



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CARTOGRÁFICA
CURSO DE GRADUAÇÃO EM ENGENHARIA CARTOGRÁFICA E DE
AGRIMENSURA

THALYTA KATTIANE BARBOSA LOBO

**Desenvolvimento de uma ferramenta web integrando tecnologias
geoespaciais para auxiliar na gestão e controle dos Tombamentos do
Departamento de Engenharia Cartográfica (DECART) da Universidade
Federal de Pernambuco (UFPE)**

RECIFE
2025

THALYTA KATTIANE BARBOSA LOBO

**Desenvolvimento de uma ferramenta web integrando tecnologias
geoespaciais para auxiliar na gestão e controle dos Tombamentos do
Departamento de Engenharia Cartográfica (DECART) da Universidade
Federal de Pernambuco (UFPE)**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia
Cartográfica e de Agrimensura da
Universidade Federal de Pernambuco,
como requisito parcial para a
obtenção do título de Engenheira
Cartógrafa e Agrimensora.

Orientador: Prof. Dr. Cézar de Oliveira Lima Júnior

RECIFE

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Lobo, Thalyta Kattiane Barbosa.

Desenvolvimento de uma ferramenta web integrando tecnologias geoespaciais para auxiliar na gestão e controle dos Tombamentos do Departamento de Engenharia Cartográfica (DECART) da Universidade Federal de Pernambuco (UFPE) / Thalyta Kattiane Barbosa Lobo. - Recife, 2025.

70 p.

Orientador(a): Cézario de Oliveira Lima Júnior

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Cartográfica e de Agrimensura - Bacharelado, 2025.

Inclui apêndices.

1. Engenharia Cartográfica. 2. Bando de Dados. 3. Sistema de Informações Geográficas . 4. Programação de Página WEB. I. Lima Júnior, Cézario de Oliveira. (Orientação). II. Título.

620 CDD (22.ed.)

THALYTA KATTIANE BARBOSA LOBO

**Desenvolvimento de uma ferramenta web integrando tecnologias
geoespaciais para auxiliar na gestão e controle dos Tombamentos do
Departamento de Engenharia Cartográfica (DECART) da Universidade
Federal de Pernambuco (UFPE)**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia
Cartográfica e de Agrimensura da
Universidade Federal de Pernambuco,
como requisito parcial para a
obtenção do título de Engenheira
Cartógrafa e Agrimensora.

Aprovado em: 02/09/2025.

BANCA EXAMINADORA

Prof^o. Dr. Cézario de Oliveira Lima Júnior (Orientador)
Universidade Federal de Pernambuco

Prof^a. Dr^a. Karoline Paes Jamur (Examinador Interno)
Universidade Federal de Pernambuco

Prof^a. Dr^a. Manuella Anaís Rodrigues Fagundes (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Agradeço primeiramente a Deus que tem me ajudado e sustentado até aqui, o caminho foi árduo, mas com sua infinita graça e misericórdia me fez chegar até aqui.

Agradeço a minha mãe Tânia Maria Barbosa. Obrigada por todo amor, esforço e dedicação para comigo ao longo desse caminho. A senhora tem sido minha força para continuar, esteve comigo em cada dificuldade e não soltou minha mão em nenhuma delas, muitas vezes abdicando de coisas para si própria para que eu pudesse concluir meus estudos, obrigada por ser minha apoiadora incondicional e por estar ao meu lado sempre, eu te amo.

Agradeço aos “meus Moisés” (pai e irmão). Obrigada por me ajudarem nessa jornada de toda forma que podiam e por apoiar os meus sonhos, a caminhada foi difícil, mas chegamos ao final, junto com minha vocês são meu alicerce eu amo vocês.

Agradeço também a Guilherme Pedro Lima da Costa meu namorado e acima de tudo, melhor amigo. Obrigada por cuidar de mim, me incentivar, acreditar em mim, estar comigo em todas as horas e principalmente, obrigada por não ter me deixado desistir nas muitas vezes em que eu quis o fazer, eu te amo infinitos infilhões.

Agradeço aos meus familiares e amigos que fiz ao longo desse caminho, vocês foram essenciais na minha evolução e crescimento. Obrigada pelas palavras de apoio e por acreditarem em mim.

Agradeço ao Profº. Dr. Cézario de Oliveira Lima Júnior pela orientação nesse trabalho.

E ao meu redor só pude ver, eles
dizendo que, eu iria desistir, mas
aqui eu sigo, vivendo o ciclo.
(Juyè, Sete Mares, 2020)

RESUMO

O presente trabalho visa o desenvolvimento de uma ferramenta web para auxiliar no controle dos tombamentos do Departamento de Engenharia Cartográfica (DECART) do Campus Recife da Universidade Federal de Pernambuco (UFPE). A criação da ferramenta visa aprimorar a administração e organização das informações do departamento através de uma interface amigável, tanto para o público interno quanto para o externo. Para isso, será utilizada uma abordagem que combina as potencialidades do software livre QGIS e o banco de dados geográficos PostgreSQL com PostGIS, promovendo uma gestão mais eficiente dos recursos e patrimônios da universidade. A ferramenta está disponível online e estruturada em duas partes: uma área pública, acessível a todos, que apresenta informações sobre a localização de salas de aula, salas de docentes, laboratórios e demais pontos de interesse; e uma área restrita, acessível apenas a usuários autenticados, onde será possível realizar a administração interna, incluindo a consulta de patrimônios, responsáveis por espaços e a gestão de problemas internos.

Palavras-chave: ferramenta web; QGIS; banco de dados geográficos.

Sumário

1 INTRODUÇÃO	9
2 OBJETIVOS	11
2.1 Objetivo Geral	11
2.2 Objetivos Específicos	11
3 REVISÃO DE LITERATURA	12
3.1 Universidade Federal de Pernambuco	12
3.2 Centro de Tecnologia e Geociências	12
3.3 Departamento de Engenharia Cartográfica	13
3.4 Banco de Dados	13
3.4.1 Banco De Dados Geográficos	14
3.5 Sistema de Informações Geográficas	15
3.6 QGIS	15
3.7 PostgreSQL	15
3.7.1 PostGIS	16
3.8 Tecnologias de Posicionamento: GNSS e Correção RTK	16
4 METODOLOGIA	18
4.1 Levantamento de dados	19
4.1.1 Planta baixa do DECart	19
4.1.2 Levantamento de dados em campo	20
4.2 Informações sobre as Salas	22
4.3 Desenvolvimentos da Ferramenta Web	23
4.3.1 Georreferenciamento da Planta Baixa	23
4.3.2 Banco de Dados	25
4.3.3 Conexão do Banco de Dados ao QGIS	29
4.3.4 Programação da ferramenta web	33
4.3.5 Interface Web	35
5 RESULTADOS	45
5.1 Georreferenciamento	45
5.2 Visualização da Camada Espacial no PostgreSQL	47
5.3 Desenvolvimento da Ferramenta Web	47
5.3.1 Estrutura e Funcionamento	47
5.3.2 Página Inicial da Ferramenta Web	48
5.3.3 Acesso Público	49

5.3.4 Acesso Restrito	51
5.3.5 Integração e Potencial de Expansão	53
6 CONCLUSÃO.....	54
REFERÊNCIAS.....	56
APÊNDICE A – CÓDIGO PYTHON CRIADO NO AMBIENTE GOOGLE COLAB (MAPA DE ACESSO PÚBLICO)	59
APÊNDICE B – CÓDIGO PYTHON CRIADO NO AMBIENTE GOOGLE COLAB (MAPA DE ACESSO RESTRITO).....	65

1 INTRODUÇÃO

A crescente demanda por uma gestão mais eficiente e acessível de informações acadêmicas e administrativas tem impulsionado a necessidade de soluções tecnológicas inovadoras nas universidades. Neste contexto, a criação de uma ferramenta na web surge como uma proposta crucial para aprimorar a administração e organização das informações do campus, integrando tecnologias geoespaciais para otimizar o acesso e a gestão dos recursos institucionais.

O presente trabalho visa desenvolver a ferramenta web que se destaca pela sua dualidade de acesso: uma área interna, restrita a usuários autenticados, voltada para a administração e organização das informações acadêmicas e administrativas, e uma área externa, acessível ao público geral, que visa fornecer informações úteis sobre a infraestrutura do Centro de Tecnologia e Geociências (CTG). Para tal usaremos como exemplo prático o Departamento de Engenharia Cartográfica, que fica localizado no segundo andar do prédio do CTG.

Na parte interna da web, a solução proposta permitirá a gestão mais eficiente dos patrimônios da universidade, incluindo salas de aula, laboratórios e outros espaços institucionais, assim como dos equipamentos. Esta seção da web oferecerá ferramentas para o rastreamento e gestão de ativos, identificação de responsáveis por cada sala e laboratório, e a possibilidade de solucionar problemas administrativos, como alocação inadequada de espaços e manutenção predial. Além disso, permitirá a gestão de informações como a disponibilidade de salas para reserva e a identificação de recursos e equipamentos disponíveis em diferentes áreas do campus.

A parte externa da web será desenvolvida com o objetivo de fornecer informações acessíveis ao público, facilitando a navegação e a localização de salas e outros pontos de interesse no campus. Esta interface permitirá aos visitantes e à comunidade acadêmica a consulta de mapas interativos e informações detalhadas sobre a disposição dos edifícios, serviços disponíveis e áreas de interesse.

Para a implementação da página na web, foi utilizado o software livre QGIS, conhecido por sua robustez e flexibilidade na manipulação de dados

geoespaciais, juntamente com o banco de dados geográfico PostgreSQL com PostGIS, que oferece suporte avançado para o gerenciamento e consulta de grandes volumes de dados espaciais. Foi feita a publicação do site, garantindo acesso contínuo e fácil aos usuários, além de assegurar a integração com outras ferramentas e sistemas existentes na universidade.

A adoção desta solução visa não apenas otimizar a administração dos recursos da UFPE, mas também proporcionar um canal de comunicação mais eficiente e transparente com a comunidade acadêmica e o público externo. Assim, este projeto representa um avanço significativo na utilização de tecnologias geoespaciais para a gestão acadêmica e administrativa, alinhando-se às melhores práticas de inovação e eficiência na administração universitária.

Ao final deste trabalho, espera-se que este estudo forneça uma visão abrangente e aprofundada do site desenvolvido para o Departamento de Engenharia Cartográfica da UFPE, demonstrando como a implementação de uma ferramenta web pode impulsionar a pesquisa, o ensino e a gestão do espaço em uma instituição de ensino superior.

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver e implementar uma ferramenta web no Departamento de Engenharia Cartográfica, utilizando ferramentas e softwares gratuitos, que integre os conceitos de Sistema de Informação Geográfica (SIG) e Banco de Dados Geográfico, visando aprimorar a gestão do espaço universitário e promover o uso eficiente da geoinformação para pesquisa e tomada de decisões.

2.2 Objetivos Específicos

- Definir as tecnologias, ferramentas e sistemas de softwares gratuitos mais adequados para a criação da ferramenta web, levando em consideração a integração de dados geoespaciais provenientes de diversas fontes;
- Desenvolver uma estrutura de Banco de Dados que permita o armazenamento eficiente e a recuperação de informações, garantindo a integridade dos dados e a segurança da informação;
- Criar interface de login que sejam intuitivas, para facilitar o acesso e a consulta de informações geoespaciais por parte da comunidade acadêmica e administrativa da UFPE;

3 REVISÃO DE LITERATURA

3.1 Universidade Federal de Pernambuco

De acordo com informações do site oficial da UFPE, a instituição foi fundada em 11 de agosto de 1946, inicialmente com a denominação de Universidade do Recife (UR), por meio do Decreto-Lei nº 9.388 da Presidência da República. A criação da UR teve como objetivo reunir diversas escolas de ensino superior já existentes em Pernambuco, entre elas a tradicional Faculdade de Direito do Recife, fundada em 1827, e a Escola de Engenharia de Pernambuco, criada em 1895. Dois anos depois, em 1948, foi iniciada a construção do campus Recife. Após debates sobre sua localização, que incluiu áreas nos bairros de Joana Bezerra, Santo Amaro, Ibura, Centro do Recife e um loteamento na Várzea, definiu-se pela instalação no antigo Engenho do Meio.

Em 1967, a instituição passou a integrar oficialmente o sistema federal de educação superior, adotando o nome de Universidade Federal de Pernambuco (UFPE). Posteriormente, em 2006, foram criados dois novos campi: o Centro Acadêmico do Agreste (CAA), em Caruaru, e o Centro Acadêmico de Vitória (CAV), localizado em Vitória de Santo Antão. A UFPE está entre as mil melhores universidades do mundo e é atualmente a 11º melhor universidade do País e a melhor do Norte-Nordeste, segundo o Ranking Universitário Folha. Atualmente a UFPE possui 120 cursos de graduação, sendo 8 a distância e 112 presencial. Possui também 88 departamentos, entre eles o Departamento de Engenharia Cartográfica (DECART) (UFPE, 2025).

3.2 Centro de Tecnologia e Geociências

A Escola de Engenharia de Pernambuco foi fundada em 3 de junho de 1895, configurando-se como a primeira instituição dedicada ao ensino de engenharia nas regiões Norte e Nordeste e a quarta mais antiga do Brasil. Criada sob responsabilidade do governo estadual, sua instalação ocorreu em um contexto de modernização marcado pela expansão ferroviária e pela intensificação do processo de urbanização do Recife no final do século XIX. Em 1946, a instituição foi incorporada à recém-criada Universidade do Recife (UR), juntamente com outras escolas superiores existentes em Pernambuco. Em 1967, a UR passou a integrar o sistema federal de ensino superior, adotando a denominação de Universidade Federal de Pernambuco (UFPE).

Atualmente, a antiga Escola de Engenharia constitui o Centro de Tecnologia e Geociências (CTG), localizado no Campus Recife da UFPE. O CTG mantém a tradição na formação em engenharia e ampliou sua atuação para áreas correlatas, oferecendo cursos de graduação em Engenharia de Alimentos, Engenharia Biomédica, Engenharia Cartográfica e de Agrimensura, Engenharia Civil, Engenharia de Controle e Automação, Engenharia Elétrica, Engenharia Eletrônica, Engenharia de Energia, Engenharia de Materiais, Engenharia Mecânica, Engenharia de Minas, Engenharia Naval, Engenharia de Produção, Engenharia Química, Engenharia de Telecomunicações, Geologia, Química Industrial e Oceanografia (UFPE, 2025).

3.3 Departamento de Engenharia Cartográfica

O Departamento de Engenharia Cartográfica (DECART) fica localizado no Centro de Tecnologia e Geociências, e faz parte do Campus Recife da UFPE. Este departamento possui áreas abertas, acessíveis ao público, e áreas restritas, acessíveis apenas para usuários autenticados. Diante disso, existe a necessidade de um portal de informações único, que conecte requisitos de localização às informações descritivas e que isso auxilie na melhor administração, consulta e conhecimento do departamento em estudo, visando promover uma gestão mais eficiente dos recursos e patrimônios da universidade. (UFPE, 2025).

3.4 Banco de Dados

A estrutura computacional compartilhada e integrada que armazena um conjunto de dados e metadados é chamado de banco de dados (Rob e Coronel, 2011). Um banco de dados é uma coleção organizada de informações ou dados, estruturada de forma que permite fácil acesso, gerenciamento e atualização. Ele pode ser usado para armazenar diversos tipos de informações, desde dados simples de contato até informações complexas de empresas, sistemas financeiros e redes sociais.

Conforme destaca Fonseca (2016), com o surgimento do modelo relacional de dados, foi necessário o desenvolvimento de uma linguagem adequada para a manipulação desses dados, assim, na década de 70, com os estudos de Edgar F. Codd, um pesquisador da IBM (*International Business Machines Corporation*), surgiu a linguagem SQL, sigla inglesa que quer dizer *Structured Query Language*, que pode ser traduzido livremente como Linguagem

de Consulta Estruturada. O sucesso da linguagem foi tamanho que obrigou o ANSI (*American National Standards Institute*) a padronizar a linguagem.

Para o gerenciamento de bancos de dados, utilizam-se softwares específicos denominados Sistemas Gerenciadores de Banco de Dados (SGBD). Esses sistemas têm como função principal armazenar e disponibilizar informações sempre que necessário, assumindo o papel de conjunto de programas responsáveis pela administração dos dados. Assim, o SGBD assume o controle do acesso, da manipulação e da organização das informações, dispensando a aplicação cliente de realizar essas funções. Além disso, fornece uma interface que possibilita aos usuários inserir, modificar ou consultar dados. No caso dos bancos de dados relacionais, essa interface é composta por APIs ou drivers do próprio SGBD, que interpretam e executam comandos na linguagem SQL (DATASUS, 2019).

De acordo com Casanova et al. (2005), os bancos de dados passaram a ocupar papel fundamental nos sistemas de informação, tanto na etapa de desenvolvimento quanto na fase operacional. Esse processo de evolução ocorreu em virtude do avanço de tecnologias consistentes voltadas ao armazenamento e ao tratamento de dados tradicionais, com destaque para os sistemas de gerenciamento de bancos de dados objeto-relacionais (SGBD-OR).

3.4.1 Banco De Dados Geográficos

O armazenamento de dados geográficos pode ser feito de forma matricial (raster) ou vetorial. O armazenamento através de linhas e colunas onde cada célula possui coordenada específica e um valor associado é chamado de matricial. Já o formato vetorial pode ser representado por ponto, linha ou polígono, definidos pelo menos, por um par de coordenadas (Câmara e Monteiro, 2005).

De acordo com Ferreira (2003), a expressão “dato geográfico ou espacial” é empregada para ilustrar elementos do mundo físico por meio de duas partes: sua localização geográfica, que se refere à sua posição conectada a um sistema de coordenadas específico; e suas características descritivas.

A união da tecnologia de Sistemas de Informação Geográfica (SIG) com bancos de dados descritivos leva à formação de bancos de dados geográficos, essenciais para a análise e gestão espacial dos dados. Dentro do contexto da UFPE, essa combinação pode ser utilizada para desenvolver uma ferramenta

web institucional, que é uma plataforma online interativa que reúne e oferece informações geoespaciais pertinentes à universidade. Ademais, a criação de um banco de dados geográficos consistente possibilita a integração de várias fontes de informações, incluindo imagens de satélite, levantamentos topográficos, sensores urbanos e dados cadastrais, promovendo assim uma atualização constante e colaborativa do sistema.

3.5 Sistema de Informações Geográficas

Os Sistemas de Informações Geográficas (SIG) são uma poderosa ferramenta de manipulação, visualização e análise de dados geoespaciais, ou seja, informações que apontem fatos e fenômenos que ocorrem no espaço geográfico, sendo eles importantes para a tomada de decisão. Como destaca Melo et. al. (2008), a capacidade de um SIG diante da geração de bancos de dados espaciais e processamento de informações tornam esses sistemas ferramentas que eliminam dificuldades em obter uma descrição espacial detalhada da região em estudo. Segundo Câmara (2005) a diferença mais significativa entre um SIG e um sistema de informação tradicional é a sua habilidade de armazenar tanto os atributos descritivos quanto as geometrias de diversas categorias de dados geográficos. Dessa forma, para cada terreno em um registro urbano, um SIG mantém, além de dados descritivos como quem é o proprietário e o montante do IPTU, também a informação geométrica que inclui as coordenadas que definem os limites do terreno.

3.6 QGIS

O QGIS é um software livre e de código aberto voltado para Sistemas de Informação Geográfica (SIG), amplamente reconhecido por sua flexibilidade e robustez na manipulação, análise e visualização de dados geoespaciais. Sua arquitetura modular permite a integração de diferentes componentes que atendem a diversas necessidades em projetos geotecnológicos. Foi utilizado, em conjunto com outras aplicações, para alcançar os objetivos principais do presente trabalho. O QGIS facilitou a publicação dos resultados em ambiente web, por meio da integração QGIS Desktop, Server e Web.

3.7 PostgreSQL

PostgreSQL é um Sistema de Gerenciamento de Banco de Dados Relacional de código aberto, o que significa que ele organiza e gerencia dados armazenados em tabelas. Cada tabela é composta por um conjunto nomeado de

linhas. Todas as linhas de uma tabela contêm o mesmo conjunto de colunas, sendo que cada coluna possui um tipo de dado específico. Embora as colunas sigam uma ordem fixa em cada linha, é importante lembrar que o SQL não garante uma ordem definida para as linhas dentro da tabela (a menos que sejam ordenadas explicitamente para exibição). As tabelas são organizadas em bancos de dados e um conjunto de bancos de dados administrados por uma única instância do servidor PostgreSQL forma um cluster de banco de dados (PostgreSQL *Global Development Group*, 2023).

3.7.1 PostGIS

O PostGIS é uma extensão espacial gratuita e de código aberto, desenvolvida sobre o sistema de gerenciamento de banco de dados objeto relacional PostgreSQL. Ele permite que objetos geoespaciais sejam armazenados e gerenciados diretamente no banco de dados. O PostGIS amplia os recursos do banco de dados relacional PostgreSQL adicionando suporte para armazenamento, indexação e consulta de dados geoespaciais. Com o PostGIS é possível armazenar dados espaciais, pesquisar e recuperá-los rapidamente com base em sua localização, filtrar e analisar, medindo distâncias e áreas, cruzando geometrias, salvando a geometria do buffer, processar geometrias, armazenar e processar dados raster, geocodificar, além de ter suporte para uso de ferramentas em outros softwares como o QGIS, GeoServer, entre outros (Postgis *Project*, 2025).

3.8 Tecnologias de Posicionamento: GNSS e Correção RTK (Real Time Kinematic)

O sistema GNSS (*Global Navigation Satellite System*) é uma tecnologia que possibilita a determinação precisa de posições geográficas na superfície terrestre por meio da recepção de sinais emitidos por satélites. Entre os sistemas GNSS mais utilizados destacam-se o GPS - *Global Positioning System* (Estados Unidos), GLONASS (Rússia), Galileo (União Europeia) e BeiDou (China) (Hofmann-Wellenhof; Lichtenegger; Collins, 2008). Tais sistemas são amplamente empregados em levantamentos topográficos, mapeamentos e aplicações geodésicas, devido à sua capacidade de fornecer coordenadas tridimensionais com alta precisão em tempo real.

Para aprimorar a precisão do posicionamento, utiliza-se o método RTK (*Real Time Kinematic*), que consiste na correção em tempo real dos dados

coletados por um receptor móvel (*rover*), a partir de um receptor fixo (base) situado em um ponto conhecido. O método RTK utiliza as fases das ondas portadoras dos sinais GNSS, em vez das informações de código, o que possibilita a obtenção de precisão centimétrica, essencial para trabalhos de engenharia e georreferenciamento detalhado (Leick, 2004). Esse método corrige erros causados por fatores atmosféricos, relógio dos satélites e outras interferências, transmitindo as correções do receptor base para o *rover* por meio de rádio ou outras formas de comunicação (Utiugova, 2019). Dessa forma, o RTK oferece coordenadas com alta confiabilidade e rapidez, permitindo aplicações dinâmicas em campo, como topografia de áreas complexas e monitoramento geotécnico (El-Mowafy, 2013).

4 METODOLOGIA

A metodologia adotada para o desenvolvimento deste trabalho foi estruturada em etapas sequenciais, conforme ilustrado no fluxograma da Figura 1, visando garantir a organização, a rastreabilidade e a consistência das informações utilizadas.

Inicialmente, realizou-se o levantamento de dados, dividido em duas vertentes principais: (i) análise da planta fornecida pela Superintendência de Infraestrutura (Sinfra) e (ii) levantamento em campo. A partir destes insumos, foram obtidas informações detalhadas sobre as salas, contemplando sua localização, numeração e nomenclatura, bem como os responsáveis e os patrimônios associados a cada ambiente.

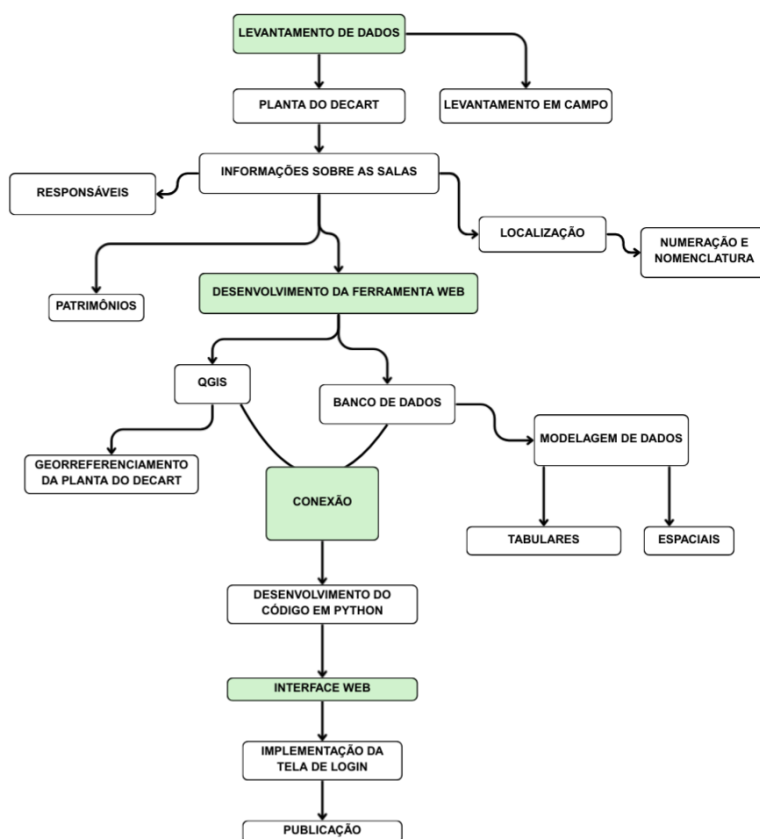
Na etapa subsequente, procedeu-se ao desenvolvimento da ferramenta web, integrando informações espaciais e tabulares. O processo iniciou-se com o georreferenciamento da planta do DECart, executado no software QGIS. A seguir, estruturou-se o banco de dados, que passou por um processo de modelagem visando o armazenamento e a manipulação de dados espaciais e tabulares de forma eficiente.

A integração entre os dados geoespaciais e o banco de dados foi viabilizada pela etapa de conexão, elemento central do sistema. Esta conexão serviu de base para o desenvolvimento do código em Python, responsável por implementar as funcionalidades da aplicação, incluindo a comunicação com o banco de dados e a manipulação de camadas geográficas.

Com a base de programação estabelecida, foi projetada a interface web, permitindo a interação do usuário final com o sistema. Nesta etapa, implementou-se a tela de login, visando garantir segurança no acesso às informações. Por fim, procedeu-se à publicação da aplicação, tornando-a acessível em ambiente web e permitindo sua utilização para fins de gestão, consulta e atualização de dados.

A metodologia descrita garantiu que o desenvolvimento do sistema seguisse um fluxo lógico, contemplando desde a coleta e organização das informações até a implementação final da ferramenta, atendendo aos requisitos técnicos e funcionais estabelecidos no projeto.

Figura 1 – Fluxograma das etapas metodológicas



Fonte: Autora (2025).

4.1 Levantamento de dados

O levantamento de dados constituiu a primeira etapa do processo, sendo subdividido em:

4.1.1 Planta baixa do DECART

Análise e extração das informações contidas na planta baixa disponibilizada pela Sinfra, contendo a disposição física e o dimensionamento dos ambientes.

Figura 2 – Planta da disposição das salas do DECart



Fonte: Autora (2025).

4.1.2 Levantamento de dados em campo

Aquisição das coordenadas geoespaciais do Centro de Tecnologia e Geociências (CTG) utilizando receptores GNSS CHCNAV de alta precisão, operando com o método RTK. Essa técnica permitiu a obtenção de coordenadas com acurácia centimétrica em tempo real. Os pontos de controle obtidos serviram como base para o georreferenciamento da planta arquitetônica, garantindo a precisão posicional do projeto. Nessa etapa também foi feita a

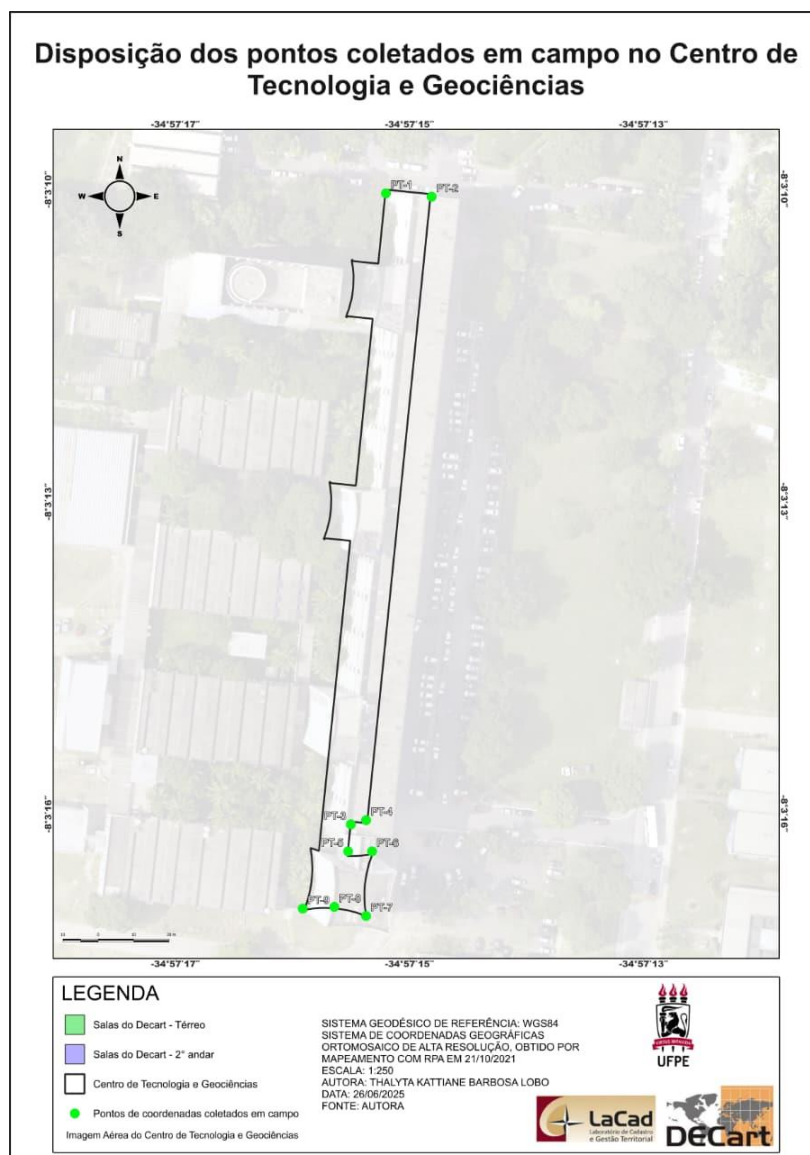
verificação presencial das condições físicas, confirmação de dimensões, localização e funcionalidades de cada sala;

Figura 3 – Levantamento de pontos em campo



Fonte: Autora (2025).

Figura 4 – Planta de pontos coletados em campo



Fonte: Autora (2025).

4.2 Informações sobre as Salas

Com base no levantamento, foram consolidadas informações referentes a:

- **Localização:** Posicionamento espacial preciso de cada ambiente, considerando coordenadas geográficas e relação com outros elementos da planta.
- **Numeração e Nomenclatura:** Padronização dos códigos identificadores e dos nomes atribuídos aos ambientes, garantindo uniformidade.

- **Responsáveis:** Registro das pessoas responsáveis pela utilização dos espaços.
- **Patrimônios:** Catalogação dos bens móveis e equipamentos presentes em cada ambiente, vinculados aos dados da sala correspondente.

4.3 Desenvolvimentos da Ferramenta Web

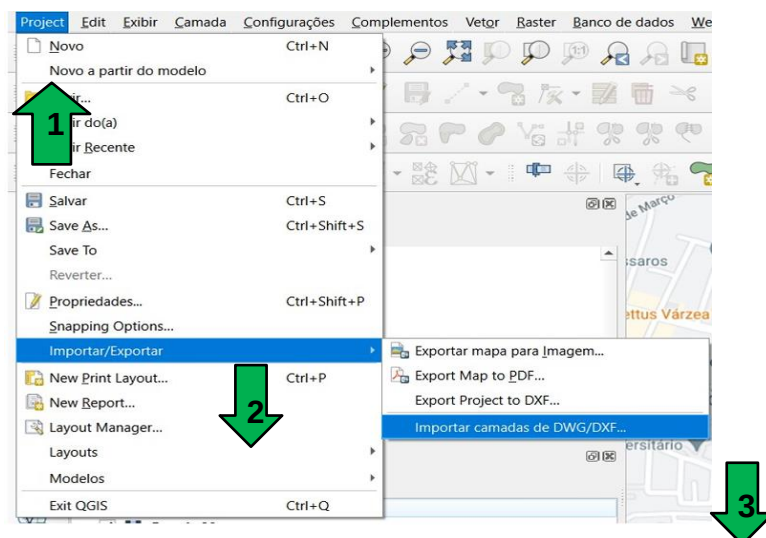
4.3.1 Georreferenciamento da Planta Baixa

O georreferenciamento foi realizado para posicionar corretamente a planta baixa, em um sistema de coordenadas oficial. Esse processo garante que o desenho esteja em sua localização precisa, esse passo foi crucial para a continuidade do trabalho.

A planta baixa foi recebida no formato DWG (desenho vetorial utilizado em programas de CAD) e não possuía informações de localização. Para determinar sua posição real, foram utilizados pontos de controle obtidos em campo com um receptor GNSS de alta precisão, modelo CHCNAV. Esses pontos foram salvos no formato txt, com coordenadas no sistema WGS 84.

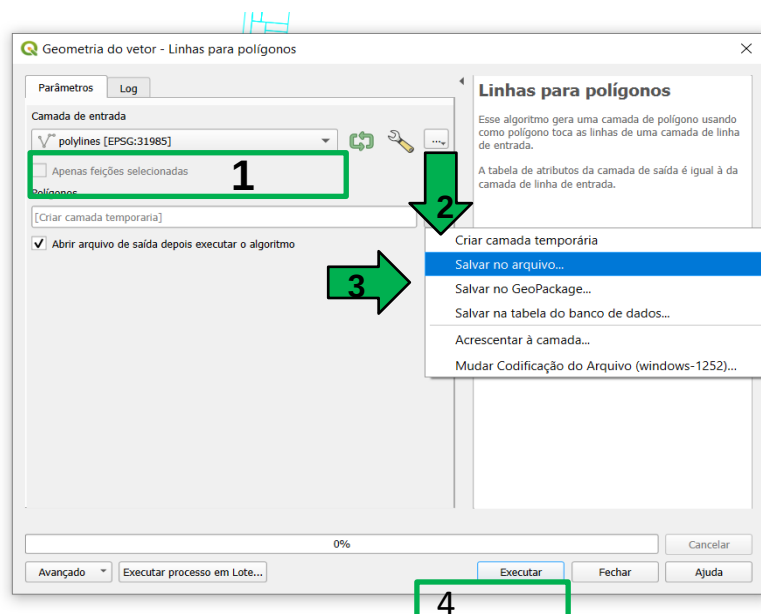
Inicialmente, o arquivo DWG foi importado no software QGIS 3.40.9 (FIGURA 4) que permite a leitura desse formato. Ao importar o desenho para o QGIS as feições vem em “polylines” e para a criação de uma tabela com informações coerentes, fez-se necessário a transformação dessas “polylines” em polígonos, essa transformação foi feita no próprio QGIS, utilizando a ‘Caixa de Ferramentas de Processamento’ o caminho seguido foi: Geometria do Vetor>Linha para polígonos, nessa etapa a camada foi gerada no formato Shapefile que é mais adequado ao ambiente de SIG (FIGURAS 5).

Figura 4 – Passo a passo para importação de arquivo DWG no QGIS



Fonte: Autora (2025).

Figura 5 – Transformar linha em polígono e gerar o shapefile



Fonte: Autora (2025).

Posteriormente o georreferenciamento foi feito com a ferramenta “Georreferenciador” (Camada>Georreferenciador) do próprio QGIS. Nesse procedimento, foram escolhidos pontos que existiam tanto no desenho quanto nos dados coletados no local, neste caso foram utilizados os vértices. Esses pontos foram distribuídos de forma equilibrada pela planta, o que ajuda a reduzir distorções e a aumentar a precisão do ajuste. Na ferramenta de Georreferenciar, existem duas formas de fazer, em uma delas os pontos são selecionados no

objeto a ser referenciado e os pontos correspondentes são selecionados no mapa, na outra os pontos são igualmente selecionados no objeto a ser referenciado, porém as coordenadas do ponto real são inseridas manualmente, no presente trabalho foi utilizada a segunda forma. Para o alinhamento da planta, foi utilizada a transformação Polinomial de 1ª ordem, que corrige deslocamentos, rotações e alterações de escala para que o desenho se encaixe no local correto. O método de reposicionamento adotado foi o “Vizinho Mais Próximo”, indicado para preservar os valores originais dos dados.

Após a configuração, o processo foi executado, resultando em um novo arquivo já georreferenciado no sistema SIRGAS 2000 / UTM Zona 25S. Esse arquivo foi adicionado ao projeto no QGIS para ser usado nas etapas seguintes do trabalho.

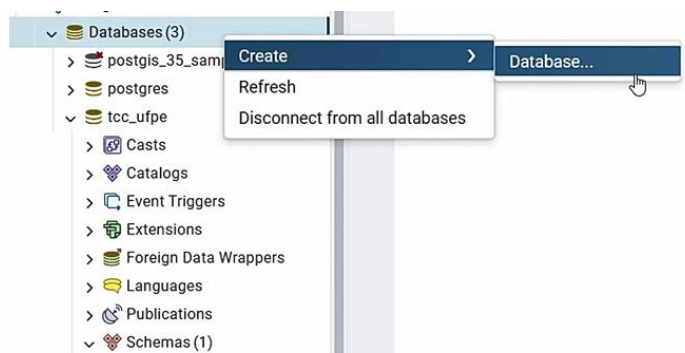
Como não havia dados iniciais associados às feições, foi necessário inserir informações básicas para possibilitar a posterior vinculação ao banco de dados criado. Para isso, abriu-se a tabela de atributos da camada e todos os registros existentes foram excluídos. Em seguida, procedeu-se à inserção manual dos dados, feição por feição. Na camada DEPART foram criados dois campos: “PREDIO”, no qual foi atribuída a informação CTG, e “SALA”, correspondente ao número de cada sala, conforme planta baixa fornecida e verificação realizada na etapa 4.2.2 do presente trabalho. Os demais atributos foram adicionados posteriormente, por meio da integração com o banco de dados, conforme será detalhado na sequência do trabalho. Concluída essa etapa, deu-se prosseguimento ao desenvolvimento do banco de dados.

4.3.2 Banco de Dados

4.3.2.1 Criação do Banco de Dados

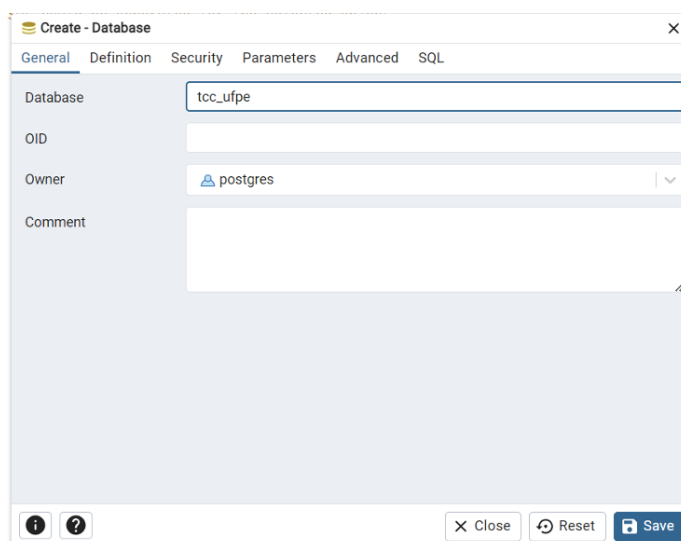
Inicialmente, procedeu-se à criação do banco de dados relacional, adotando-se como sistema gerenciador o PostgreSQL, em razão de sua extensão espacial denominada PostGIS, a qual possibilita o armazenamento e a manipulação de dados geoespaciais. Para a administração do sistema, utilizou-se o pgAdmin 4, interface gráfica destinada ao gerenciamento do PostgreSQL. A conexão ao servidor PostgreSQL 17 foi realizada a partir do pgAdmin 4, exigindo a inserção da senha cadastrada durante a instalação do software. Em seguida, foi criado o banco de dados denominado tcc_ufpe, por meio da própria interface do sistema (Figuras 6 e 7).

Figura 6 – Criar o Banco de Dados no PostgreSQL



Fonte: Autora (2025).

Figura 7 – Criar o Banco de Dados no PostgreSQL

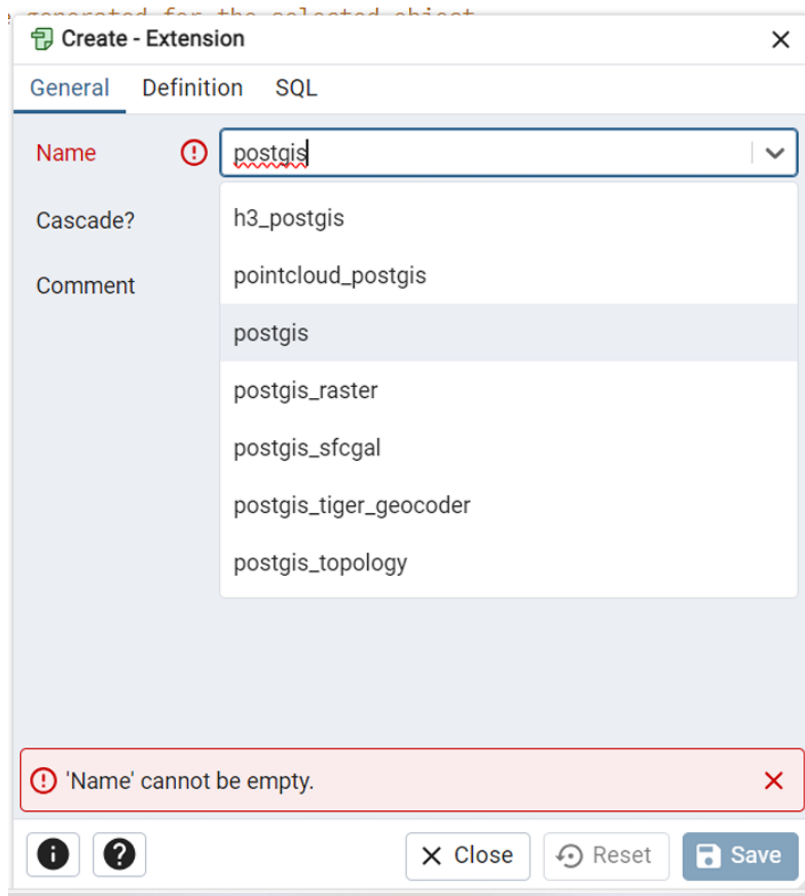


Fonte: Autora (2025).

4.3.2.2 Instalação da Extensão Espacial

Com o banco de dados devidamente criado, tornou-se necessário habilitar recursos de suporte espacial. Para isso, adicionou-se a extensão PostGIS (Figura 8), incorporada ao banco de dados por meio da ferramenta de gerenciamento de extensões do pgAdmin 4. A instalação foi concluída com êxito, permitindo o tratamento de informações geoespaciais no ambiente de armazenamento.

Figura 8 – Criar extensão espacial



Fonte: Autora (2025).

4.3.2.3 Estruturação das Tabelas

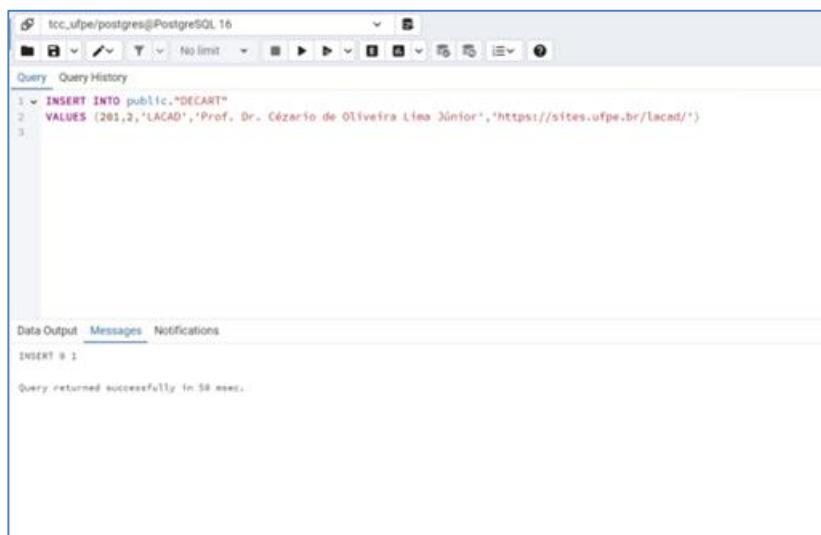
No banco de dados criado, foram desenvolvidas duas tabelas para a organização das informações. A primeira, denominada 'decart', foi estruturada com cinco colunas (sala, andar, descrição, responsável e site), sendo destinada ao registro das informações da feição vetorial previamente criada e ajustada no software QGIS. A segunda, denominada 'patrimonios', foi composta por seis colunas (sala, patrimônio, descrição, marca, modelo, situação), destinada ao gerenciamento de dados referentes aos bens patrimoniais. A criação das tabelas ocorreu por meio do pgAdmin 4, dentro do esquema do banco de dados.

4.3.2.4 Inserção dos Dados

Posteriormente, realizou-se a inserção dos registros com o auxílio da ferramenta Query Tool, disponível no pgAdmin 4. O comando SQL utilizado foi o INSERT INTO (Figura 9), o qual permitiu a inclusão das informações na ordem previamente definida pelas colunas de cada tabela. Como os dados foram

inseridos em conformidade com a estrutura estabelecida, não foi necessária a especificação explícita da ordem dos campos no comando.

Figura 9 – Inserir dado na tabela



Fonte: Autora (2025).

4.3.2.5 Verificação do Preenchimento das Tabelas

Para verificar o correto preenchimento das tabelas, utilizou-se o comando 'SQL SELECT * FROM nome_da_tabela' (Figura 10). Esse procedimento permitiu conferir os registros inseridos, assegurando que os dados foram armazenados adequadamente, em conformidade com a estrutura previamente definida.

Figura 10 – Verificar preenchimento da tabela

The screenshot shows a PostgreSQL query window with the query `select * from public."DECART"` executed. The results are displayed in a table with 5 columns: SALA, ANDAR, DESCRICAO, RESPONSÁVEL, and SITE. The table contains 15 rows of data. A status bar at the bottom indicates 'Total rows: 15' and 'Query complete 00:00:00.198'. A green message box in the bottom right corner says 'Successfully run. Total'.

SALA	ANDAR	DESCRICAO	RESPONSÁVEL	SITE
1	1	Museu de topografia Prof. Barretinho	Profª. Drª Maria de Lourdes de Aquino Macedo Gonçalves	https://miquino.wixsite.com/museutopografiaufpe/
2	1	LACOST	Profª. Drª Karoline Paes Jamur	https://www.ufpe.br/lacost
002A	1	SALA DE PROFESSOR	Prof. Dr. Francisco Jaime de B. Mendonça	Não possui
002B	1	SALA DE PROFESSOR	Profª. Drª Karoline Paes Jamur	Não possui
3	1	LASENSO	Profª. Drª Ana Lucia B. Candeias	https://www.ufpe.br/lasenso/
4	1	SALA DE PROFESSOR	Profª. Drª Karoline Paes Jamur	Não possui
5	1	LATOP	Profª. Drª Maria de Lourdes de Aquino Macedo Gonçalves	https://www.ufpe.br/latop
005A	1	LATOP	Profª. Drª Maria de Lourdes de Aquino Macedo Gonçalves	https://www.ufpe.br/latop
005AT	1	SALA DE AULA	Profª. Drª Karoline Paes Jamur	Não possui

Fonte: Autora (2025).

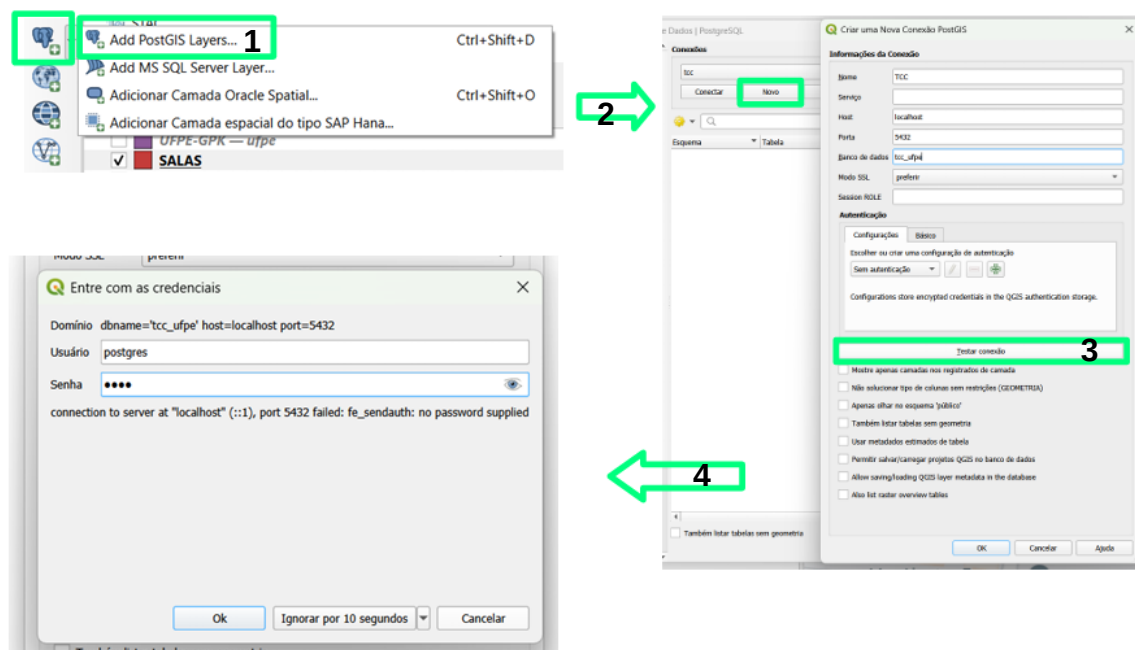
4.3.3 Conexão do Banco de Dados ao QGIS

4.3.3.1 Estabelecimento da Conexão

Para possibilitar a integração entre o banco de dados e o ambiente de geoprocessamento, realizou-se a conexão do PostgreSQL ao software QGIS. Esse procedimento teve como finalidade permitir o acesso, a visualização e a manipulação direta das informações armazenadas no banco de dados.

A conexão foi realizada por meio do menu de gerenciamento de fontes de dados do QGIS, selecionando-se a opção PostgreSQL. Em seguida, foram inseridas as informações necessárias para configuração da conexão, incluindo nome do banco de dados, usuário, senha, host e porta. Após o preenchimento dos parâmetros e a realização do teste de conexão, confirmou-se o acesso ao banco de dados tcc_ufpe (Figura 11).

Figura 11 – Conectar o QGIS ao PostgreSQL



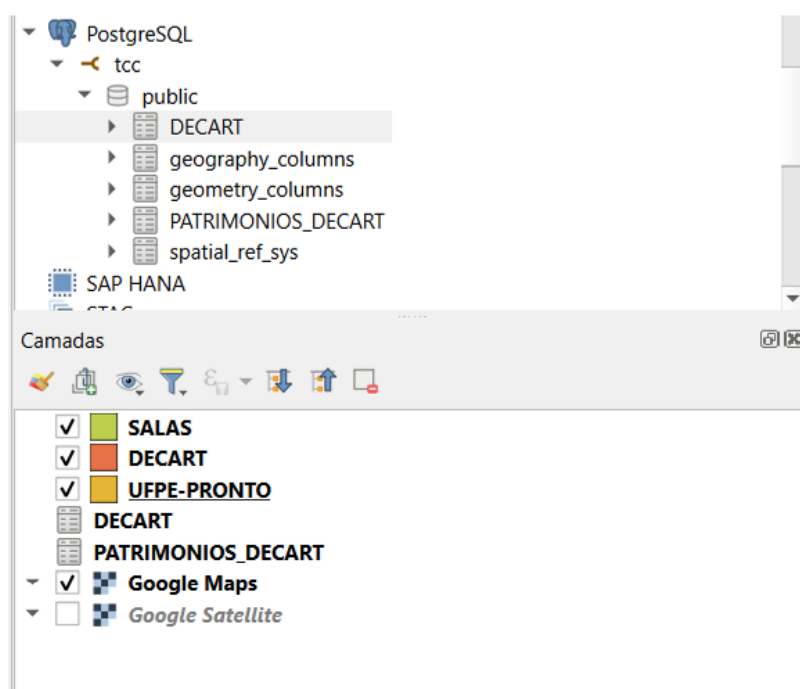
Fonte: Autora (2025).

4.3.3.2 Visualização das Tabelas Relacionais

Com a conexão devidamente configurada, tornou-se possível acessar as tabelas criadas anteriormente, permitindo a manipulação das informações diretamente no QGIS. Esse procedimento viabilizou a integração entre os dados relacionais e as feições espaciais, assegurando maior eficiência no processo de análise e representação cartográfica.

As tabelas puderam ser visualizadas na aba Navegador, onde, ao selecionar a tabela desejada com um duplo clique, esta foi automaticamente adicionada à aba de Camadas do software (Figura 12). Como se tratam de dados exclusivamente relacionais, as tabelas foram exibidas sem geometria associada, restringindo-se ao conteúdo alfanumérico armazenado no banco de dados.

Figura 12 – Visualização das tabelas relacionais no Qgis



Fonte: Autora (2025).

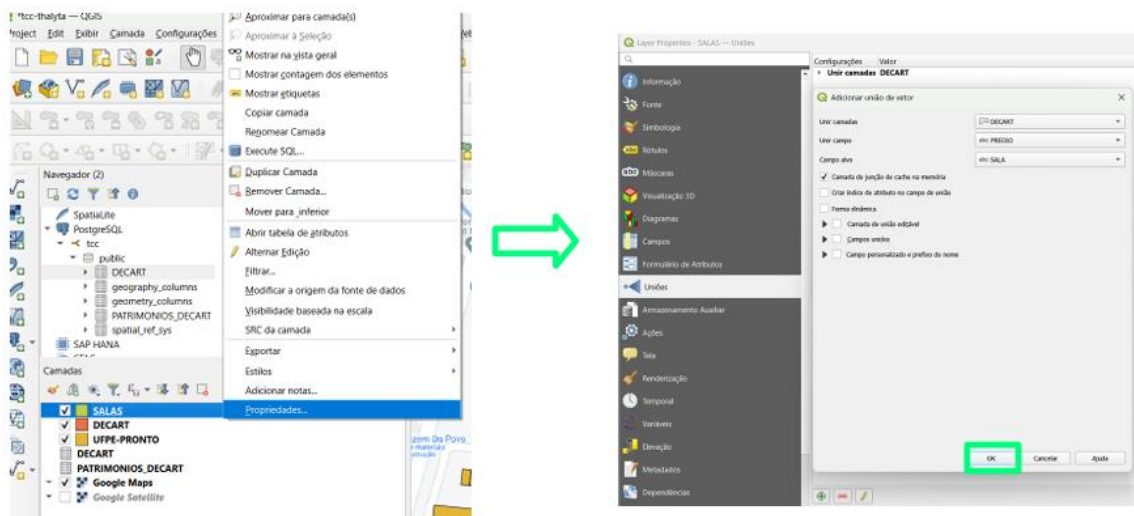
4.3.3.3 Vinculação das Tabelas Relacionais às Camadas Vetoriais

Com a conexão das tabelas relacionais ao QGIS concluída, procedeu-se à integração dessas informações com a camada vetorial previamente criada. Para realizar essa vinculação, utilizou-se a ferramenta de união de atributos, acessível por meio do menu de propriedades da camada vetorial.

O procedimento consistiu em selecionar a camada desejada, acessar a aba Propriedades, escolher a seção Uniões e adicionar uma nova vinculação, indicando a tabela do banco de dados e os campos correspondentes a serem integrados (Figura 13). É fundamental que os campos escolhidos contenham informações equivalentes entre a tabela relacional e a camada vetorial. Caso haja divergência nos dados, o QGIS não realizará a operação, retornando uma mensagem de erro. Ressalta-se que os nomes dos campos não precisam ser idênticos, desde que os atributos correspondam corretamente, garantindo que cada registro alfanumérico seja associado à feição espacial correta.

Essa integração possibilitou que os dados armazenados no banco fossem incorporados à camada vetorial, permitindo análises espaciais e visualização conjunta dos atributos e da geometria, aumentando a precisão e a eficiência do geoprocessamento.

Figura 13 – Vincular a tabela relacional com a camada vetorial



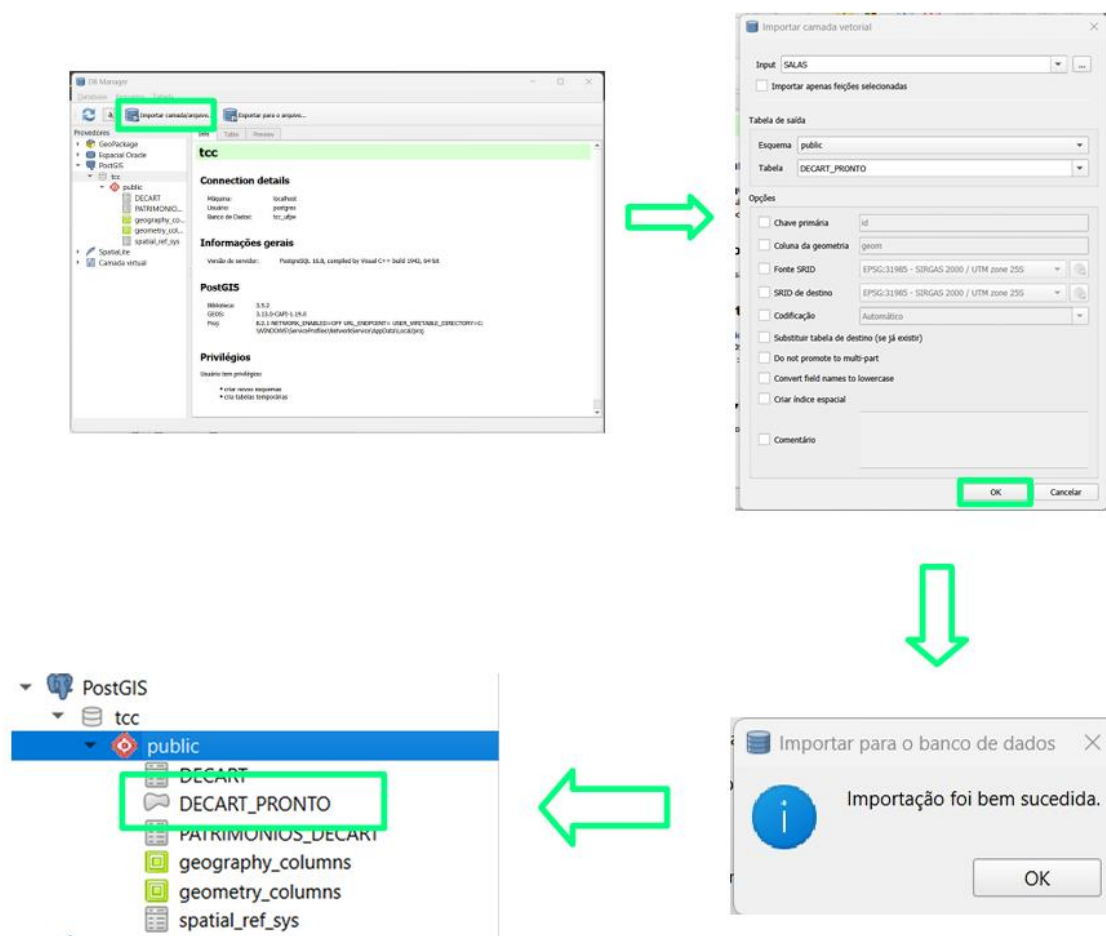
Fonte: Autora (2025).

4.3.3.4 Importação das Informações Espaciais para o PostgreSQL

Após a camada vetorial ter sido integrada às informações descritivas do banco de dados, procedeu-se à importação dos dados espaciais para o PostgreSQL, que até então armazenava apenas atributos alfanuméricos. Esse processo permitiu que tanto o ambiente de geoprocessamento quanto o banco de dados possuísem informações completas, possibilitando o trabalho simultâneo e integrado entre os dois sistemas.

A importação foi realizada por meio da ferramenta Gerenciador BD do QGIS, utilizando-se o complemento para conexão com o PostGIS. No procedimento, selecionou-se o banco de dados de destino, clicou-se em “Importar camada/arquivo” e, na janela subsequente, definiu-se a camada vetorial a ser importada, o esquema de armazenamento e o nome da nova tabela a ser criada (Figura 14). Após a conclusão do processo, a visualização da geometria da camada importada foi realizada por meio da função Geometry Viewer do PostgreSQL, garantindo a conferência da correta transferência das informações espaciais para o banco de dados.

Figura 14 – Importar a camada espacial para o PostgreSQL



Fonte: Autora (2025).

4.3.4 Programação da ferramenta web

Após a integração bem-sucedida entre os dados descritivos e geométricos, iniciou-se a etapa de programação da ferramenta web. Para isso, foi utilizada a linguagem Python, que é reconhecida por sua simplicidade, clareza de sintaxe e ampla aplicação em análise de dados e geoprocessamento (Lutz, 2013). No contexto deste trabalho, a biblioteca Folium foi escolhida para a criação de mapas interativos. Folium é baseada na biblioteca JavaScript fonte Leaflet.js, permitindo a construção de mapas dinâmicos diretamente a partir de códigos fonte Python, sem a necessidade de conhecimentos avançados em programação web (*Python Visualization*, 2025). Essa biblioteca possibilita a incorporação de camadas geoespaciais, a criação de marcadores e pop-ups, e a manipulação de dados vetoriais no formato .GeoJSON.

O ambiente de desenvolvimento utilizado foi o Google *Colab*, um ambiente de desenvolvimento baseado em nuvem que permite a execução de código *Python* diretamente no navegador. Ele oferece recursos como integração com o Google Drive e a capacidade de compartilhar *notebooks* interativos. O *Colab* é amplamente utilizado para análise de dados, aprendizado de máquina e visualização interativa, proporcionando um ambiente colaborativo e acessível para desenvolvedores e pesquisadores (Google, 2025). Antes de iniciar a programação, as camadas de interesse foram exportadas do QGIS no formato .GeoJSON e carregadas no Google Drive. Essa abordagem garante acesso rápido aos dados e facilita a leitura das camadas diretamente pelo Colab. Durante a programação, cada camada foi carregada no ambiente *Python*, permitindo a manipulação e a estilização dos dados geoespaciais, bem como a construção do mapa interativo final, que pode ser visualizado e compartilhado facilmente.

4.3.4.1 Primeira etapa – Mapa de acesso público

Como primeira etapa do projeto, foi desenvolvido o mapa de acesso público da ferramenta web, utilizando o arquivo GeoJSON contendo informações das salas, prédios, andares, descrições, responsáveis e sites. Cada feição foi estilizada com base no atributo *decart_andar*, permitindo distinção visual dos andares através de cores específicas. Foram configurados *tooltips* e *pop-ups* para exibição de informações detalhadas das salas ao passar o mouse ou clicar nos elementos do mapa.

Foram incorporadas diversas camadas base, como *OpenStreetMap*, *CartoDB Positron*, *CartoDB Dark Matter*, *Stamen Terrain* e *Esri Satellite*, permitindo alternância entre diferentes visões do mapa. Além disso, foram adicionadas ferramentas interativas, incluindo régua de medição (*MeasureControl*) para cálculo de distâncias e áreas, exibição de coordenadas ao clicar (*LatLngPopup*) e duas barras de pesquisa configuradas para buscar pelos atributos *decart_descricao* e *SALA*. Uma barra de pesquisa apresentou o texto personalizado: “Digite o Nome do Laboratório”, e a outra “Digite o Número da Sala” centralizando a feição correspondente ao resultado da busca, facilitando a localização de salas ou laboratórios específicos, conforme código no ‘Apêndice A’.

4.3.4.2 Segunda etapa – Mapa de acesso restrito

A etapa subsequente foi realizada de forma semelhante, utilizando arquivo GeoJSON contendo os patrimônios do DECART. O procedimento manteve a configuração de camadas vetoriais, estilos condicionais, *tooltips*, *pop-ups* e camadas base. Diferentemente da primeira etapa, esta fase terá acesso restrito a pessoas cadastradas, garantindo que informações sensíveis sobre os patrimônios sejam visualizadas apenas por usuários autorizados. Nessa etapa temos um campo chamado patrimônios que ao clicar em ‘Abrir Relatório’ um pdf com os patrimônios da sala é aberto em uma nova aba do navegador.

A barra de pesquisa nesta etapa foi ajustada para permitir buscas pelas salas, mantendo a consistência na interação e navegação do usuário dentro da ferramenta web, conforme código no ‘Apêndice B’. Essa abordagem permite que cada etapa do projeto seja modular e controlada, atendendo às necessidades de visualização restrita dos dados.

4.3.5 Interface Web

A ferramenta web foi organizada em quatro páginas principais: a interface inicial, que permite escolher o tipo de acesso (público ou restrito); a interface de acesso público; interface de login e a interface de acesso restrito. Na página de escolha de acesso, o usuário seleciona seu tipo de perfil. Caso opte pelo acesso público, acessará um sistema de login e cadastro independente, permitindo que novos usuários criem uma conta fornecendo nome, e-mail, instituição e senha. Após o login, o visitante é direcionado automaticamente à interface pública ferramenta, onde pode visualizar as camadas espaciais geradas no projeto. Já ao selecionar o acesso restrito, o usuário será encaminhado para a página de login, onde deverá inserir suas credenciais previamente cadastradas no banco de dados, obtendo acesso às funcionalidades destinadas à administração. O objetivo desta área restrita é vincular as informações registradas no banco de dados à ferramenta web, permitindo o acesso a dados não públicos.

Para a publicação da ferramenta em servidor web, é necessário criar um domínio e uma hospedagem. Para tanto, foi utilizado o serviço “Infinity Free” (<https://www.infinityfree.com/>), que oferece hospedagem gratuita e ilimitada, suporte a PHP e MySQL, certificado de segurança, criação de contas e subdomínios, além da possibilidade de envio de arquivos para o servidor via cliente FTP. Entretanto, a versão gratuita do serviço não oferece suporte ao

banco de dados PostgreSQL, inicialmente previsto para este trabalho. Diante dessa limitação, o sistema de gerenciamento de banco de dados utilizado passou a ser o MySQL (exclusivamente para armazenamento de dados de login), disponibilizado nativamente pela própria plataforma de hospedagem. Assim, o banco de dados responsável pelo armazenamento das informações de login foi desenvolvido em MySQL, garantindo compatibilidade com o servidor e viabilizando a implementação da ferramenta, conforme será detalhado posteriormente.

A primeira etapa para utilização da hospedagem consiste no cadastro na plataforma Infinity Free, o qual pode ser realizado utilizando uma conta do Google ou preenchendo os dados solicitados. Após o login, o usuário é direcionado à página inicial, na qual é possível gerenciar contas existentes ou criar novas contas para acesso à ferramenta. O processo de criação de conta consiste em etapas padronizadas de configuração do serviço de hospedagem, incluindo a escolha do plano de hospedagem, definição do domínio e fornecimento de informações adicionais para registro da conta. Para o desenvolvimento desta ferramenta, optou-se pelo plano de hospedagem gratuito disponibilizado pelo serviço Infinity Free. O domínio escolhido foi “tcc-thalyta” com a extensão “.ct.ws”, garantindo a disponibilidade do endereço para publicação. Concluídas essas etapas, a plataforma permite o acesso à área de administração do site, possibilitando a visualização de informações gerais da conta, como o nome do usuário, endereço IP, status da conta, além de dados relacionados ao FTP e ao banco de dados MySQL. Adicionalmente, o painel de controle disponibiliza funcionalidades de gerenciamento de arquivos, configuração do site e monitoramento de estatísticas, garantindo o suporte necessário para a manutenção e atualização da ferramenta web.

O Painel de Controle da plataforma utilizada concentra as ferramentas de administração e modificação do site, incluindo gerenciamento de arquivos, bancos de dados, domínios, segurança, programas auxiliares e configurações da conta. Nesta seção, toda a edição e configuração do site foram realizadas. Como a hospedagem gratuita utilizada (Infinity Free) não permite a criação de bancos de dados via comandos SQL, o banco de dados necessário para o armazenamento das credenciais de login foi criado diretamente através do painel de controle da plataforma. O serviço gera automaticamente um nome único para

o banco, neste caso identificado como if0_39730999_tcc_thalyta. Com o banco de dados já disponível, procedeu-se à criação da tabela login por meio da interface phpMyAdmin. A tabela foi estruturada com quatro colunas: CPF, NOME_COMPLETO, USUARIO e SENHA. A coluna CPF foi definida como chave primária (primary key), com tipo de dado VARCHAR e limite de onze caracteres, enquanto as demais colunas também foram definidas como VARCHAR, com limites de 100, 15 e 8 caracteres, respectivamente.

Para fins de teste e validação do sistema, a tabela foi inicialmente preenchida com dez registros fictícios, sendo o primeiro um exemplo de credencial da autora e os demais registros simulados. Na interface phpMyAdmin, você seleciona a tabela criada e tem a opção 'INSERIR', em que você pode ir preenchendo cada dado (Figura 15). Essa etapa permitiu testar a integração entre a tabela de login e a interface da ferramenta web, garantindo que o sistema de autenticação estivesse funcional antes da implementação dos códigos fonte PHP para login e controle de acesso. Com a tabela de login estruturada, iniciou-se a etapa de criação e implementação dos códigos fonte em PHP, responsáveis por estabelecer a conexão entre a interface web da ferramenta e o banco de dados. O PHP é uma linguagem de programação voltada para desenvolvimento de aplicações web, permitindo a comunicação entre o servidor e a interface do usuário, bem como o processamento de dados provenientes do banco.

Figura 15 – Inserir os dados na tabela

A imagem mostra a interface de inserção de dados no phpMyAdmin. No topo, há uma barra de navegação com opções: Visualizar, Estrutura, SQL, Procurar, Inserir (destacado), Exportar, Importar e Operações. Abaixo, há uma tabela com os seguintes campos:

Coluna	Tipo	Função	Nulo	Valor
CPF	varchar(11)			
NOME_COMPLETO	varchar(100)			
USUARIO	varchar(15)			
SENHA	varchar(8)			

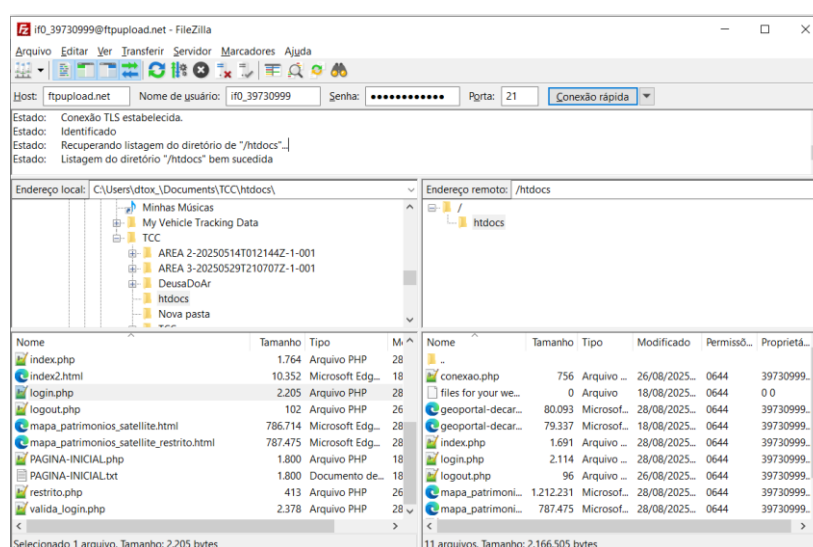
Na parte inferior direita, há um botão "Executar".

Fonte: Autora (2025).

Para o desenvolvimento dos códigos fonte, utilizou-se o editor de código Notepad++, onde foram criados os arquivos responsáveis pela conexão com o

banco de dados, autenticação de usuários e controle de acesso às páginas restritas. No ambiente de hospedagem gratuito utilizado (Infinity Free), os arquivos do site devem ser enviados para o servidor, processo que pode ser realizado diretamente pelo gerenciador de arquivos da plataforma ou por meio de um cliente FTP, como o FileZilla. Este último foi utilizado, pois permite a transferência simultânea de múltiplos arquivos e pastas, tornando o processo mais rápido e eficiente. Para realizar a conexão via FTP, são utilizadas as credenciais fornecidas pelo Infinity Free, disponíveis no painel de controle do serviço (Figura 16). Com a transferência dos arquivos concluída, a ferramenta passou a estar disponível na web, permitindo o acesso à interface do usuário e a integração com o banco de dados por meio dos códigos fonte PHP previamente desenvolvidos, incluindo o arquivo responsável pela conexão com o banco de dados.

Figura 16 – Interface do FileZilla



Fonte: Autora (2025).

Com a tabela de login estruturada e populada, iniciou-se a etapa de desenvolvimento dos códigos fonte em PHP, responsáveis por integrar o banco de dados MySQL à interface web da ferramenta. O primeiro código fonte denominado index.php (Figura 17), configura a página inicial da web, o segundo código fonte, denominado conexao.php (Figura 18), estabelece a conexão com o banco de dados, utilizando as credenciais fornecidas pelo Infinity Free, incluindo host, usuário, senha e nome do banco. Esse arquivo funciona como base para todas as páginas que necessitam acessar o banco de dados.

Figura 17 – Código fonte da página inicial do site em linguagem php

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="UTF-8">
5 <title>Ferramenta Web Do DECART - Acesso</title>
6 <style>
64 </head>
65 <body>
66 <div class="container">
67 <h1> Ferramenta Web Do DECART</h1>
68 <p>Escolha o tipo de acesso:</p>
69 <a href="geportal-decart-estilizado.html" class="btn btn-publico"> Acesso Público</a>
70 <a href="login.php" class="btn btn-restrito"> Acesso Restrito</a>
71 </div>
72 </body>
73 </html>
74

```

Fonte: Autora (2025).

Figura 18 – Código fonte da Conexão do site em linguagem php

```

1 <?php
2
3
4
5 $host = "sql302.infinityfree.com"; // substitua
6 $usuario= "if0_39730999"; // substitua
7 $senha = "Tkbl98245428"; // substitua
8 $banco = "if0_39730999_tcc_thalyta"; // substitua
9
10 $conn = @new mysqli($host, $usuario, $senha, $banco);
11
12 if ($conn->connect_errno) {
13 // Mensagem amigável sem vaziar credenciais
14 http_response_code(500);
15 die("Falha na conexão ao banco de dados. Tente novamente mais tarde.");
16 }
17
18 // Garante acentuação correta
19 $conn->set_charset("utf8mb4");
20

```

Fonte: Autora (2025).

Em seguida, foi desenvolvido o código fonte login.php (Figura 19), que apresenta o formulário de autenticação para o usuário. Por meio desse código fonte, os dados inseridos (usuário e senha) são comparados com os registros armazenados na tabela login. Caso as credenciais estejam corretas, o usuário é redirecionado para a interface restrita da ferramenta web; caso contrário, uma mensagem de erro é exibida, informando falha na autenticação.

Figura 19 – Código fonte do Login do site em linguagem php

```

1  <?php
2  <!DOCTYPE html>
3  <html lang="pt-br">
4  <head>
5  <meta charset="UTF-8">
6  <title>Login - Geoportal</title>
7  </head>
8  <body>
9  <div class="container">
10 <h2>Área Restrita</h2>
11
12 <?php if ($erro): ?>
13 <p class="erro">⚠ Usuário ou senha inválidos!</p>
14 <?php endif; ?>
15
16 <form action="valida_login.php" method="POST">
17 <input type="text" name="usuario" placeholder="Usuário" required>
18 <input type="password" name="senha" placeholder="Senha" required>
19 <button type="submit">Entrar</button>
20 </form>
21
22 <!-- Botão Voltar para a página inicial ou escolha de acesso -->
23 <a href="index.php" class="voltar">← Voltar</a>
24 </div>
25 </body>
26 </html>

```

Fonte: Autora (2025).

Para garantir a segurança das páginas restritas, foi implementado um mecanismo de verificação de sessão, o código fonte denominado validação_login.php. Com ele, apenas usuários autenticados podem acessar as funcionalidades administrativas, prevenindo acessos indevidos (Figura 20). Por fim, foi desenvolvido o código fonte logout.php, responsável por encerrar a sessão do usuário e redirecioná-lo à interface de login (Figura 21).

Figura 20 – Código fonte da Validação do login do site em linguagem php

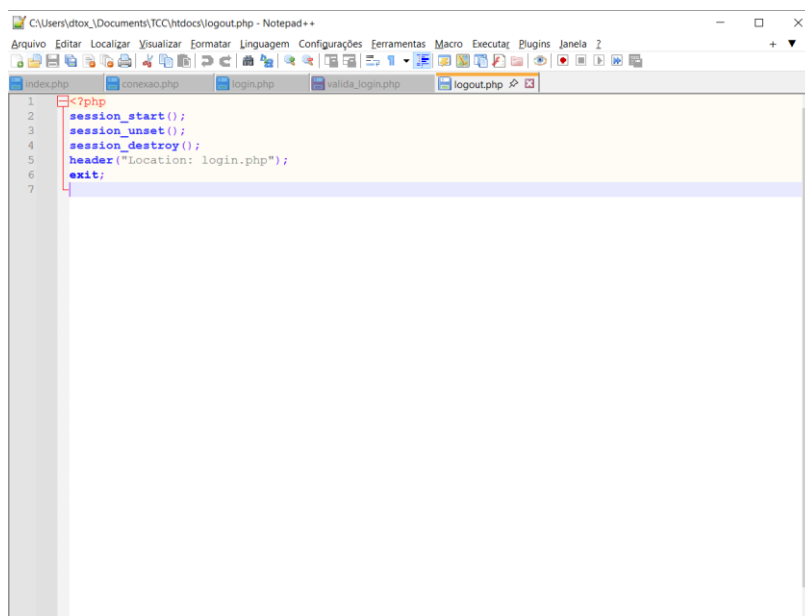
```

1  <?php
2  session_start();
3  include("conexao.php");
4
5  if (isset($_POST['usuario']) && isset($_POST['senha'])) {
6      $usuario = trim($_POST['usuario']);
7      $senha = trim($_POST['senha']);
8
9      // Consulta segura com prepared statement
10     $stmt = $conn->prepare("SELECT USUARIO, SENHA, NOME_COMPLETO
11                             FROM login
12                             WHERE USUARIO = ? LIMIT 1");
13
14     if (!$stmt) {
15
16         $stmt->bind_param("s", $usuario);
17         $stmt->execute();
18         $stmt->store_result();
19
20         if ($stmt->num_rows > 0) {
21
22             $stmt->close();
23
24             // Campos não preenchidos - mensagem + botão voltar
25             echo "<div style='text-align:center; margin-top:50px; font-family:Arial, sans-serif;'>";
26             echo "<h2>⚠ Por favor, preencha usuário e senha.</h2>";
27             echo "<a href='index.php' style='text-decoration:none; padding:10px 20px; background:#e74c3c; co";
28             echo "</div>";
29
30         }
31
32     }
33
34     $conn->close();
35 }

```

Fonte: Autora (2025).

Figura 21 – Código fonte do Logout do site em linguagem php

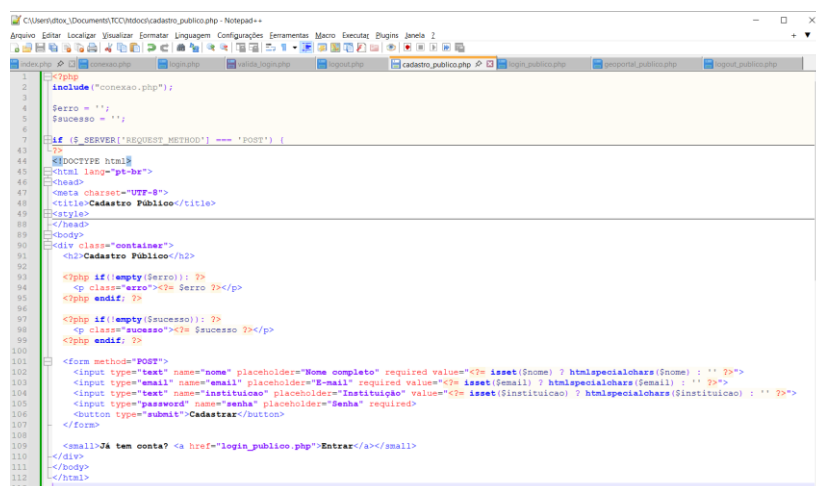


```
1 <?php
2 session_start();
3 session_unset();
4 session_destroy();
5 header("Location: login.php");
6 exit;
7
```

Fonte: Autora (2025).

Além das funcionalidades restritas, foram desenvolvidos códigos PHP específicos para o acesso público, visando possibilitar que usuários não cadastrados pudessem interagir com a ferramenta de forma controlada e segura. O primeiro arquivo, `cadastro_publico.php` (Figura 22), foi desenvolvido para permitir o registro de novos usuários. Este código captura os dados inseridos pelo usuário no formulário de cadastro, tais como nome, e-mail, instituição e senha, realizando validações para garantir que todos os campos obrigatórios estejam preenchidos corretamente. Antes de inserir os dados no banco, o código fonte verifica se já existem registros iguais, prevenindo duplicidades e garantindo a integridade dos dados. Em caso de sucesso, os dados são armazenados na tabela de usuários públicos, e uma mensagem de confirmação é exibida ao usuário, informando que o cadastro foi realizado com êxito.

Figura 22 - Código fonte do Cadastro público do site em linguagem php



```

1  <?php
2  include("conexao.php");
3
4  $erro = '';
5  $sucesso = '';
6
7  if ($_SERVER["REQUEST_METHOD"] === 'POST') {
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
26
```

Figura 23 - Código fonte do Login público do site em linguagem php

```

1  <?php
2  session_start();
3  include("conexao.php");
4
5  $erro = '';
6
7  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
8      $email = trim($_POST['email']);
9      $senha = trim($_POST['senha']);
10
11      // consulta segura
12      $stmt = $conn->prepare("SELECT senha FROM usuarios_publico WHERE email = ?");
13      $stmt->bind_param("s", $email);
14      $stmt->execute();
15      $stmt->store_result();
16
17      if ($stmt->num_rows > 0) {
18          $stmt->close();
19          $conn->close();
20      }
21  }
22
23  <?>
24  <!DOCTYPE html>
25  <html lang="pt-br">
26  <head>
27      <meta charset="UTF-8">
28      <title>Login Público</title>
29  </head>
30  <body>
31  </body>
32  </html>

```

Fonte: Autora (2025).

Por fim, o arquivo `logout_publico.php` (Figura 24) foi desenvolvido para gerenciar a finalização da sessão do usuário público. Este código fonte encerra a sessão ativa, removendo as informações de autenticação armazenadas, e redireciona o usuário de volta à página inicial do sistema. A implementação deste mecanismo garante segurança, prevenindo que usuários não autorizados continuem acessando o sistema web após encerrar suas atividades.

Figura 24 - Código fonte do Logout público do site em linguagem php

```

1  <?php
2  session_start();
3  session_destroy();
4  header("Location: login_publico.php");
5  exit;
6
7  <?>

```

Fonte: Autora (2025).

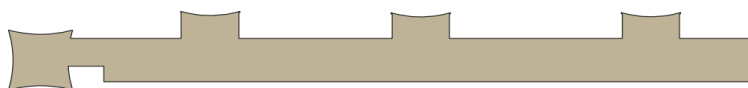
Esses códigos fonte em PHP permitem, portanto, a gestão segura do acesso, vinculando o banco de dados às funcionalidades da ferramenta web e garantindo que apenas usuários autorizados possam acessar informações restritas.

5 RESULTADOS

5.1 Georreferenciamento

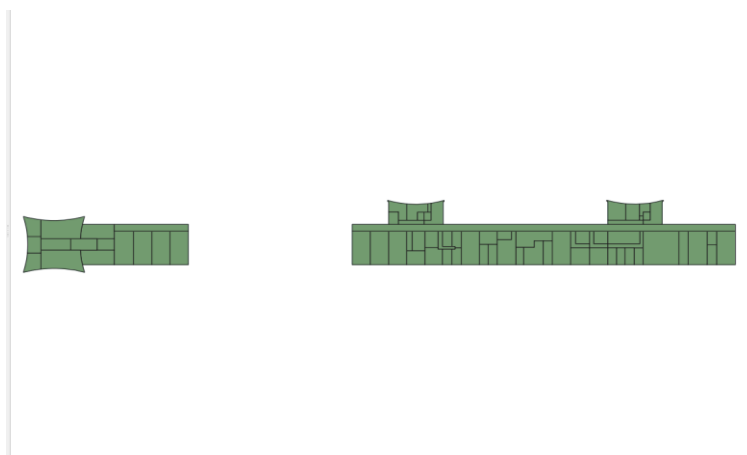
O processo de georreferenciamento constituiu uma etapa fundamental para a correta inserção da planta baixa do Departamento de Engenharia Cartográfica (DECART) no sistema de referência oficial. Inicialmente, a planta foi recebida em formato DWG, desprovida de informações espaciais. Como resultado, obteve-se um arquivo vetorial posicionado corretamente em seu referencial cartográfico, possibilitando sua integração a bases espaciais mais amplas e a vinculação com informações descritivas (Figuras 25, 26, 27 e 28).

Figura 25 – CTG antes do georreferenciamento



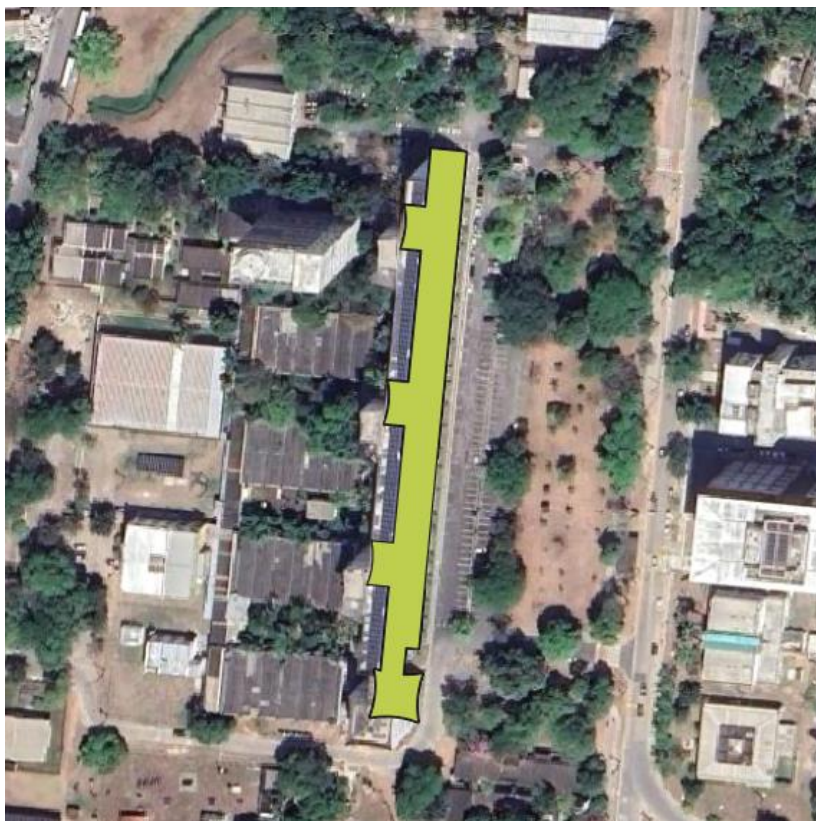
Fonte: Autora (2025).

Figura 26 – DECART antes do georreferenciamento



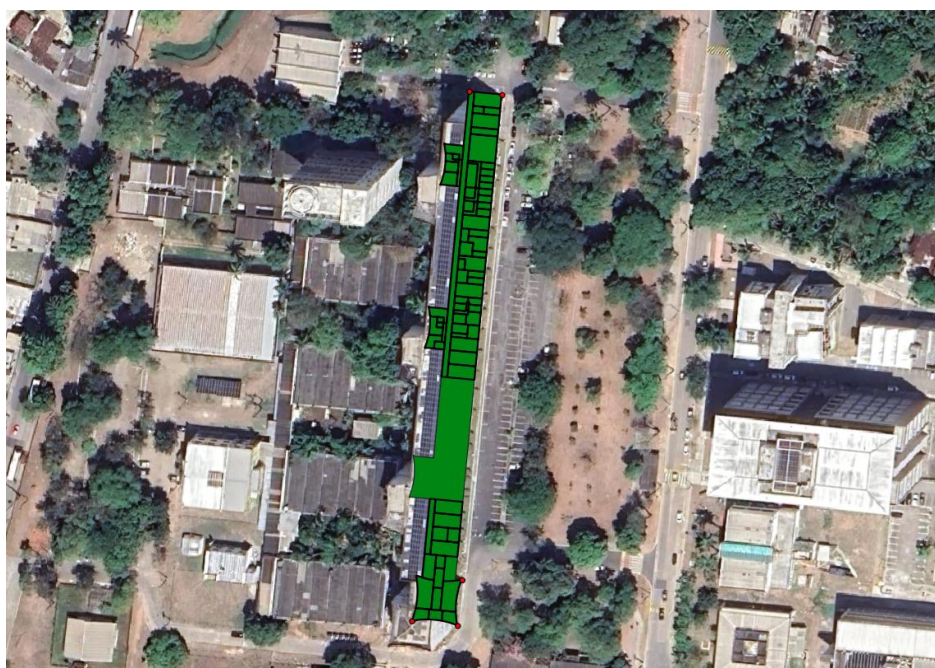
Fonte: Autora (2025).

Figura 27 – CTG georreferenciado



Fonte: Autora (2025).

Figura 28 – DE CART georreferenciado



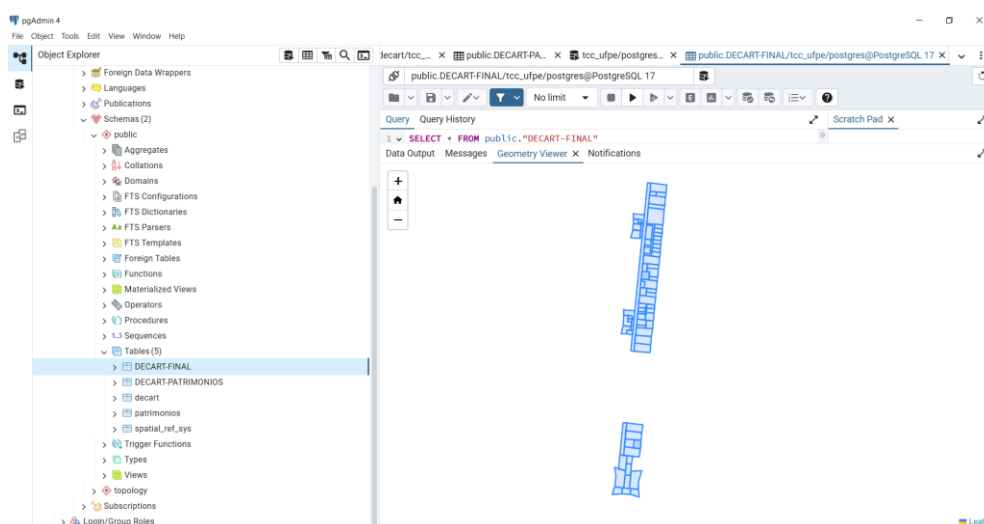
Fonte: Autora (2025).

5.2 Visualização da Camada Espacial no PostgreSQL

As camadas georreferenciadas foram importadas para o PostgreSQL com extensão PostGIS, possibilitando o armazenamento seguro e estruturado das feições espaciais, a realização de consultas espaciais avançadas e a geração de mapas temáticos, além da integração direta com a ferramenta web desenvolvida.

A visualização das camadas no PostgreSQL foi realizada através da ferramenta *Geometry Viewer*, permitindo conferir atributos e geometria de cada feição.

Figura 29 – Visualização da Camada Espacial no PostgreSQL



Fonte: Autora (2025).

5.3 Desenvolvimento da Ferramenta Web

O desenvolvimento da Ferramenta Web constituiu a etapa final do projeto, reunindo as informações coletadas e estruturadas em um ambiente interativo e acessível via web. A ferramenta foi concebida de forma modular, com duas interfaces principais: uma de acesso público, destinada à comunidade acadêmica e ao público externo, e outra de acesso restrito, voltada para a administração interna do DECARTE.

5.3.1 Estrutura e Funcionamento

A estrutura do sistema baseou-se na combinação de diferentes tecnologias. O Python foi utilizado como linguagem principal de programação, em conjunto com a biblioteca Folium, que permite a criação de mapas interativos

baseados na API JavaScript fonte Leaflet. Esse conjunto proporcionou a inserção de camadas vetoriais, estilização de feições e configuração de elementos interativos, como marcadores, pop-ups e caixas de pesquisa.

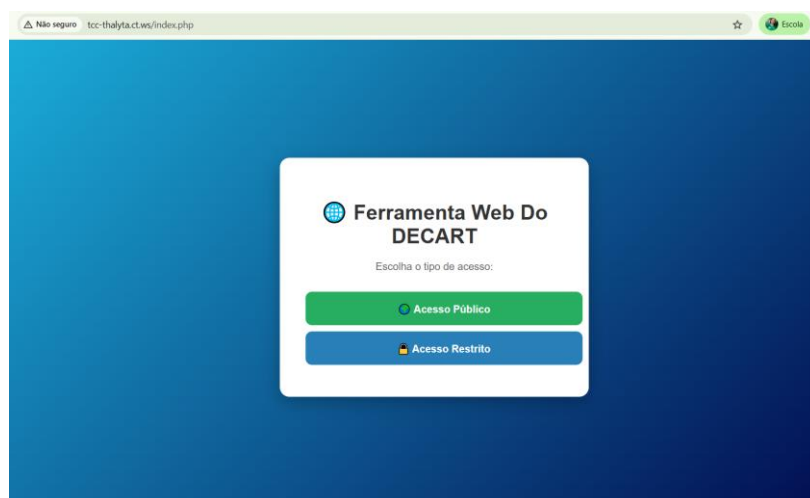
O desenvolvimento foi conduzido no ambiente Google Colab, integrado ao Google Drive para armazenamento e leitura dos arquivos no formato GeoJSON, exportados a partir do QGIS. Essa escolha garantiu agilidade no acesso às camadas e favoreceu a reprodutibilidade do projeto, uma vez que o Colab permite versionamento e compartilhamento dos códigos fonte.

Para a publicação em servidor web, optou-se pela plataforma Infinity Free, que oferece hospedagem gratuita com suporte a PHP e MySQL. Essa etapa exigiu adaptações, uma vez que o serviço não disponibiliza suporte ao PostgreSQL, inicialmente previsto para a integração direta dos dados espaciais. Dessa forma, o PostgreSQL/PostGIS foi utilizado como repositório e ambiente de desenvolvimento local, enquanto o MySQL foi empregado exclusivamente para o gerenciamento de credenciais de login.

5.3.2 Página Inicial da Ferramenta Web

A página inicial da Ferramenta Web foi concebida como ponto de entrada para os usuários, oferecendo duas opções principais de acesso: público e restrito. No acesso público, o usuário é direcionado diretamente ao mapa interativo, onde pode consultar salas, laboratórios e demais dependências do DECART. No acesso restrito, é apresentada a tela de login, na qual o usuário deve inserir suas credenciais previamente cadastradas no banco de dados. Somente após a autenticação, são liberadas funcionalidades administrativas, como consulta e atualização de patrimônios. Essa estrutura garante uma navegação simples e intuitiva, ao mesmo tempo em que preserva a segurança das informações internas.

Figura 30 – Página inicial



Fonte: Autora (2025).

5.3.3 Acesso Público

A interface de acesso público foi concebida para facilitar a navegação de alunos, professores, técnicos e visitantes pelo espaço físico do DECART. Por meio de um mapa interativo, é possível localizar salas, laboratórios e demais dependências do departamento. Cada feição foi estilizada com cores distintas para diferenciar os andares do prédio, proporcionando uma visualização intuitiva. Além disso, foram implementadas funcionalidades de busca, permitindo ao usuário digitar o número da sala ou o nome do laboratório para localizar rapidamente o espaço desejado. O sistema centraliza o mapa na feição correspondente, exibindo em pop-ups informações adicionais, como nome do espaço e descrição associada.

O usuário também dispõe de diferentes camadas base, incluindo OpenStreetMap e Esri Satellite, que possibilitam alternar a visualização conforme a necessidade. Ferramentas adicionais, como régua para medição de distâncias e exibição de coordenadas ao clicar no mapa, complementam a interação.

Figura 31 – Área de login para acesso público

A screenshot of a public login form titled "Login Público". The form is centered on a light blue background. It features two input fields: "E-mail" and "Senha". Below these fields is a blue button labeled "Entrar". At the bottom of the form, there is a link that says "Não tem conta? [Cadastrar-se](#)".

Login Público

E-mail

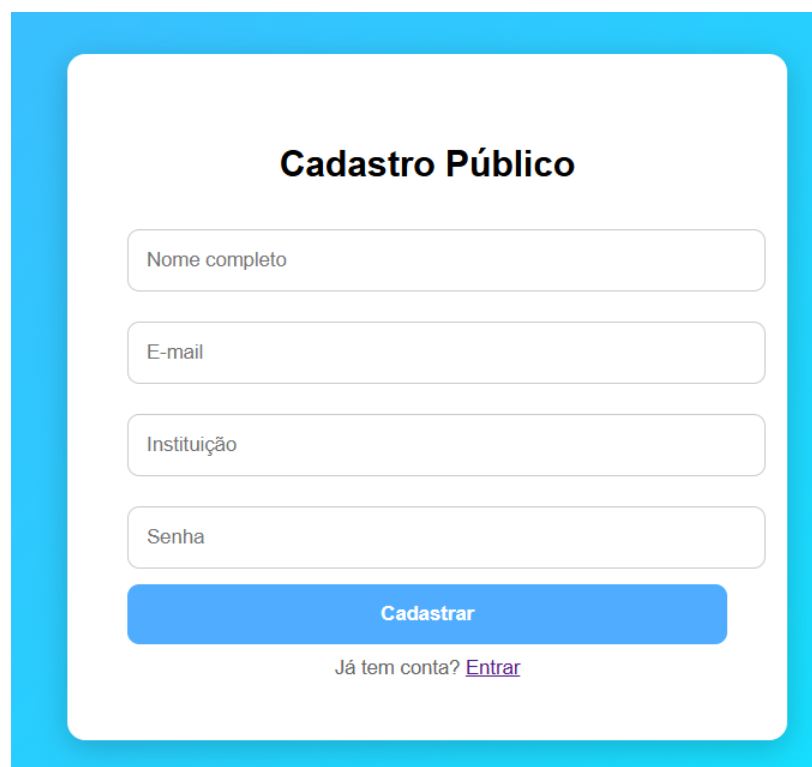
Senha

Entrar

Não tem conta? [Cadastrar-se](#)

Fonte: Autora (2025).

Figura 32 – Área de cadastro para acesso público

A screenshot of a public registration form titled "Cadastro Público". The form is centered on a light blue background. It features four input fields: "Nome completo", "E-mail", "Instituição", and "Senha". Below these fields is a blue button labeled "Cadastrar". At the bottom of the form, there is a link that says "Já tem conta? [Entrar](#)".

Cadastro Público

Nome completo

E-mail

Instituição

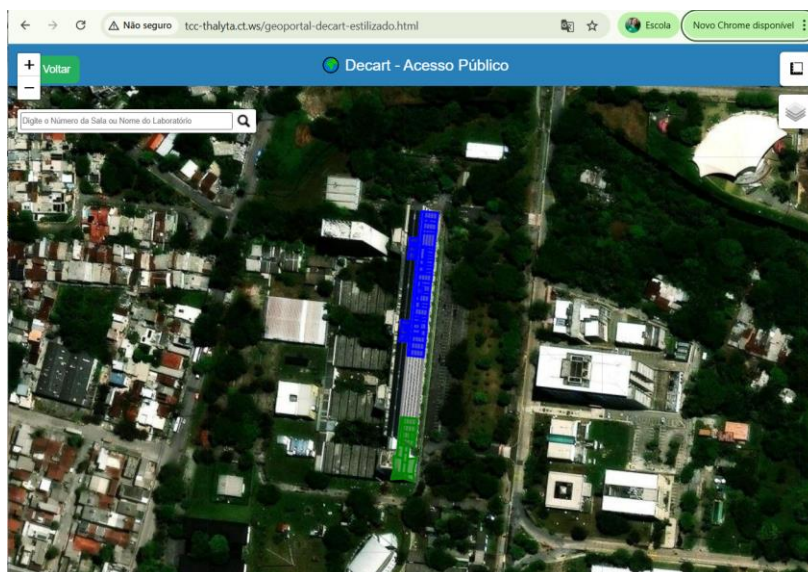
Senha

Cadastrar

Já tem conta? [Entrar](#)

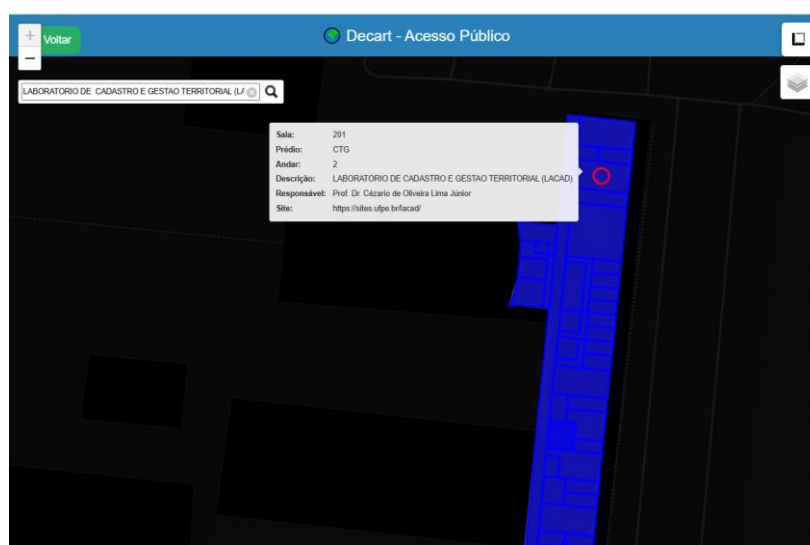
Fonte: Autora (2025).

Figura 33 – Ferramenta web de Acesso Público



Fonte: Autora (2025).

Figura 34 – Pesquisa de Laboratório na área de acesso público



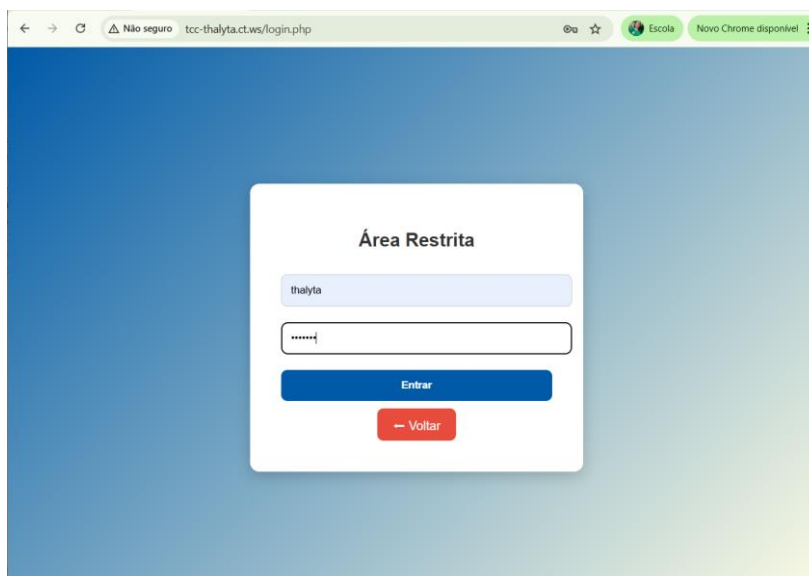
Fonte: Autora (2025).

5.3.4 Acesso Restrito

A área de acesso restrito foi projetada para atender às necessidades administrativas do DECart. Nessa interface, apenas usuários autenticados podem visualizar e gerenciar informações mais sensíveis, como a relação de patrimônios, responsáveis por cada ambiente e observações administrativas.

O sistema de autenticação (Figura 35) foi implementado em PHP, conectado ao banco de dados MySQL hospedado na plataforma Infinity Free.

Figura 35 – Área de login para acesso restrito

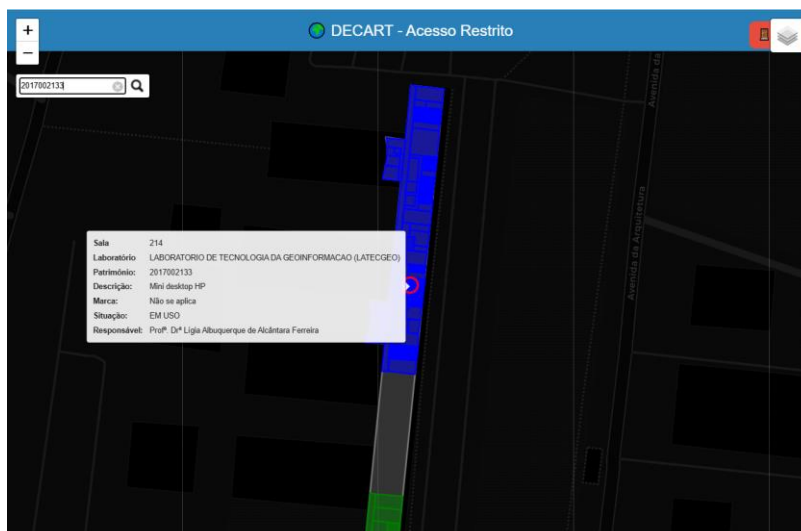


Fonte: Autora (2025).

Após o login, o usuário passa a ter acesso a uma versão ampliada do mapa, na qual cada sala está associada a um conjunto de atributos adicionais, como bens patrimoniais cadastrados e responsáveis por sua gestão.

O uso de camadas vetoriais no formato GeoJSON, em conjunto com filtros de busca, possibilitou ao administrador localizar rapidamente equipamentos ou salas específicas, aumentando a eficiência na resolução de problemas administrativos, como manutenção, alocação de equipamentos ou reuso de espaços.

Figura 36 – Pesquisa de patrimônio em área restrita



Fonte: Autora (2025).

5.3.5 Integração e Potencial de Expansão

Embora inicialmente desenvolvido para o DECART, a Ferramenta Web foi estruturada de forma escalável, possibilitando sua aplicação em outros departamentos da UFPE. Novas camadas e atributos podem ser incorporados ao banco de dados, permitindo a ampliação da ferramenta para contemplar diferentes setores do campus. A utilização de softwares livres como QGIS, PostgreSQL/PostGIS e Folium reforça a sustentabilidade da solução, evitando custos adicionais com licenciamento e garantindo a possibilidade de evolução contínua do sistema.

O protótipo do sistema encontra-se disponível em ambiente web no endereço: <http://tcc-thalyta.ct.ws>, permitindo acesso às funcionalidades descritas.

6 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de uma ferramenta web para auxiliar na gestão e no controle do Departamento de Engenharia Cartográfica (DECART) da Universidade Federal de Pernambuco (UFPE). A proposta uniu processos de levantamento em campo, georreferenciamento de plantas, estruturação de banco de dados geográfico e a implementação de uma ferramenta acessível em ambiente web, resultando em um sistema integrado de apoio à administração acadêmica e patrimonial.

O georreferenciamento da planta baixa permitiu posicionar corretamente a representação do DECART em um sistema de referência oficial, garantindo precisão posicional a partir de pontos coletados com receptores GNSS de alta acurácia. Esse procedimento conferiu confiabilidade às etapas subsequentes de armazenamento e visualização de dados.

No âmbito do banco de dados, a utilização do PostgreSQL com a extensão PostGIS possibilitou a criação de uma base estruturada para o gerenciamento de informações espaciais e descritivas, assegurando consistência e interoperabilidade. A vinculação das camadas vetoriais no QGIS e a importação para o banco evidenciaram a capacidade de integração entre diferentes ambientes tecnológicos.

O resultado mais significativo foi a implementação da ferramenta web, que se apresenta em duas versões: pública, voltada à comunidade acadêmica e ao público externo, e restrita, destinada à administração interna. A interface pública permite a navegação intuitiva pelos ambientes do DECART, oferecendo recursos de busca e visualização interativa. Já a interface restrita possibilita o gerenciamento de patrimônios e de informações administrativas de forma segura, destacando-se como instrumento de apoio à tomada de decisão.

A publicação em servidor web demonstrou a viabilidade de disponibilizar os dados de maneira acessível e prática, ainda que com limitações técnicas impostas pela hospedagem gratuita. Apesar dessas restrições, o protótipo comprova o potencial de expansão do sistema, que pode ser adaptado para outros departamentos e setores da UFPE, tornando-se um modelo escalável para a gestão institucional baseada em geotecnologias.

Em termos acadêmicos, o trabalho reforça a relevância da integração entre Sistemas de Informação Geográfica, bancos de dados e desenvolvimento

web, evidenciando que tais tecnologias podem ser aplicadas de forma inovadora no contexto universitário. Já em termos práticos, a solução proposta contribui para a melhoria da organização dos espaços e do controle patrimonial do DECART, apresentando-se como um avanço em direção à modernização dos processos administrativos.

Assim, conclui-se que os objetivos estabelecidos foram alcançados, resultando em um sistema funcional que alia precisão cartográfica, gestão de dados e acessibilidade em ambiente web. Como perspectivas futuras, destacam-se a migração para um servidor mais robusto, a integração direta com PostgreSQL/PostGIS no ambiente online e a ampliação da ferramenta para outros setores da instituição.

REFERÊNCIAS

CÂMARA, G.; MONTEIRO, A. M. V. **Conceitos básicos em geoprocessamento**. São José dos Campos: INPE, 2005. Disponível em: <http://www.dpi.inpe.br/gilberto/livro>. Acesso em: 6 jan. 2025.

CÂMARA, G.; ORTIZ, J. M. **Sistemas de informação geográfica para aplicações ambientais e cadastrais: uma visão geral**. Divisão de Processamento de Imagens – DPI. Instituto Nacional de Pesquisas Espaciais – INPE, 2005. Disponível em: <http://www.dpi.inpe.br/gilberto/present/segeo.html>. Acesso em: 6 jan. 2025.

CASANOVA, M. A.; CÂMARA, G.; DAVIS, C.; VINHAS, L.; QUEIROZ, G. R. **Banco de dados geográficos**. 2005.

CONSULTA DE CURSOS – GRADUAÇÃO. Disponível em: <https://sigaa.ufpe.br/sigaa/public/curso/lista.jsf>. Acesso em: 7 jan. 2025.

DATASUS. **Sistema gerenciador de banco de dados (SGBD)**. 2019. Disponível em: <https://datasus.saude.gov.br/glossario/sistema-gerenciador-de-banco-de-dados-sqbd/>. Acesso em: 6 jan. 2025.

EL-MOWAFY, A. **GNSS Applications and Methods**. Boca Raton: CRC Press, 2013.

FERREIRA, K. R. **Interface para operações espaciais em banco de dados geográficos**. 2003. Dissertação (Mestrado) – Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2003.

FONSECA, C. C. da. **Implementação de banco de dados**. Rio de Janeiro: SESES, 2016.

GOOGLE. Ambientes de execução e notebooks do Colab. Disponível em: <https://cloud.google.com/colab/docs/runtimes?hl=pt-br>. Acesso em: 17 abril 2025.

HISTÓRIA da UFPE. Disponível em: <https://www.ufpe.br/institucional/historia>. Acesso em: 14 dez. 2024.

HOFMANN-WELLENHOF, B.; LICHTENEGGER, H.; COLLINS, J. Global Positioning System: Theory and Practice. 5. ed. Wien: Springer-Verlag, 2008.

LEICK, A. GPS Satellite Surveying. 3. ed. New York: Wiley, 2004.

LUTZ, M. Learning Python. 5. ed. Sebastopol: O'Reilly Media, 2013.

MELO, H. A.; CUNHA, J. B. L.; NÓBREGA, R. L. B.; RUFINO, I. A. A.; GALVÃO, C. O. **Modelos hidrológicos e sistemas de informação geográfica (SIG): integração possível**. In: **IX Simpósio de Recursos Hídricos do Nordeste**, 2008.

PYTHON SOFTWARE FOUNDATION. Python Documentation. 2025. Disponível em: <https://www.python.org/>. Acesso em: 17 abril 2025.

POSTGIS PROJECT. *PostGIS Documentation*. Disponível em: <https://postgis.net/documentation/>. Acesso em: 24 mar 2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL Documentation*. Version 15. Disponível em: <https://www.postgresql.org/docs/15/index.html>. Acesso em: 24 mar 2025.

QGIS. **Download**. Disponível em: <https://qgis.org/download/>. Acesso em: 6 jan. 2025.

RANKING UNIVERSITÁRIO FOLHA. Disponível em: <https://ruf.folha.uol.com.br/2024/ranking-de-universidades/principal/>. Acesso em: 14 dez. 2024.

UNIVERSIDADE FEDERAL DE PERNAMBUCO. **Centro de Tecnologia e Geociências**. Recife: UFPE, 2025. Disponível em: <https://www.ufpe.br/ctg>. Acesso em: 11 jul. 2025.

Utiugova, Olga. EMLID. Introdução ao RTK GPS. Disponível em: <https://blog.emlid.com/introduction-to-rtk-gps/>. Acesso em: 14 jul. 2025.

APÊNDICE A – CÓDIGO PYTHON CRIADO NO AMBIENTE GOOGLE COLAB (MAPA DE ACESSO PÚBLICO)

```
import folium
import geopandas as gpd
from folium import plugins
from folium.plugins import Search

# Caminho do arquivo
caminho = "/content/drive/MyDrive/TCC-GEO/decart-web.geojson"

# Ler GeoJSON
gdf = gpd.read_file(caminho)

# □ Converter o campo FOTOS em miniatura clicável
def formatar_imagem(url):
    if isinstance(url, str) and url.startswith("http"):
        return f'<a href="{url}" target="_blank"></a>'
    return "Sem foto disponível"

if "FOTOS" in gdf.columns:
    gdf["FOTOS"] = gdf["FOTOS"].apply(formatar_imagem)

# □ Converter campo Site em link clicável
def formatar_url(url):
    if isinstance(url, str) and (url.startswith("http://") or
url.startswith("https://")):
        return f'<a href="{url}" target="_blank">{url}</a>'
    return "Sem site disponível"

if "decart_site" in gdf.columns:
    gdf["decart_site"] = gdf["decart_site"].apply(formatar_url)

# Criar mapa centralizado com zoom máximo maior
mapa = folium.Map(
    location=[-8.05, -34.9], # centro da UFPE
    zoom_start=16,
    max_zoom=21
```

```

)

# Estilo condicional com cores mais claras e bordas verde floresta /
azul marinho
def estilo(feature):
    andar = feature['properties'].get('decart_andar', '')
    if andar is None:
        andar = ''
    andar = str(andar).upper()

    if andar == "TERREO":
        return {
            "color": "#228B22",          # borda verde floresta
            "weight": 2,
            "fillColor": "#2ecc71",      # preenchimento verde claro
            "fillOpacity": 0.25
        }
    elif andar == "2":
        return {
            "color": "#000080",          # borda azul marinho
            "weight": 2,
            "fillColor": "#3498db",      # preenchimento azul claro
            "fillOpacity": 0.25
        }
    else:
        return {
            "color": "#566573",          # borda cinza escura
            "weight": 2,
            "fillColor": "#95a5a6",      # preenchimento cinza claro
            "fillOpacity": 0.15
        }

# Atributos para tooltip e popup
atributos = [
    'SALA',
    'PREDIO',
    'decart_andar',
    'decart_descricao',
    'decart_responsavel',

```

```

        'decart_site',
        'FOTOS'
    ]

# Camada GeoJSON
geojson_layer = folium.GeoJson(
    gdf,
    name="Camada Vetorial",
    style_function=estilo,
    tooltip=folium.GeoJsonTooltip(
        fields=['SALA', 'PREDIO', 'decart_andar', 'decart_descricao', 'decart_responsavel'],
        aliases=["Sala:", "Prédio:", "Andar:", "Tipo de Sala:", "Responsável:"],
        localize=True
    ),
    popup=folium.GeoJsonPopup(
        fields=atributos,
        aliases=["Sala:", "Prédio:", "Andar:", "Tipo de Sala:", "Responsável:", "Site:", "Foto:"],
        localize=True,
        labels=True,
        parse_html=True # necessário para renderizar <a> e <img>
    )
).add_to(mapa)

# Barra de pesquisa
Search(
    layer=geojson_layer,
    search_label="decart_descricao",
    placeholder="Digite o Nome do Laboratório",
    collapsed=False
).add_to(mapa)

# Barra de pesquisa
Search(
    layer=geojson_layer,
    search_label="SALA",
    placeholder="Digite o Número da Sala",

```

```

        collapsed=False
    ).add_to(mapa)

# Ajustar zoom para a camada
mapa.fit_bounds(geojson_layer.get_bounds())

# Camadas base
folium.TileLayer("OpenStreetMap").add_to(mapa)
folium.TileLayer("CartoDB positron").add_to(mapa)
folium.TileLayer("CartoDB dark_matter").add_to(mapa)
folium.TileLayer(
    tiles="Stamen Terrain",
    name="Stamen Terrain",
    attr="Map tiles by Stamen Design, Data by OpenStreetMap"
).add_to(mapa)
folium.TileLayer(
    tiles="https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}",
    attr="Tiles © Esri",
    name="Esri Satellite",
    overlay=False,
    control=True
).add_to(mapa)

# Ferramentas extras
plugins.MeasureControl().add_to(mapa)
folium.LatLngPopup().add_to(mapa)
folium.LayerControl().add_to(mapa)

# ----- PERSONALIZAÇÃO ESTÉTICA -----

# Salvar mapa temporariamente
temp_file = "/content/ferramenta_web_temp.html"
mapa.save(temp_file)

# Ler HTML gerado pelo Folium
with open(temp_file, "r", encoding="utf-8") as f:
    html = f.read()

```

```
# CSS e cabeçalho customizado
```

```
custom_css = ""
```

```
<style>
```

```
  header {
```

```
    background: #2980b9;
```

```
    color: white;
```

```
    padding: 15px;
```

```
    text-align: center;
```

```
    position: fixed;
```

```
    top: 0;
```

```
    left: 0;
```

```
    width: 100%;
```

```
    z-index: 1000;
```

```
    box-shadow: 0 2px 10px rgba(0,0,0,0.2);
```

```
  }
```

```
  header h1 {
```

```
    margin: 0;
```

```
    font-size: 20px;
```

```
  }
```

```
  header a {
```

```
    position: absolute;
```

```
    left: 15px;
```

```
    top: 15px;
```

```
    text-decoration: none;
```

```
    background: #27ae60;
```

```
    color: white;
```

```
    padding: 8px 12px;
```

```
    border-radius: 8px;
```

```
    font-size: 14px;
```

```
    transition: 0.3s;
```

```
  }
```

```
  header a:hover {
```

```
    background: #1e8449;
```

```
  }
```

```
#map {
```

```
  height: calc(100% - 60px) !important;
```

```
  margin-top: 60px;
```

```
}
```

```
.leaflet-top.leaflet-left {
```

```

        top: 80px !important;
    }
    .leaflet-top.leaflet-right {
        top: 80px !important;
    }
</style>
"""

header_html = """
<header>
    <a href="index.php">␣ Voltar</a>
    <h1>␣ Decart - Acesso Público</h1>
</header>
"""

# Injetar CSS e cabeçalho
html = html.replace("</head>", custom_css + "\n</head>")
html = html.replace("<body>", "<body>\n" + header_html)

# Salvar arquivo final estilizado no Drive
saida = "/content/drive/MyDrive/TCC-GEO/ferramenta    web-decart-
estilizado.html"
with open(saida, "w", encoding="utf-8") as f:
    f.write(html)

```

APÊNDICE B – CÓDIGO PYTHON CRIADO NO AMBIENTE GOOGLE COLAB (MAPA DE ACESSO RESTRITO)

```
import folium
import geopandas as gpd
from folium.plugins import Search, MeasureControl

# Caminho do arquivo
caminho = "/content/drive/MyDrive/TCC-GEO/patrimonios-web.geojson"

# Ler o arquivo
gdf = gpd.read_file(caminho)
gdf = gdf.explode(index_parts=False).reset_index(drop=True)

# □ Converter o campo PATRIMONIOS em link HTML (se for URL bruta)
def formatar_link(valor):
    if isinstance(valor, str) and valor.startswith("http"):
        return f'<a href="{valor}" target="_blank">Abrir  
Relatório</a>'
    return valor

if "PATRIMONIOS" in gdf.columns:
    gdf["PATRIMONIOS"] = gdf["PATRIMONIOS"].apply(formatar_link)

# □ Converter o campo FOTOS em miniatura clicável
def formatar_imagem(url):
    if isinstance(url, str) and url.startswith("http"):
        return f'<a href="{url}" target="_blank"></a>'
    return "Sem foto disponível"

if "FOTOS" in gdf.columns:
    gdf["FOTOS"] = gdf["FOTOS"].apply(formatar_imagem)

# Criar mapa inicial
mapa = folium.Map(location=[0, 0], zoom_start=2, max_zoom=21
)

# ----- CAMADAS BASE -----
```

```

folium.TileLayer("OpenStreetMap").add_to(mapa)
folium.TileLayer("CartoDB positron").add_to(mapa)
folium.TileLayer("CartoDB dark_matter").add_to(mapa)
folium.TileLayer(
    tiles="Stamen Terrain",
    name="Stamen Terrain",
    attr="Map tiles by Stamen Design, Data by OpenStreetMap"
).add_to(mapa)
folium.TileLayer(
    tiles="https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}",
    attr="Tiles © Esri",
    name="Esri Satellite",
    overlay=False,
    control=True
).add_to(mapa)

# ----- Função de estilo -----
def estilo(feature):
    andar = feature['properties'].get('decart_andar', '')
    if andar is None:
        andar = ''
    andar = str(andar).upper()

    if andar == "TERREO":
        return {
            "color": "#228B22",          # borda verde floresta
            "weight": 2,                  # espessura da borda
            "fillColor": "#2ecc71",       # preenchimento verde claro
            "fillOpacity": 0.25           # transparente
        }
    elif andar == "2":
        return {
            "color": "#000080",           # borda azul marinho
            "weight": 2,
            "fillColor": "#3498db",       # preenchimento azul claro
            "fillOpacity": 0.25
        }
    else:

```

```

        return {
            "color": "#566573",          # borda cinza escura
            "weight": 2,
            "fillColor": "#95a5a6",      # preenchimento cinza claro
            "fillOpacity": 0.15
        }

# Campos que queremos mostrar no popup e tooltip
atributos = [
    'PREDIO',
    'SALA',
    'decart_andar',
    'decart_descricao',
    'decart_responsavel',
    'PATRIMONIOS',
    'FOTOS'
]

# ----- GeoJSON -----
geojson_layer = folium.GeoJson(
    gdf,
    name="Patrimônios",
    style_function=estilo,
    tooltip=folium.GeoJsonTooltip(
        fields=['PREDIO', 'SALA', 'decart_andar', 'decart_descricao', 'd
ecart_responsavel'],
        aliases=['Prédio:', 'Sala:', 'Andar:', 'Tipo de
Sala:', 'Responsável:'],
        localize=True
    ),
    popup=folium.GeoJsonPopup(
        fields=atributos,
        aliases=['Prédio:', 'Sala:', 'Andar:', 'Tipo de
Sala:', 'Responsável:', 'Relatório PDF:', 'Foto:'],
        localize=True,
        labels=True,
        parse_html=True
    )
).add_to(mapa)

```

```

# ----- Barra de pesquisa -----
Search(
    layer=geojson_layer,
    search_label='SALA',
    placeholder="Digite a Sala",
    zoom=18,
    collapsed=False
).add_to(mapa)

# ----- Ferramenta de medição -----
mapa.add_child(MeasureControl(
    primary_length_unit='meters',
    secondary_length_unit='kilometers',
    primary_area_unit='sqmeters',
    secondary_area_unit='hectares'
))

# Controle de camadas
folium.LayerControl().add_to(mapa)
mapa.fit_bounds(geojson_layer.get_bounds())

# ----- SALVAR TEMPORÁRIO -----
temp_file = "/content/ferramenta_web_temp.html"
mapa.save(temp_file)

# ----- PERSONALIZAÇÃO ESTÉTICA -----
with open(temp_file, "r", encoding="utf-8") as f:
    html = f.read()

# CSS e cabeçalho com botão "Deslogar"
custom_css = """
<style>
    header {
        background: #2980b9;
        color: white;
        padding: 15px;
        text-align: center;
        position: fixed;

```

```

    top: 0;
    left: 0;
    width: 100%;
    z-index: 1000;
    box-shadow: 0 2px 10px rgba(0,0,0,0.2);
  }
  header h1 {
    margin: 0;
    font-size: 20px;
  }
  header a {
    position: absolute;
    right: 15px;
    top: 15px;
    text-decoration: none;
    background: #e74c3c;
    color: white;
    padding: 8px 12px;
    border-radius: 8px;
    font-size: 14px;
    transition: 0.3s;
  }
  header a:hover {
    background: #c0392b;
  }
  #map {
    height: calc(100% - 60px) !important;
    margin-top: 60px;
  }
  /* Ajustar posição dos controles abaixo do cabeçalho */
  .leaflet-top.leaflet-left {
    top: 80px !important;
  }
  .leaflet-top.leaflet-right {
    top: 80px !important;
  }
}
</style>
"""

```

```

header_html = """
<header>
    <h1>☐ DECart - Acesso Restrito</h1>
    <a href="logout.php">☐ Sair</a>
</header>
"""

# Injetar CSS e cabeçalho
html = html.replace("</head>", custom_css + "\n</head>")
html = html.replace("<body>", "<body>\n" + header_html)

# ----- SALVAR ARQUIVO FINAL -----
saida = "/content/drive/MyDrive/TCC-
GEO/mapa_patrimonios_satellite_restrito.html"
with open(saida, "w", encoding="utf-8") as f:
    f.write(html)

```