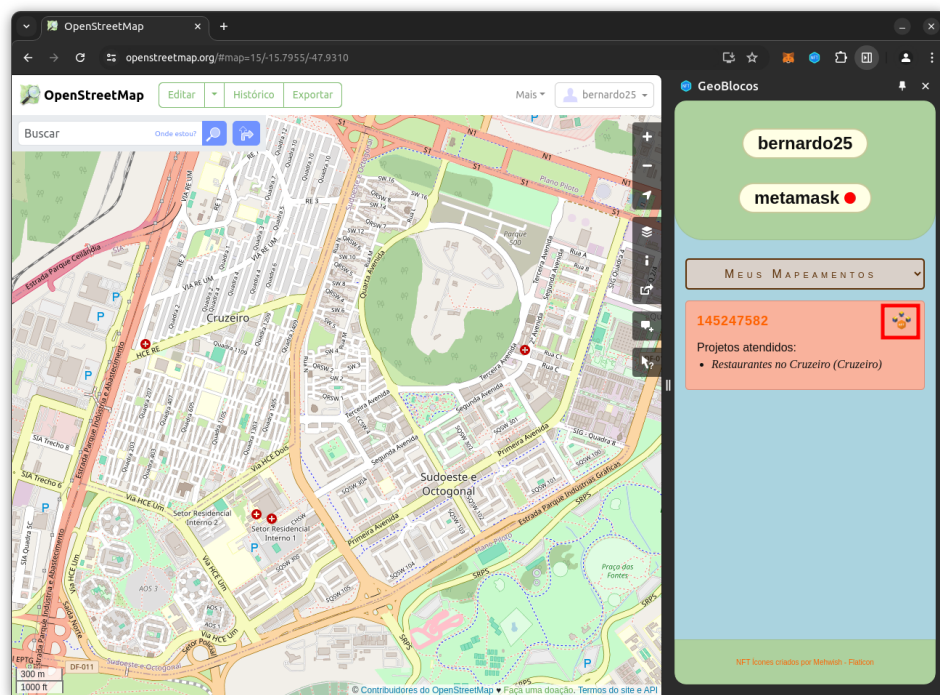


Figura 6.16: Criação de um novo NFT.

6. Para criar um novo projeto de mapeamento deve-se clicar no botão destacado abaixo da tela **Meus Projetos**, preencher os dados solicitados e clicar em **Criar Projeto**.

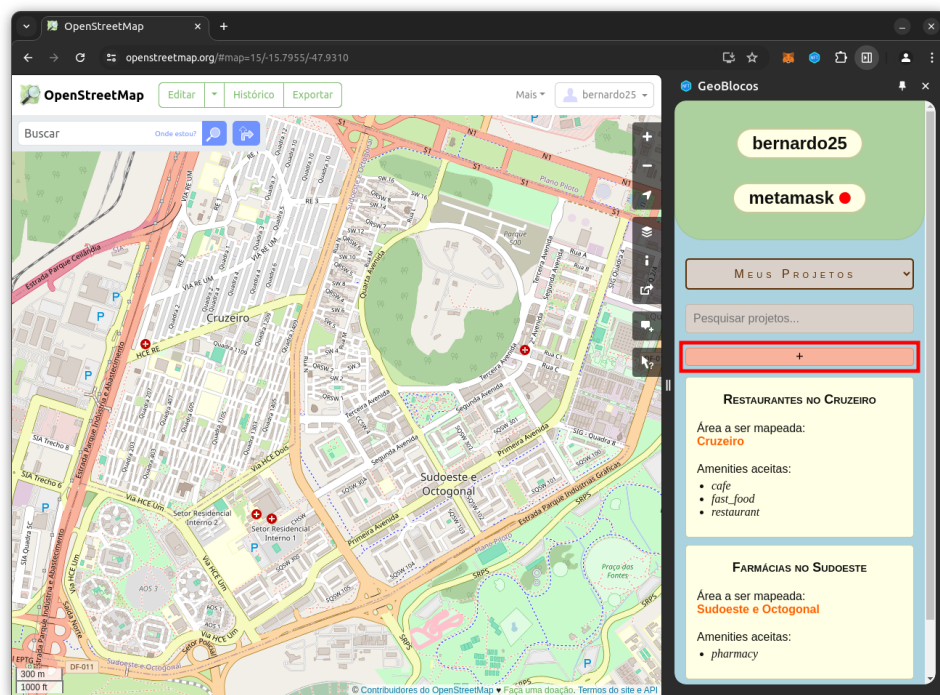


Figura 6.17: Criação de novo projeto.

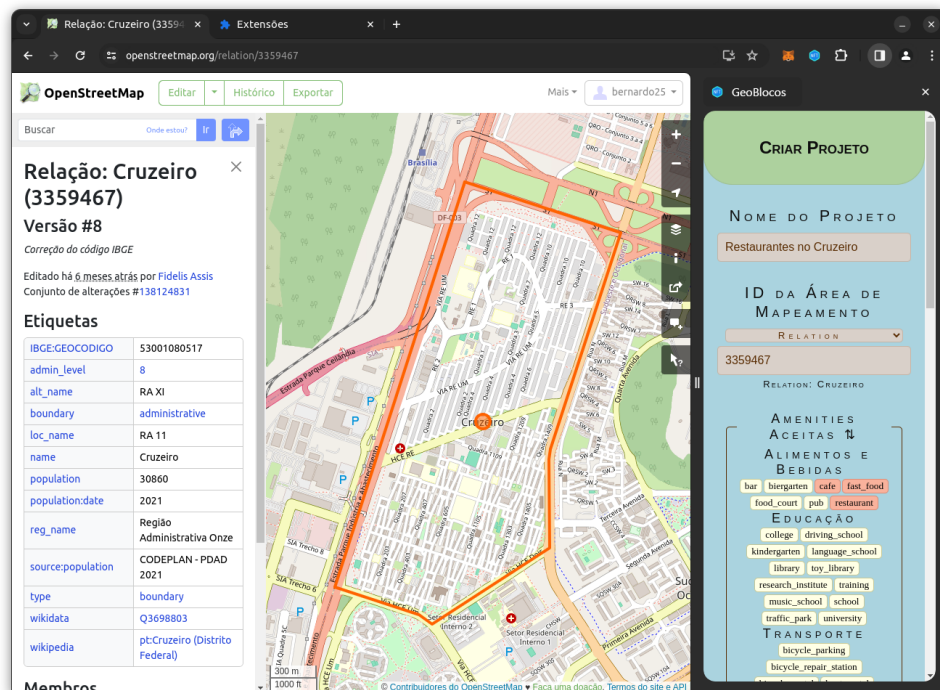


Figura 6.18: Preenchimento das informações de um novo projeto.

Capítulo 7

Discussões

7.1 Avaliação dos resultados

Este trabalho explorou uma das possibilidades da construção de aplicações baseadas na tecnologia *blockchain*, a utilizando para a persistência de dados de um *software* que objetiva incentivar a contribuição de Informações Geográficas Voluntárias para a base de dados aberta do OpenStreetMap. O incentivo proposto foi a recompensa por meio de tokens digitais do tipo Non-Fungible Tokens (NFT) para informações fornecidas pelos usuários que atendam a projetos de mapeamento criados e delimitados também na aplicação desenvolvida.

O modelo de recompensas que atua como impulsionador das contribuições de dados geográficos foi criado com base no hábito já estabelecido pela Web 2.0 de construção do conteúdo disponível *online* por informações enviadas pelos próprios usuários aliada a alguns dos recursos disponíveis como parte da Web3, a descentralização oferecida pelas redes *blockchain* e os mecanismos de posse de ativos digitais dessas redes. Cada contribuição enviada para o OpenStreetMap gera para o usuário criador um ativo único, não duplicável, colecionável e que pode ser transacionado dentro da rede Ethereum como um NFT. Esses tokens são associados a projetos de mapeamento listados no aplicativo Geo-Blocos e são uma representação da recompensa pelo levantamento de informação sobre o espaço geográfico relacionado.

A aplicação desenvolvida pode ser considerada como descentralizada pois não depende de um único ponto de controle ou de uma entidade central para seu funcionamento. Todo o processamento de dados necessários para execução da aplicação acaba sendo computado ou no computador do usuário, acessada por meio de um navegador de internet, ou pela computação distribuída na rede Ethereum utilizando *smart contracts* executados pela Ethereum Virtual Machine (EVM).

Entretanto, processamentos realizados utilizando os recursos computacionais da rede

blockchain Ethereum geram um custo a ser pago com a criptomoeda Ether, sendo esse custo gerado toda vez em que há inclusão, alteração ou exclusão de dados armazenados pelos *smart contracts*. Essa característica aliada à elevação da cotação de preços do Ether acabaram limitando os testes pretendidos da aplicação desenvolvida neste trabalho com grupos de usuários, pois mesmo para o uso da rede de testes do Ethereum, a rede Sepolia, é exigido um saldo mínimo ativo na rede principal para cada carteira de usuário que faça uso dos *smart contracts* do sistema, resultando assim em um custo financeiro para a realização desses testes com usuários. Desta forma, destacando uma das desvantagens encontradas em aplicações descentralizadas baseadas em *blockchains*, o custo de disponibilidade do sistema fica distribuído a todos os seus usuários, algo que pode ser um fator desencorajador do uso de aplicações do tipo.

Considerando parte da proposta inicial deste trabalho, que foi também o desenvolvimento de uma aplicação que sirva de ferramenta educacional, podemos observar que se encontra consonância com propostas com a do modelo do Bairro-Escola. Com o uso da aplicação em sala de aula e seu modelo de incentivo a contribuição de Informações Geográficas Voluntárias, abre-se a oportunidade de promover a integração da escola com o espaço geográfico que a envolve, destacando especialmente o conhecimento sobre esse espaço e utilizando o conhecimento sobre ele como ferramenta de ensino e aprendizado aos envolvidos: alunos, professores e comunidade.

7.2 Trabalhos futuros

Como sugestão de trabalhos a serem realizados em continuidade a este, destaca-se a possibilidade de uso da aplicação com grupos de usuários para melhor avaliar a adequação do *software* desenvolvido como uma solução para a pergunta inicial desta pesquisa, de como a produção de dados geográficos voluntários pode ser incentivada com a possibilidade de construção de aplicações descentralizadas na Web3, explorando a arquitetura de soluções computacionais baseadas em redes *blockchain*. Especialmente para testes com usuários que possam ser feitos em ambiente escolar e como atividade didática promotora da integração escola-comunidade. Além desta aplicação, encoraja-se também a exploração da aplicação como ferramenta didática para o desenvolvimento do letramento geográfico dos estudantes.

A realização de testes pode também ser usada para a identificação de novas funcionalidades que sejam interessantes para a finalidade da aplicação, expandindo o seu rol de usos ou aprimorando as funcionalidades já existentes. Outro ponto de atenção no *feedback* de usuários a ser observado é a adequada facilidade de uso do sistema, observando quais pontos estão passíveis de melhorias ou até mesmo alterações visuais que tornem o seu uso

mais agradável e atrativo para os usuários. Tais observações abrem ainda a possibilidade de elaboração de um material didático de guia e incentivo de uso que seja adequado a esses fins, especialmente considerando um uso em ambiente educacional.

Considerando as dificuldades encontradas para realização de testes da aplicação com usuários, pode-se também considerar a reelaboração da aplicação utilizando redes *blockchain* diferentes da Ethereum. A elevação dos custos de uso da rede pode ser um fator bastante desencorajador do seu uso e invalidar o seu propósito de existência como incentivador do mapeamento de regiões que não possuam recursos públicos o suficiente e onde também não existe interesse econômico do ponto de vista de iniciativas do setor privado. Uma das possibilidades seria o uso de *blockchains* com custos de processamento menores.

Referências

- [1] Santos, M., B.K. Becker e C.A.F. da Silva: *Território, territórios: ensaios sobre o ordenamento territorial*. Coleção espaço, território e paisagem. Lamparina, ISBN 978-85-98271-42-2. 1
- [2] Costa, Natacha: *Apresentação - bairro-escola: comunidades educativas por uma educação integral*. Em Singer, Helena (editor): *Articulação Escola-Comunidade*, volume 5 de *Tecnologias do Bairro-escola*, páginas 9–15. Moderna, ISBN 978-85-64569-04-1. https://www.cidadeescolaaprendiz.org.br/wp-content/uploads/2014/08/Tecnologias-do-Bairro-escola_Vol5_articulacao-escola-comunidade.pdf. 1
- [3] Bolfe, Édson, Marcos Ferreira e Lindon Matias: *SISTEMAS DE INFORMAÇÃO GEOGRÁFICA: UMA ABORDAGEM CONTEXTUALIZADA NA HISTÓRIA*. 33(1):69–88. 2, 8
- [4] *Wikimapia / about*. <https://wikimapia.org/about/>, acesso em 2024-07-16. 2, 8
- [5] *About - wikiloc*. <https://pt.wikiloc.com/wikiloc/about-us.do>, acesso em 2024-07-16. 2, 8
- [6] *About OpenStreetMap - OpenStreetMap wiki*. https://wiki.openstreetmap.org/wiki/About_OpenStreetMap, acesso em 2023-12-14. 2, 8, 21
- [7] Ferster, Colin J., Trisalyn Nelson, Colin Robertson e Rob Feick: *1.04 - current themes in volunteered geographic information*. Em Huang, Bo (editor): *Comprehensive Geographic Information Systems*, páginas 26–41. Elsevier, ISBN 978-0-12-804793-4. <https://www.sciencedirect.com/science/article/pii/B9780124095489096202>. 2, 8
- [8] Nakamoto, Satoshi: *Bitcoin: A peer-to-peer electronic cash system*. <https://bitcoin.org/bitcoin.pdf>. 2, 13
- [9] Atlas, L. Godlin, C. Magesh Kumar, Rajakumari e P. Hamsagayathi: *APPLICATIONS WITH BLOCKCHAIN TECHNIQUE*. Em *Cryptocurrencies and Blockchain Technology Applications*, páginas 157–179. John Wiley & Sons, Ltd, ISBN 978-1-119-62120-1. <https://onlinelibrary.wiley.com/doi/10.1002/9781119621201.ch9>. 2, 13
- [10] Aggarwal, Shubhani e Neeraj Kumar: *Blockchain 2.0: Smart contracts*. Em Aggarwal, Shubhani, Neeraj Kumar e Pethuru Raj (editores): *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*, volume 121 de *Advances in*

- Computers*, páginas 301–322. Elsevier. <https://www.sciencedirect.com/science/article/pii/S006524582030070X>, ISSN: 0065-2458. 2, 18
- [11] Singh, Harjot: *Chapter 6 - DApps: Decentralized applications for blockchains*. Em Pandey, Rajiv, Sam Goundar e Shahnaz Fatima (editores): *Distributed Computing to Blockchain*, páginas 87–104. Academic Press, ISBN 978-0-323-96146-2. <https://www.sciencedirect.com/science/article/pii/B9780323961462000218>. 3, 19, 20
 - [12] Fernandes, Jorge Henrique Cabral: *Leitura das territorialidades no OpenStreetMap: Uma oficina de ensino-aprendizagem para promoção da TEIA - Territórios Escolares Inteligentes e Abertos*. <https://sigaa.unb.br/sigaa/link/public/extensao/visualizacaoAcaoExtensao/4292>. 3, 11
 - [13] Wan, Shicheng, Hong Lin, Wensheng Gan, Jiahui Chen e Philip S. Yu: *Web3: The next internet revolution*. <https://arxiv.org/abs/2304.06111>, [_eprint: 2304.06111](https://arxiv.org/abs/2304.06111). 3
 - [14] Azevedo, Rogério Cabral e Leonardo Ensslin: *Metologia de Pesquisa para Engenharias*. PPGE/CEFET-MG, 1ª edição, ISBN 978-65-00-10268-0. 5
 - [15] Sommerville, Ian: *Processos de software*. Em *Engenharia de Software*, páginas 18–37. Pearson Prentice Hall, 9ª edição, ISBN 978-85-7936-108-1. 5
 - [16] Câmara, Gilberto, Marco Casanova, Andrea Hemerly, Geovane Magalhães e Claudia Medeiros: *Sistemas de informação geográfica*. Em *Anatomia de Sistemas de Informação Geográfica*, páginas 21–35. https://www.researchgate.net/publication/288981872_Anatomia_de_Sistemas_de_Informacao_Geografica. 8
 - [17] Goodchild, Michael F.: *Citizens as sensors: the world of volunteered geography*. 69(4):211–221, ISSN 1572-9893. <https://doi.org/10.1007/s10708-007-9111-y>. 8
 - [18] Poorthuis, Ate: *Geotag*. Em *International Encyclopedia of Human Geography (Second Edition)*, páginas 137–140. Elsevier, 2ª edição, ISBN 978-0-08-102296-2. <https://www.sciencedirect.com/science/article/pii/B9780081022955105517>. 10
 - [19] Walter, Christin: *Future trends in geospatial information management: the five to ten year vision – third edition*. <https://ggim.un.org/UN-GGIM-publications/#FutureTrends>. 10
 - [20] Johnson, Peter A. e Renee E. Sieber: *Situating the adoption of VGI by government*. Em *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*, páginas 65–81. Springer Netherlands, ISBN 978-94-007-4587-2. https://doi.org/10.1007/978-94-007-4587-2_5. 10
 - [21] Ganapati, Sukumar: *Using geographic information systems to increase citizen engagement*. <https://www.businessofgovernment.org/sites/default/files/GanapatiReport.pdf>. 10

- [22] Goodchild, Michael F. e Linna Li: *Assuring the quality of volunteered geographic information*. 1:110–120. <https://www.sciencedirect.com/science/article/pii/S2211675312000097>. 10
- [23] Costa, Natacha: *Educação, cidade e democracia: a agenda do bairro-escola*. Em Singer, Helena (editor): *Territórios educativos: experiências em diálogo com o Bairro-Escola*, volume 1 de *Coleção territórios educativos*, páginas 11–22. Moderna, 1ª edição, ISBN 978-85-16-09696-0. https://www.cidadeescolaaprendiz.org.br/wp-content/uploads/2015/03/Territorios-Educativos_Vol1.pdf. 11
- [24] Dowbor, Ladislau: *Educação e desenvolvimento local*. 261:15–30. http://lam.ibam.org.br/revista_detalhe.asp?idr=649. 12
- [25] Medeiros Filho, Barnabé e Mônica Beatriz Galiano: *Bairro-escola : uma nova geografia do aprendizado : a tecnologia da Cidade Escola Aprendiz para integrar escola e comunidade*. Tempo D’Imagem, ISBN 85-87314-07-6. 12
- [26] Singer, Helena: *Apresentação*. Em Singer, Helena (editor): *Territórios educativos : experiências em diálogo com o Bairro-Escola*, volume 1 de *Coleção territórios educativos*, páginas 5–10. Moderna. https://www.cidadeescolaaprendiz.org.br/wp-content/uploads/2015/03/Territorios-Educativos_Vol1.pdf. 12
- [27] Xavier, Iara Rolnik: *Um olhar sobre o território na estratégia do bairro-escola*. Em Singer, Helena (editor): *Territórios educativos : experiências em diálogo com o Bairro-Escola*, volume 2 de *Coleção territórios educativos*, páginas 25–43. Moderna. https://www.cidadeescolaaprendiz.org.br/wp-content/uploads/2015/03/Territorios-Educativos_Vol2.pdf. 12
- [28] Cedraz, Aroldo: *Levantamento da tecnologia blockchain*. <https://portal.tcu.gov.br/levantamento-da-tecnologia-blockchain.htm>. 14, 15, 16
- [29] Saghiri, Ali Mohammad: *Blockchain architecture*. Em *Advanced Applications of Blockchain Technology*, páginas 161–176. Springer Singapore, ISBN 978-981-13-8775-3. https://doi.org/10.1007/978-981-13-8775-3_8. 15
- [30] Kaur, Avinash, Anand Nayyar e Parminder Singh: *BLOCKCHAIN: A PATH TO THE FUTURE*. Em *Cryptocurrencies and Blockchain Technology Applications*, páginas 25–42. John Wiley & Sons, Ltd, ISBN 978-1-119-62120-1. <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119621201.ch2>. 16
- [31] Bowman, Mic, Debajyoti Das, Avradip Mandal e Hart Montgomery: *On elapsed time consensus protocols*. <https://eprint.iacr.org/2021/086>, Published: Cryptology ePrint Archive, Paper 2021/086. 17
- [32] Bentov, Iddo, Charles Lee, Alex Mizrahi e Meni Rosenfeld: *Proof of activity: Extending bitcoin’s proof of work via proof of stake*. <https://eprint.iacr.org/2014/452>, Published: Cryptology ePrint Archive, Paper 2014/452. 17

- [33] *Smart contracts in blockchain.* <https://www.geeksforgeeks.org/smart-contracts-in-blockchain/>, acesso em 2023-12-14, Section: Blockchain. 18
- [34] Buterin, Vitalik: *A next-generation smart contract and decentralized application platform.* https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf. 18, 26
- [35] *Elements - OpenStreetMap wiki.* <https://wiki.openstreetmap.org/wiki/Elements>, acesso em 2023-12-14. 22
- [36] *Pt:map features - OpenStreetMap wiki.* https://wiki.openstreetmap.org/wiki/Pt:Map_Features, acesso em 2023-12-14. 23
- [37] *Databases and data access APIs - OpenStreetMap wiki.* https://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs, acesso em 2023-12-14. 24, 31
- [38] *API v0.6 - OpenStreetMap wiki.* https://wiki.openstreetmap.org/wiki/API_v0.6, acesso em 2024-07-26. 24, 46
- [39] *Overpass API - OpenStreetMap wiki.* https://wiki.openstreetmap.org/wiki/Overpass_API, acesso em 2024-07-26. 24, 46
- [40] *Extensões / desenvolver / chrome for developers.* <https://developer.chrome.com/docs/extensions/develop?hl=pt-br>, acesso em 2023-12-14. 25
- [41] *Intro to ethereum.* <https://ethereum.org/en/developers/docs/intro-to-ethereum/>, acesso em 2023-12-14. 25
- [42] *Introduction to dapps.* <https://ethereum.org/en/developers/docs/dapps/>, acesso em 2023-12-14. 26
- [43] *Getting started with MetaMask.* <https://support.metamask.io/hc/en-us/articles/360015489531-Getting-started-with-MetaMask>, acesso em 2023-12-14. 26
- [44] *About the MetaMask APIs / MetaMask developer documentation.* <https://docs.metamask.io/wallet/concepts/apis/>, acesso em 2023-12-14. 27
- [45] *IPFS documentation / IPFS docs.* <https://docs.ipfs.tech/>, acesso em 2023-12-14. 27
- [46] Benet, Juan: *IPFS - content addressed, versioned, p2p file system.* <https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf>. 27
- [47] *ERC-721: Non-fungible token standard.* <https://eips.ethereum.org/EIPS/eip-721>, acesso em 2023-12-14. 27, 31, 36, 39
- [48] Sommerville, Ian: *Modelagem de sistemas.* Em *Engenharia de Software*, páginas 82–102. Pearson Prentice Hall, 9ª edição, ISBN 978-85-7936-108-1. 29

- [49] Hofmeister, Christine, Robert Nord e Dilip Soni: *Applied Software Architecture*. Addison-Wesley object technology series. Addison-Wesley, ISBN 0-201-32571-3. 30
- [50] *Quickstart / NFT.storage*. <https://app.nft.storage/v1/docs/intro>, acesso em 2024-07-25. 31, 43
- [51] Sommerville, Ian: *Projeto de arquitetura*. Em *Engenharia de Software*, páginas 103–123. Pearson Prentice Hall, 9ª edição, ISBN 978-85-7936-108-1. 33
- [52] *ERC721 - OpenZeppelin docs*. <https://docs.openzeppelin.com/contracts/4.x/erc721>, acesso em 2024-07-25. 36, 39
- [53] Sommerville, Ian: *Testes de software*. Em *Engenharia de Software*, páginas 144–163. Pearson Prentice Hall, 9ª edição, ISBN 978-85-7936-108-1. 47
- [54] *Welcome to remix's documentation! — remix - ethereum IDE 1 documentation*. <https://remix-ide.readthedocs.io/en/latest/>, acesso em 2024-07-26. 47, 48, 49
- [55] *Redes / ethereum.org*. <https://ethereum.org/pt-br/developers/docs/networks/>, acesso em 2024-07-26. 48
- [56] etherscan.io: *TESTNET sepolia (ETH) blockchain explorer*. <https://sepolia.etherscan.io/>, acesso em 2024-07-26. 48, 49