

INSTITUTO FEDERAL DO TOCANTINS - CAMPUS COLINAS DO TOCANTINS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DAVI BRITO SOARES

GEOIACTA: DESENVOLVENDO UMA EXPERIÊNCIA GAMIFICADA NO ENSINO
DE GEOGRAFIA

COLINAS DO TOCANTINS - TO
2026

DAVI BRITO SOARES

**GEOIACTA: DESENVOLVENDO UMA EXPERIÊNCIA GAMIFICADA NO ENSINO
DE GEOGRAFIA**

Trabalho de curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Federal do Tocantins Campus Colinas do Tocantins, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Dr. Ronneesley Moura Teles, e coorientação de Esp. André Luís Cardoso Silva.

COLINAS DO TOCANTINS - TO

2026

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

S676g Soares, Davi

Geolacta : Desenvolvendo uma experiência gamificado no ensino de geografia / Davi Soares. – Colinas do Tocantins, TO, 2026.
86 p. : il. color.

Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, Campus Colinas, Colinas, TO, 2026.

Orientador: Dr. Ronneesley Teles
Coorientador: Esp. André Luís Cardoso

1. Aplicações web. 2. Gamificação na educação. 3. Sistemas de informação geográfica. I. Teles, Ronneesley. II. Cardoso, André Luís. III. Título.

CDD 004.2

DAVI BRITO SOARES

GEOIACTA: DESENVOLVENDO UMA EXPERIÊNCIA GAMIFICADA NO ENSINO DE GEOGRAFIA

Trabalho de curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Federal do Tocantins Campus Colinas do Tocantins, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Dr. Ronneesley Moura Teles, e coorientação de Esp. André Luís Cardoso Silva.

Banca Examinadora



Documento assinado digitalmente

RONNEESLEY MOURA TELES

Data: 24/06/2026 17:49:07-0300

Verifique em <https://validar.iti.gov.br>

Dr. Ronneesley Moura Teles

Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente



ANDRE LUIS CARDOSO SILVA

Data: 24/06/2026 10:00:19-0300

Verifique em <https://validar.iti.gov.br>

Esp. André Luís Cardoso Silva

Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente



LAERCIO PONTIN JUNIOR

Data: 24/06/2026 10:34:17-0300

Verifique em <https://validar.iti.gov.br>

Me. Laercio Pontin Junior

Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente



FERNANDO TURIBIO DE MOURA

Data: 24/06/2026 09:16:58-0300

Verifique em <https://validar.iti.gov.br>

Me. Fernando Turibio de Moura

Instituto Federal do Tocantins Campus Colinas do Tocantins

Aprovada em 01 de junho de 2026

Este trabalho é todo dedicado aos meus pais,
pois é graças aos seus esforços, tenho condições
de seguir a minha jornada.

AGRADECIMENTOS

Agradeço a Deus e minha família por terem me dado a oportunidade e as condições de estudar e seguir os meus sonhos. Aos meus amigos e colegas que me ajudaram nessa jornada. E ao professor Ronneesley, que se esforçou não apenas para nos orientar nessa jornada, mas também para melhorar a qualidade do ensino para os alunos do IFTO Campus Colinas.

"Alea iacta est."

[Os dados foram lançados.]

— Caio Júlio César

RESUMO

A gamificação tem se destacado como uma estratégia pedagógica voltada ao aumento do engajamento e da efetividade no ensino de geografia. Nesse contexto, este trabalho apresenta o *Geolacta*, uma plataforma web interativa que propõe a integração de elementos de jogos com conteúdos geoespaciais para auxiliar no aprendizado de localização geográfica. A aplicação foi desenvolvida utilizando Python e o framework Flask no backend, com integração ao banco de dados PostgreSQL estendido com PostGIS para armazenamento e manipulação de dados geográficos. A interface do usuário foi construída com HTMX. Além da mecânica central de adivinhação de locais em mapas, o sistema incorpora um módulo de visualização de dados estatísticos voltado à apresentação de métricas de desempenho e evolução do usuário ao longo do tempo. A proposta busca contribuir para o ensino de geografia por meio de uma abordagem interativa e orientada por dados, oferecendo recursos que podem auxiliar tanto estudantes quanto educadores no acompanhamento do processo de aprendizagem. Por se tratar de uma proposta de caráter experimental, o trabalho concentra-se na concepção, implementação e apresentação da plataforma, estabelecendo fundamentos para futuras avaliações práticas e estudos de validação pedagógica.

Palavras-chave: gamificação. ensino de geografia. sistemas web. geotecnologias. visualização de dados.

ABSTRACT

Gamification has emerged as a pedagogical strategy aimed at increasing engagement and effectiveness in geography education. In this context, this work presents *Geolacta*, an interactive web platform that proposes the integration of game elements with geospatial content to support the learning of geographic location. The application was developed using Python and the Flask framework on the backend, integrated with the PostgreSQL database extended with PostGIS for the storage and manipulation of geographic data. The user interface was built with HTMX. In addition to the core mechanic of guessing locations on maps, the system incorporates a statistical data visualization module designed to present user performance metrics and progress over time. The proposal seeks to contribute to geography education through an interactive and data-driven approach, offering features that may assist both students and educators in monitoring the learning process. As an experimental proposal, this work focuses on the conception, implementation, and presentation of the platform, establishing foundations for future practical evaluations and pedagogical validation studies.

Keywords: gamification. geography education. web systems. geotechnologies. data visualization.

LISTA DE ILUSTRAÇÕES

Figura 1 – GeoGuessr e Google Trends.	15
Figura 2 – Gráfico de pessoas com acesso à Internet ao redor do mundo.	19
Figura 3 – Informações de uma rodada do GeoGuessr.	24
Figura 4 – Pontuação adquirida após um palpite no GeoGuessr.	25
Figura 5 – Arquitetura em camadas do <i>Geolacta</i>	46
Figura 6 – Modelo Entidade-Relacional do <i>Geolacta</i>	48
Figura 7 – Diagrama de sequência para submissão de palpite	50
Figura 8 – Interface de seleção de modos de jogo	62
Figura 9 – Interface de carregamento de sessão	63
Figura 10 – Interface de Jogo	64
Figura 11 – Interface de resultados	65
Figura 12 – Interface de resultados finais	66
Figura 13 – Painel de estatísticas e histórico de perfil do usuário	67
Figura 14 – Gráficos de perfil e estatísticas de desempenho	68
Figura 15 – Interface cartográfica de estatísticas de desempenho	69
Figura 16 – Protótipos <i>desktop</i> : login e tela inicial.	79
Figura 17 – Protótipos <i>desktop</i> : perfil e <i>lobby</i>	80
Figura 18 – Protótipos <i>desktop</i> : <i>gameplay</i> e resultados.	81
Figura 19 – Protótipos <i>mobile</i> : telas de <i>login</i> , registro, seleção de mapas e perfil do usuário.	82
Figura 20 – Protótipos <i>mobile</i> : tela de características do mapa, tela de jogo e tela de exibição de resultados.	83
Figura 21 – Interface de seleção de mapas em dispositivos móveis	84
Figura 22 – Interface de jogo em dispositivos móveis.	85

LISTA DE TABELAS

Tabela 1 – Requisitos funcionais do sistema	39
Tabela 2 – Requisitos não funcionais do sistema	40

LISTA DE ABREVIATURAS E SIGLAS

PNAD	Pesquisa Nacional por Amostra de Domicílios
IBGE	Instituto Brasileiro de Geografia e Estatística
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicações)
UIT	União Internacional de Telecomunicações
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
ORM	<i>Object-Relational Mapping</i> (Mapeamento Objeto-Relacional)
HTML	<i>Hypertext Markup Language</i> (Linguagem de Marcação de Hipertexto)
DOM	<i>Document Object Model</i> (Modelo de Objetos do Documento)
REST	<i>Representational State Transfer</i> (Transferência de Estado Representacional)
OAuth	Protocolo de autorização padrão para delegação de acesso
HTTPS	<i>Hypertext Transfer Protocol Secure</i> (Protocolo de Transferência de Hipertexto Seguro)
CSS	<i>Cascading Style Sheets</i> (Folhas de Estilo em Cascata)
JSON	<i>JavaScript Object Notation</i> (Notação de Objetos JavaScript)
FDD	<i>Feature Driven Design</i> (Design Orientado a Funcionalidades)
GeoJSON	<i>Geographic JavaScript Object Notation</i> (Notação de Objetos Geográficos em JavaScript)
MPV	<i>Mínimo Produto Viável</i>
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
CTA	<i>Call to Action</i> (Chamada para Ação)
KPI	<i>Key Performance Indicator</i> (Indicador-Chave de Desempenho)
OSM	<i>OpenStreetMap</i>

PBF	<i>Protocol Buffer Binary Format</i> (formato binário utilizado para distribuição de dados OSM)
SUS	<i>System Usability Scale</i> (Escala de Usabilidade do Sistema)

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo Geral	16
1.2	Objetivos Específicos	16
1.3	Justificativa	17
1.4	Organização do trabalho	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	Mudanças de paradigmas educacionais	18
2.2	A importância da Gamificação no ensino	21
2.3	O Jogo Geoguessr	23
2.4	O conceito de paisagem e suas aplicações no ensino de Geografia	25
2.5	O Impacto da Análise de Dados no Desempenho do Usuário	28
2.6	Tecnologias utilizadas	30
2.6.1	Git e GitHub	30
2.6.2	Docker	30
2.6.3	Python	31
2.6.4	HTMX	32
2.6.5	Tailwind CSS	33
2.6.6	D3.js	33
2.6.7	Leaflet	34
2.6.8	GeoJSON Maps e Natural Earth	35
2.6.9	Google Maps Platform	35
3	METODOLOGIA DE DESENVOLVIMENTO	37
3.1	Especificação de software	37
3.1.1	Requisitos funcionais	38
3.1.2	Requisitos não funcionais	40
3.2	Desenvolvimento de software	40
3.2.1	Padrão de desenvolvimento	40
3.2.2	Ordem de implementação dos requisitos	41
3.3	Validação de software	42
3.4	Evolução de software	42
4	PROJETO	44

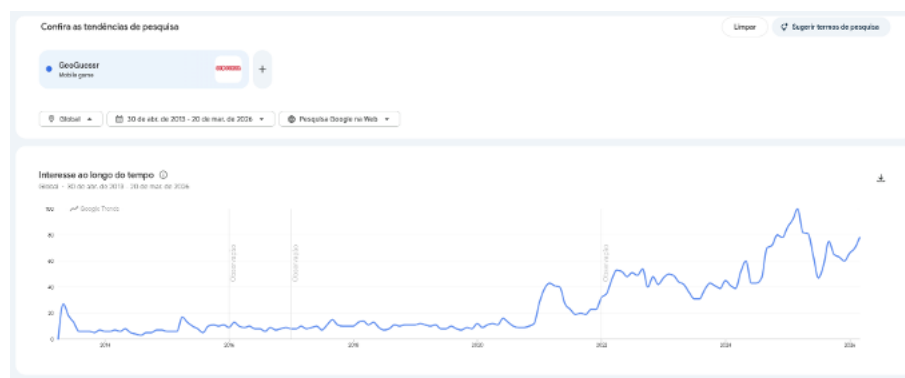
4.1 Visão Geral da Arquitetura	44
4.1.1 MVC e a Camada de Serviços	45
4.1.2 Organização em Camadas	45
4.1.3 Modelo de Dados	47
4.1.4 Fluxo de controle: submissão de palpite	49
4.1.5 Ambiente de Desenvolvimento e Implantação com Containerização	52
4.1.6 Justificativa da arquitetura adotada	53
4.2 Algoritmos propostos	53
4.2.1 Carregamento do mapa e das suas fronteiras	54
4.2.2 Geração de localização aleatória válida	56
4.2.3 Cálculo da Distância Geográfica e Sistema de Pontuação	58
5 RESULTADOS E DISCUSSÃO	61
5.1 Telas desenvolvidas	61
5.2 Ressalvas e Limitações	69
5.2.1 Erro 429 em requisições à API do Google Street View	69
5.2.2 Algoritmos de amostragem mais eficientes	70
5.2.3 Ausência de avaliação com usuários externos	70
6 CONSIDERAÇÕES FINAIS	72
6.1 Trabalhos Futuros	72
REFERÊNCIAS	74
APÊNDICE A PROTÓTIPOS	79
APÊNDICE B INTERFACES MOBILE	84

1 INTRODUÇÃO

Nas últimas décadas, observou-se um crescimento significativo no consumo de *softwares* e plataformas digitais, impulsionado pela ampla disseminação de dispositivos móveis e pelo acesso facilitado à Internet. Dados da Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD Contínua), realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostram que, em 2024, 93,6% dos domicílios brasileiros possuíam acesso à rede, e a posse de telefone celular atingiu o maior percentual da série histórica (97,0%) (IBGE, 2025).

Com a consolidação do mercado de desenvolvimento de *software* e a democratização dos dispositivos eletrônicos, inúmeros aplicativos e *websites* voltados à educação emergiram nas últimas décadas, cada um com características e propostas distintas que visam solucionar um problema específico ou explorar novas abordagens de ensino. Entre esses, destacam-se os jogos que propõem uma maneira interativa e dinâmica de testar conhecimentos geográficos, estimulando o raciocínio multidisciplinar. Aplicações inspiradas no *GeoGuessr*, por exemplo, popularizaram uma mecânica na qual o usuário deve identificar a localização de imagens reais a partir de pistas visuais e alcançaram enorme popularidade nos últimos anos, como evidenciado pelo crescimento consistente nas buscas pela plataforma demonstrado na Figura 1.

Figura 1 – GeoGuessr e Google Trends.



Fonte: Google Trends: capturada em 20/03/2026

Apesar da relevância desse tipo de ferramenta, ainda é comum que essas soluções ofereçam recursos limitados para acompanhar o desempenho do jogador ao longo do tempo. Dessa forma, surge a necessidade de um sistema que, além de proporcionar a experiência lúdica de adivinhação geográfica, também permita ao usuário visualizar seu progresso, analisar seus resultados e acompanhar seu desempenho de forma mais clara.

Entretanto, mesmo que o *GeoGuessr* e aplicativos similares sejam opções interessantes para complementar a educação e o desenvolvimento do conhecimento geográfico, tais ferramentas carecem de um sistema de visualização de dados que demonstre estatísticas envolvendo o progresso e o aprendizado do jogador. A literatura acadêmica reforça que *dashboards* de análise de aprendizagem, quando bem projetados, podem oferecer *feedback* acionável e motivacional, aumentando a motivação e a persistência do usuário no ambiente educacional (Borrella; Ponce-Cueto, 2025). Além disso, *dashboards* que fornecem atualizações em tempo real sobre o progresso de aprendizagem e ferramentas de autoavaliação são apontados como elementos-chave para o engajamento do aluno (Montréal, 2025).

Embora *dashboards* de aprendizagem tragam benefícios teóricos, plataformas comerciais de adivinhação geográfica priorizam o entretenimento e negligenciam as funcionalidades necessárias para o acompanhamento histórico do usuário. Diante disso, emerge o seguinte problema de pesquisa: Como desenvolver um jogo de adivinhação geográfica, que possa registrar e exibir o desempenho do usuário de maneira intuitiva para contribuir com ensino de geografia?

1.1 Objetivo Geral

O objetivo geral deste trabalho é projetar e desenvolver um protótipo funcional de um sistema inspirado no *GeoGuessr*, integrando mecanismos de autenticação de usuários, algoritmos de geração aleatória de localizações dentro de limites geográficos pré-definidos. Juntamente com um módulo analítico de visualização de progresso espacial, que possa permitir com que o usuário realize uma auto-avaliação eficiente a respeito do seu desempenho.

Apesar de discutir os amplamente a literatura envolvendo tópicos relacionados à aplicação de ferramentas de ensino em sala de aula, o trabalho se trata de uma pesquisa experimental com um enfoque tecnológico, não sendo validada em ambientes pedagógicos. Dessa forma, o sistema implementado constitui uma base técnica robusta para estudos futuros, nos quais será possível testar as hipóteses acerca do impacto das visualizações de progresso no aprendizado geográfico.

1.2 Objetivos Específicos

A partir do objetivo geral, foram definidos os seguintes objetivos específicos:

- Implementar um mecanismo de autenticação simples para diferenciar usuários e permitir o armazenamento individualizado do histórico de partidas;
- Criar quatro mapas pré-definidos (países ou continentes) nos quais o usuário possa jogar em áreas delimitadas, gerando localizações aleatórias dentro dos limites de cada mapa;
- Desenvolver um algoritmo de seleção aleatória de coordenadas geográficas restrito aos limites de cada mapa, garantindo variabilidade nas partidas e respeito às fronteiras definidas;
- Criar um componente de visualização geográfica (mapa interativo) que exiba, com pontos coloridos conforme a pontuação obtida, a localização dos palpites do usuário.

1.3 Justificativa

A justificativa para o desenvolvimento deste trabalho fundamenta-se na carência de *feedback* visual estruturado em aplicações de aprendizagem geográfica baseadas em geolocalização, até a confecção desse trabalho. Conforme apontam Borrella e Ponce-Cueto (2025), a representação gráfica de métricas de desempenho tende a elevar o engajamento e a retenção de conhecimento. Nesse sentido, a implementação de um mapa interativo que exiba os palpites do usuário em contraposição a localização real, possui um potencial autoavaliativo e pedagógico maior, suprimindo uma lacuna identificada nas plataformas comerciais existentes.

1.4 Organização do trabalho

A estrutura deste trabalho foi organizada em capítulos que acompanham as etapas do desenvolvimento do sistema. O capítulo 2 apresenta a revisão bibliográfica que fundamenta teoricamente a proposta, abordando os paradigmas educacionais, a gamificação e as tecnologias empregadas. O capítulo 3 descreve a metodologia de desenvolvimento adotada, detalhando as ferramentas, os requisitos e o processo de construção do software.

O capítulo 4 expõe a arquitetura do sistema, os diagramas de modelagem, os algoritmos propostos e as decisões técnicas que nortearam a implementação. Por fim, os capítulos 5 e 6 dedicam-se, respectivamente, à apresentação das interfaces desenvolvidas e à análise crítica dos resultados, culminando com as conclusões do trabalho e as perspectivas para desenvolvimentos futuros.

2 REVISÃO BIBLIOGRÁFICA

A presente revisão bibliográfica teve como propósito estruturar o embasamento teórico para o desenvolvimento de uma ferramenta educacional inspirada no jogo *GeoGuessr*, articulando os conceitos de transformação dos paradigmas de ensino, gamificação, uso da paisagem no ensino de Geografia e análise de desempenho do usuário. Inicialmente, examinou-se o impacto da difusão da Internet e das tecnologias digitais sobre as práticas educacionais, evidenciando a necessidade de se repensarem metodologias tradicionais diante das características das gerações contemporâneas. Nesse contexto, a gamificação consolidou-se como abordagem promissora, ao incorporar elementos como metas, regras, sistemas de *feedback* e participação voluntária, capazes de estimular a motivação intrínseca dos estudantes.

A análise voltou-se, então, ao *GeoGuessr* e ao conceito de paisagem, compreendido como constructo perceptual que mobiliza conhecimentos múltiplos desde aspectos físicos e climáticos até elementos culturais e infraestruturais. Estudos de aplicação do jogo em contextos escolares revelaram ganhos em engajamento, interdisciplinaridade e mesmo na dimensão afetiva dos alunos, ampliando o significado do aprendizado geográfico. Entretanto, identificou-se uma lacuna nas plataformas comerciais consolidadas: a ausência de ferramentas nativas para coleta, sistematização e visualização de métricas de desempenho do jogador. Para superar essa limitação, discutiu-se o papel da análise visual de dados como painéis de desempenho e gráficos de evolução na promoção da autorregulação da aprendizagem, permitindo ao usuário identificar padrões de erro, monitorar seu progresso e refletir sobre suas dificuldades espaciais.

Dessa forma, a revisão bibliográfica conduziu à delimitação do problema de pesquisa e à proposição de um sistema que integra a mecânica lúdica de adivinhação geográfica a um módulo analítico-visual. O objetivo central consistiu em oferecer não apenas uma experiência recreativa, mas um instrumento formativo que transforma cada tentativa em dado passível de análise, contribuindo para a consolidação de conhecimentos geográficos de maneira contínua e reflexiva.

2.1 Mudanças de paradigmas educacionais

Globalmente, a União Internacional de Telecomunicações (UIT) estimou que, em 2025, aproximadamente 6 bilhões de pessoas estavam conectadas à Internet, representando cerca de três quartos da população mundial (UIT, 2025), seguindo uma tendência crescente, que é

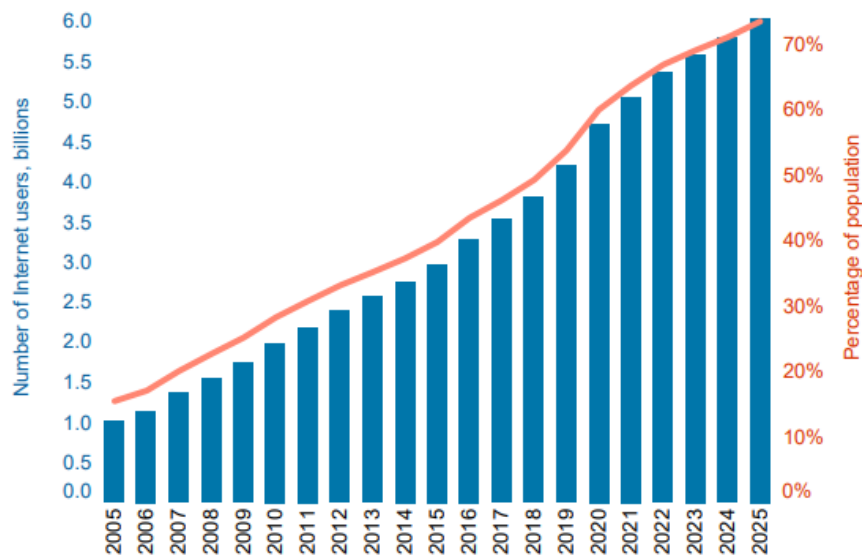
demonstrada na Figura 2.

Figura 2 – Gráfico de pessoas com acesso à Internet ao redor do mundo.

Internet use

Almost three-quarters of the population are online

Individuals using the Internet



Fonte: União Internacional de Telecomunicações

A adoção rápida e quase sem precedentes dessas tecnologias ao redor do mundo alterou profundamente hábitos e costumes de nossa sociedade como um todo. Conforme Policarpo e Santaella (2018), o avanço dos aparatos digitais promove uma reconfiguração contínua dos ambientes socioculturais. Essa transformação conecta-se à perspectiva de Castells (2017), em relação a consolidação da sociedade em rede, uma estrutura social baseada em redes operadas por tecnologias de comunicação que geram e distribuem informação a partir de conhecimento acumulado em nós interligados. Sob essa nova lógica, a interação humana passa a ser mediada pela "virtualidade real", na qual a virtualidade não é uma dimensão separada, mas a própria refundação da realidade através de novas formas de comunicação socializável que integram todas as esferas da vida em uma rede global.

Uma observação prática dessas mudanças pode ser vista de maneira clara em como o acesso à Internet e a ambientes digitais ressignificou vários de nossos conceitos tradicionais de consumo, como amplamente discutido por Schwab (2019). E redes sociais alteram fortemente a maneira como nos comunicamos ou compartilhamos informações (Felipe, 2013). Todas essas

transformações em nossa sociedade podem corroborar com mudanças nos paradigmas de ensino utilizados.

De acordo com Kuhn (1998), paradigmas no domínio científico são referidos como pressupostos teóricos, valores e práticas compartilhadas por uma comunidade, que servem como fundação para diferentes estudiosos formarem suas ideias. No campo da educação, paradigmas podem se referir aos diferentes conjuntos de concepções sobre o ensino, aprendizagem, aquisição de conhecimento e papel dos sujeitos no ambiente educativo em um determinado contexto histórico e social.

Diferentes paradigmas emergiram conforme o tempo e as demandas de cada geração (Libâneo, 1999), sendo empregado o termo “tendências pedagógicas” para se referir aos mais influentes do Brasil na educação, diz o seguinte a respeito da educação tradicional, ainda muito influente no ensino atual:

Na Pedagogia Tradicional, a Didática é uma disciplina normativa, um conjunto de princípios e regras que regulam o ensino. A atividade de ensinar é centrada no professor, que expõe e interpreta a matéria. Às vezes são utilizados meios como a apresentação de objetos, ilustrações, exemplos, mas o meio principal é a palavra, a exposição oral. Supõe-se que ouvindo e fazendo exercícios repetitivos, os alunos “gravem” a matéria para depois reproduzi-la, seja através das interrogações do professor, seja por meio das provas. Para isso, é importante que o aluno “preste atenção”, porque ouvindo facilita-se o registro do que se transmite, na memória. O aluno é, assim, um receptor da matéria e sua tarefa é decorá-la. Os objetivos, explícitos ou implícitos, referem-se à formação de um aluno ideal, desvinculado da sua realidade concreta. O professor tende a encaixar os alunos em um modelo idealizado de homem que nada tem a ver com a vida presente e futura. A matéria de ensino é tratada isoladamente, isto é, desvinculada dos interesses dos alunos e dos problemas reais da sociedade e da vida. O método é dado pela lógica e sequência da matéria, é o meio utilizado pelo professor para comunicar a matéria e não dos alunos para aprendê-la (Libâneo, 1999, p. 67).

Apesar de ainda serem proeminentes, as metodologias de ensino tradicionais possuem efetividade menor a respeito de estudantes mais novos, especialmente os nascidos a partir do século XXI, período no qual ocorreram as principais mudanças referentes à adoção em massa de recursos digitais e da Internet. Segundo (Prensky, 2004), pessoas nascidas com acesso à Internet, referidas por ele de “nativos digitais”, demandariam práticas educacionais distintas das tradicionais. Embora que esse conceito não seja isento de críticas (Bennett; Maton; Kervin, 2008), serve como um útil ponto de partida para compreender as transformações geracionais e a consequente necessidade de se repensarem as práticas educacionais.

Mesmo antes da Internet se popularizar, a crítica perante os paradigmas tradicionais de ensino e o apelo para a modernização das tendências pedagógicas já era proeminente no

meio acadêmico. Nesse contexto, Libâneo (1999) classifica modelos educacionais de diferentes filosofias com o intuito de renovar os paradigmas tradicionais de ensino.

Dentre os modelos mais influentes, estavam presentes os modelos da Escola Nova, que visava tornar o aluno protagonista, valorizando sua liberdade, iniciativa e interesses, promovendo consigo o desenvolvimento de habilidades intelectuais sobre o conteúdo (Libâneo, 1999). Como também o paradigma da pedagogia da libertação, que, partindo de um movimento político e acadêmico oposto, visava focar no diálogo e em “temas geradores” extraídos da realidade social dos alunos, reforçando a importância da conscientização política dos seus indivíduos (Libâneo, 1999).

Conforme as diferentes críticas a respeito dos métodos tradicionais de ensino, a adaptação de paradigmas educacionais em meio às novas demandas de ensino dos alunos mais novos se demonstra essencial. De acordo com as análises feitas por Oliveira (2019), a partir dessas mudanças em nossa sociedade, diferentes ferramentas de ensino digitais emergem como alternativa real e eficaz de ensino a respeito de novas gerações, que já não se adequam totalmente aos métodos tradicionais e difundidos de ensino.

2.2 A importância da Gamificação no ensino

Em meio aos diferentes desafios que emergiram conforme o avanço da pesquisa acadêmica a respeito das novas demandas de ensino para as novas gerações, e consequentemente da adaptação de antigos paradigmas de ensino. Foi amplamente experimentado o uso de elementos relacionados a videogames como uma forma de aprendizado, conceito este conhecido por gamificação.

Vianna et al. (2013) define que gamificação é:

A gamificação (do original em inglês *gamification*) corresponde ao uso de mecanismos de jogos orientados ao objetivo de resolver problemas práticos ou de despertar engajamento entre um público específico (Vianna et al., 2013, p. 13).

A gamificação como parte dos processos de aprendizagem em diversas disciplinas é uma área de estudo bastante ampla. Há muito tempo, pesquisadores têm estudado seus efeitos e características, com o objetivo de criar recursos para, sobretudo, motivar os alunos e demais usuários a se motivarem com as respectivas atividades.

Conforme (Gomes, 2025 apud Nascimento et al., 2025), a gamificação se fundamenta em pressupostos derivados da psicologia comportamental e cognitiva, estando especialmente

associada ao estímulo de respostas motivacionais em ambientes controlados. Nesse sentido, os autores explicam:

A gamificação, como abordagem pedagógica, fundamenta-se em princípios motivacionais derivados da psicologia comportamental e cognitiva. Ao incorporar elementos típicos dos jogos — como pontos, níveis e recompensas — no contexto educativo, pretende-se despertar o interesse dos estudantes e promover o engajamento contínuo com os conteúdos curriculares, respeitando o ritmo de aprendizagem individual (Nascimento et al., 2025, p. 88).

De acordo com Vianna et al. (2013), existem quatro tipos de fatores principais presentes em videogames capazes de fazer com que os usuários se engajem, dentre eles:

- **Meta:** representa o propósito central que orienta o esforço do jogador, fornecendo um senso de direção. Diferente de um objetivo simples, a meta transcende a conclusão de tarefas imediatas, podendo ser um estado de vitória ou a busca contínua pela excelência dentro do ambiente virtual;
- **Regras:** constituem o conjunto de restrições que delimitam as ações possíveis, impondo limites que estimulam a criatividade e o pensamento estratégico. Sua função principal é equilibrar a complexidade do desafio, garantindo que ele não seja nem impossível nem trivial, ao ponto de desmotivar o participante;
- **Sistema de *feedback*:** atua como um mecanismo de comunicação em tempo real, informando ao jogador sua progressão e desempenho em relação à meta e às regras. Esse retorno constante é essencial para manter a motivação, permitindo que o usuário visualize o impacto de suas decisões e o avanço alcançado;
- **Participação voluntária:** estabelece o contrato psicológico entre o sujeito e o jogo, exigindo a aceitação deliberada da meta, das regras e do *feedback*. Este elemento assegura que todos os envolvidos estejam dispostos a interagir dentro das condições propostas, transformando a atividade em uma experiência de jogo legítima.

A eficácia da gamificação no ambiente educacional reside na sua capacidade de estimular a motivação intrínseca dos discentes. Ao integrar elementos como sistemas de recompensas e *feedbacks* imediatos, a ferramenta atende às necessidades psicológicas fundamentais descritas pela Teoria da Autodeterminação (Ryan; Deci, 2017). Além disso, a aplicação prática de

conhecimentos em contextos simulados torna o aprendizado mais significativo e envolvente para o estudante (Hamari; Koivisto; Sarsa, 2014).

Aplicando esses conceitos adequadamente, os jogos ganham um grande potencial na educação. Pode-se observar alguns casos nos quais videogames foram utilizados como instrumentos pedagógicos, com diferentes níveis de sucesso. Borges (2023) emprega o uso de diferentes jogos, como o *Kahoot* e o *Wordwall*, no formato de *quizzes*. Esse experimento teve resultados positivos no que se refere ao estudo de geografia, reforçando o potencial que tais ferramentas possuem nesse domínio.

Tal análise se alinha com o que foi descrito por Breda (2018), que realizou diferentes testes em relação à aplicação de jogos de geografia com crianças. Foi notado um aumento de interesse entre os estudantes, o que obteve resultados positivos. Entre os principais benefícios, foi notado o aumento na motivação e no interesse dos alunos em relação às atividades e aos temas propostos, reforçando, assim, a ideia de que os jogos ou conceitos de jogos podem, sim, ser utilizados efetivamente como ferramenta no auxílio do aprendizado, alinhando-se com os estudos de Prensky (2004).

2.3 O Jogo Geoguessr

O *GeoGuessr* é um jogo de geolocalização que consiste em gerar uma localização aleatória. O jogador deve usar os elementos visuais da imagem apresentada para estimar o lugar onde a fotografia foi capturada. Utiliza-se a *Application Programming Interface* (API) do Google *Street View*, abrangendo imagens em 360 graus, assim como imagens estáticas (Figlerowicz, 2015). Criado em 2013 por Anton Wallen, o jogo ganhou grande popularidade, com a relevância do projeto aumentando principalmente após os anos da pandemia, como se pode observar na Figura 1 apresentada anteriormente na introdução.

Apesar de ter uma proposta simples, foi notado que esse tipo de aplicação contribui muito para o conhecimento geral do usuário, com um potencial pedagógico expressivo. O jogador precisa conectar diversas matérias e conceitos para obter uma pontuação maior e, conseqüentemente, é capaz de integrar diferentes conhecimentos de forma interativa, dinâmica e multidisciplinar (Siqueira; Moreno, 2021).

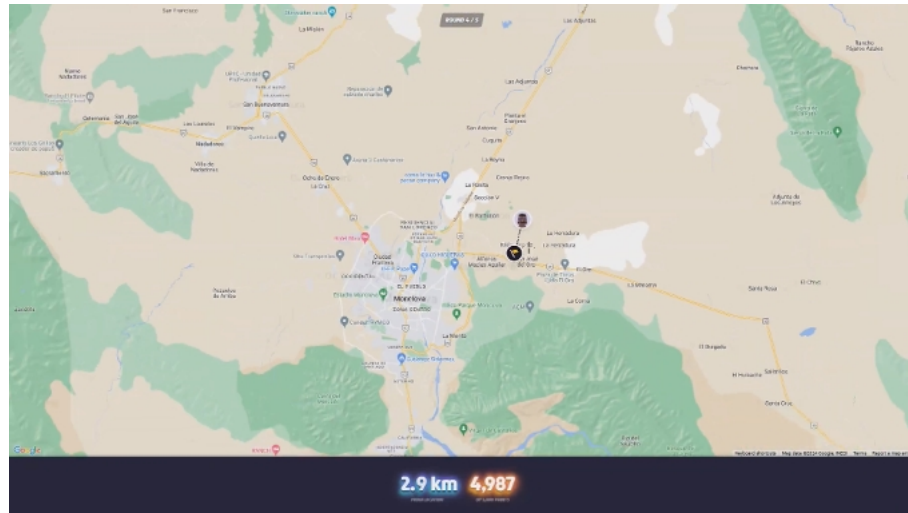
Figura 3 – Informações de uma rodada do GeoGuessr.



Fonte: Captura de tela do jogo *Geoguessr*

Conforme observado na Figura 3, o estágio inicial de uma rodada no *GeoGuessr* coloca o usuário diante de uma projeção panorâmica do Google *Street View*, simulando a presença física em uma localidade desconhecida. Nesse cenário, o processo de pontuação depende diretamente da capacidade analítica do jogador em decodificar elementos da paisagem. O usuário deve analisar a paisagem em busca de evidências geográficas e culturais: a incidência solar e a posição das sombras podem indicar o hemisfério; o tipo de vegetação e a aridez do solo sugerem zonas climáticas; enquanto elementos da infraestrutura, como placas de sinalização, arquitetura das edificações e até o lado da via em que os veículos trafegam, servem como indicadores geopolíticos cruciais. Essa etapa reforça a utilidade do jogo como uma ferramenta de observação científica, na qual o jogador deixa de ser um espectador passivo para se tornar um investigador do espaço geográfico (Figlerowicz, 2015).

Figura 4 – Pontuação adquirida após um palpite no GeoGuessr.



Fonte: Captura de tela do jogo *Geoguessr*

A Figura 4 ilustra a etapa de *feedback* e consolidação do conhecimento após a inserção do palpite no mapa interativo. O sistema calcula a precisão da resposta com base na distância linear entre a coordenada real e a escolhida pelo usuário, convertendo essa métrica em uma pontuação que varia de 0 a 5.000 pontos. Mais do que um simples resultado numérico, essa interface oferece uma lição de cartografia e escala em tempo real: a visualização da discrepância entre os pontos permite que o jogador compreenda erros de interpretação e refine seu repertório espacial para as próximas partidas.

Em suma, dadas as características apresentadas, o *GeoGuessr* atua como uma ferramenta com grande potencial para o campo do ensino, motivando e promovendo uma atitude interativa em relação ao aprendizado e à aplicação de diferentes conhecimentos geográficos, como é amplamente discutido em Sena (2019), Siqueira e Moreno (2021) e Figlerowicz (2015).

2.4 O conceito de paisagem e suas aplicações no ensino de Geografia

Os resultados positivos dos experimentos com jogos educacionais, discutidos nas sessões anteriores e aliados às características inerentes aos jogos de geografia, apontam para o potencial do uso de imagens de paisagens como recurso didático. Dentre esses jogos, destacam-se jogos que se aproveitam do reconhecimento de paisagem.

Conforme (Santos, 2007 apud Oliveira, 2022), o conceito de paisagem é descrito da seguinte maneira:

A dimensão da paisagem é a dimensão da percepção, o que chega aos sentidos. Por isso, o aparelho cognitivo tem importância crucial nessa apreensão, pelo fato de que toda nossa educação, formal ou informal, é feita de forma seletiva: pessoas diferentes apresentam diversas versões do mesmo fato. Por exemplo, coisas que um arquiteto, um artista veem, outros não podem ver ou o fazem de maneira distinta. Isso é válido, também, para profissionais com diferente formação e para o homem comum (Santos, 2007, p.61).

Dessa forma, é possível conceituar que o ensino de Geografia auxiliado por paisagens pode ter grande valor, pois desafia as concepções prévias dos alunos em relação a um espaço geográfico. Ao observar a imagem de uma paisagem, experiências prévias são mobilizadas para atribuir sentido ao que é visto. Um arquiteto, por exemplo, terá uma leitura de um espaço urbano distinta de um agrônomo, que por sua vez também terá outra perspectiva em relação a um estudante de História.

Essa multiplicidade de olhares sobre a mesma paisagem é justamente o que torna jogos que envolvem o reconhecimento de paisagens uma ferramenta pedagogicamente rica. O jogo desafia o aluno a integrar conhecimentos de diferentes áreas: vegetação, clima, arquitetura, infraestrutura e linguagem podem articulados pelo estudante como ferramentas de orientação espacial, enriquecendo o valor pedagógico da aplicação.

Essa ampla variedade de conhecimentos (mobilizados consciente ou inconscientemente para decodificar as informações contidas em uma paisagem) abre caminho para a combinação de elementos de jogos com fins educativos, aproveitando a natureza multidisciplinar desse tipo de exercício. Combinando o potencial que as imagens têm de recrutar informações de múltiplas disciplinas de maneira quase imediata com a capacidade dos videogames educativos, obtêm-se portanto, uma combinação de grande potencial capaz de auxiliar no ensino, sobretudo de Geografia.

Sena (2019), em sua tese de doutorado, realizou um experimento com crianças em seu projeto *GeoMinasCraft*. Utilizando o jogo *Minecraft*, ele representou a paisagem da Serra de Ouro Preto (MG) e demonstrou a história e relevância da região de forma gamificada. Foram relatados resultados positivos quanto ao aprendizado e à motivação dos alunos. Reforçando o amplo potencial em que os jogos possuem para ensinar diferentes conceitos e disciplinas.

A partir dessa combinação de conceitos, jogos similares ao *GeoGuessr* se torna uma ferramenta extremamente interessante, pois exige do aluno um esforço ativo de identificação de padrões para obter melhor pontuação, utilizando diversas disciplinas e conhecimentos sobre paisagens que, previamente, poderiam não estar necessariamente interligados.

Siqueira e Moreno (2021) relata que a interdisciplinaridade atribuída ao *GeoGuessr* contribui para o aprendizado de diferentes conceitos geográficos, descrevendo uma experiência positiva ao empregar o jogo em sala de aula com alunos da rede pública do Estado de São Paulo.

Stefenon (2016), por sua vez, utilizou o *GeoGuessr* em sala de aula e relatou os seguintes resultados:

Os depoimentos dos alunos permitem identificar diferentes possibilidades de temas e saberes da Geografia escolar que podem ser mobilizados pelo jogo. Como se pode abstrair das contribuições destacadas, além de promover formas diferenciadas de uso do mapa para encontrar distâncias e localizações ao redor do globo, o jogo também demonstra potencial para o trabalho tanto com elementos mais ligados à temática da natureza, quanto com assuntos inseridos nos campos dos estudos econômicos e culturais da Geografia. Em outras palavras, o *GeoGuessr* permite abordar as paisagens geográficas a partir de uma compreensão ampla do conceito (Sauer, 1998; Bertrand, 1971), entendendo-a como resultado de diferentes movimentos de interação entre elementos naturais e humanos (Stefenon, 2016, p. 38-39).

Adicionalmente, (Stefenon, 2016) descreve um resultado inusitado do uso do jogo em sala de aula:

O *GeoGuessr* também possibilitou a oportunidade de sonhar com os lugares observados na tela. Aqui, caberia, novamente, uma discussão social, pensando nas condições socioeconômicas da maioria dos alunos da rede pública do Estado de São Paulo, privados, muitas vezes, do lazer e da expansão criativa que uma viagem pode proporcionar. Essa atividade esteve, em alguns momentos, estreitamente relacionada ao Projeto de Vida dos alunos, que associaram a oportunidade de conhecer os países da Europa que apareceram no jogo ao sucesso no trabalho e, consequentemente, na vida financeira. (Stefenon, 2016, p. 11)

O relato de Stefenon (2016) sobre a oportunidade do usuário de poder "sonhar com os lugares" evidencia uma dimensão frequentemente negligenciada no ensino de Geografia: o aspecto afetivo e projetivo da aprendizagem. Ao associar os países observados no *GeoGuessr* ao sucesso profissional e à realização financeira, os alunos da rede pública paulista revelam que o jogo não apenas transmite informações geográficas, mas também mobiliza desejos, imaginários e expectativas de futuro.

Essa constatação reforça o potencial pedagógico do *GeoGuessr* para além da mera localização cartográfica: ele conecta o conhecimento geográfico à vida concreta dos estudantes, tornando a paisagem um elemento vivo e significativo, alinhando-se com os paradigmas pedagógicos de Freire (1996), que preconiza uma educação que parte da 'leitura de mundo' do educando, transformando o saber geográfico em uma ferramenta de conscientização e projeto

de vida. E também com a teoria de aprendizagem significativa de Ausubel (2003), na qual o novo conhecimento (a Geografia) ancora-se em 'subsunçores' afetivos e sociais já presentes na estrutura cognitiva do estudante, tornando a informação pessoalmente relevante.

Ao longo desta seção, partiu-se do conceito de paisagem formulado por Milton Santos, entendido como produto da percepção e das experiências prévias de cada sujeito, para demonstrar como imagens geográficas podem desafiar concepções consolidadas e estimular o pensamento multidisciplinar. Os experimentos de Sena (2019) com o *GeoMinasCraft* e de Borges (2023) com quizzes gamificados indicaram resultados positivos em termos de motivação e aprendizado. Já os trabalhos de Siqueira e Moreno (2021) e Stefenon (2016) aplicaram diretamente o *GeoGuessr* em contextos escolares, identificando ganhos na interdisciplinaridade, no engajamento e, como visto, na dimensão afetiva e projetiva dos alunos.

2.5 O Impacto da Análise de Dados no Desempenho do Usuário

Entretanto, apesar de o *GeoGuessr* e seus desenvolvedores terem excelentes conceitos e aplicações sobre como auxiliar o ensino de Geografia com jogos, podemos observar algumas maneiras muito claras de melhorar ainda mais a sua experiência de usuário adicionando algumas funcionalidades. Até a data de confecção deste documento, as plataformas comerciais consolidadas desse nicho carecem de ferramentas nativas estruturadas para coletar métricas profundas de desempenho do jogador e exibi-las de forma analítica e centralizada para ele.

A simples coleta de dados, entretanto, não garante melhorias no desempenho. Para que tais informações sejam úteis ao usuário, torna-se necessário transformá-las em representações compreensíveis e acionáveis. É nesse ponto que a visualização de dados assume papel central. Segundo revisão sistemática conduzida por Vieira, Parsons e Byrd (2018), as ferramentas de análise visual de aprendizagem auxiliam estudantes e educadores a interpretar grandes volumes de informação, identificar padrões de comportamento e reconhecer dificuldades de aprendizagem de maneira mais rápida e intuitiva.

Painéis de desempenho (*dashboards*) têm sido amplamente utilizados em ambientes educacionais justamente por sintetizar indicadores relevantes em interfaces visuais (Rets et al., 2021). Esses sistemas agregam evidências em tempo real para fornecer um "instantâneo" progresso do aluno por meio de uma ou mais visualizações. Essa abordagem permite acompanhar o progresso, a participação (como cliques no ambiente virtual) e o tempo de atividade, o que favorece processos de autorregulação da aprendizagem, permitindo que os estudantes monitorem e

ajustem seu comportamento para atingir metas.

Além disso, a visualização de dados facilita a identificação de falhas específicas que poderiam passar despercebidas em avaliações tradicionais. Em vez de apenas informar um resultado final, sistemas analíticos podem revelar, por exemplo, tópicos com maior índice de erro, padrões recorrentes de dificuldade ou queda de desempenho em determinadas etapas. Pesquisa recente sobre a plataforma *LearningViz* demonstrou que representações multidimensionais de desempenho permitem detectar lacunas de aprendizagem, avaliar dificuldades por tópico e orientar intervenções mais precisas (Pei et al., 2024).

Esse princípio possui aplicação direta em jogos educacionais e plataformas gamificadas. Quando combinadas com mecânicas de jogo, métricas de desempenho tornam-se também instrumentos motivacionais. Em estudo sobre um *dashboard* gamificado aplicado ao ensino superior, observou-se que a integração entre visualização analítica e elementos lúdicos contribuiu para maior engajamento e melhores experiências de aprendizagem entre os participantes (Alam et al., 2023).

No contexto desta pesquisa, tais fundamentos justificam a implementação de um sistema inspirado no *GeoGuessr* que incorpora análise estatística individual por mapas e sessões de jogo. Diferentemente de experiências baseadas apenas em pontuação final, a proposta desenvolvida permite ao usuário visualizar em quais regiões apresenta melhor desempenho, onde comete mais erros e como evolui ao longo do tempo. Dessa forma, o jogo deixa de ser apenas recreativo e passa a atuar como ferramenta diagnóstica e formativa, estimulando aprendizagem contínua em Geografia por meio da reflexão sobre os próprios resultados.

É diante desse cenário que se estabelece a principal motivação para a concepção do *Geolacta*. A aplicação surge não apenas como um clone recreativo das mecânicas de adivinhação geográfica existentes, mas sim como uma resposta técnica e pedagógica à falta de devolutivas estatísticas encontradas no mercado atual. Ao acoplar um módulo analítico espacial ao núcleo do jogo, o projeto busca mitigar a desconexão entre a jogabilidade estocástica (baseada em tentativas isoladas) e a retenção cognitiva de longo prazo, transformando cada palpite errado em um dado visual permanente que pode ser estudado e superado pelo usuário.

Dessa forma, mesmo configurando-se como um protótipo inicial e incremental, o *Geolacta* diferencia-se substancialmente de seus congêneres ao deslocar o foco do placar efêmero para a construção de um histórico georreferenciado de aprendizagem. A implementação de um componente de visualização gráfica dedicado a comparar as distribuições espaciais de

erros e acertos valida o sistema como uma ferramenta viável de autoavaliação. Essa capacidade de revelar padrões comportamentais e lacunas geográficas de forma visual assegura ao software um papel genuinamente formativo, preenchendo a lacuna de engenharia e usabilidade apontada pela literatura acadêmica.

2.6 Tecnologias utilizadas

O desenvolvimento do *Geolacta* demandou a utilização de um conjunto de tecnologias modernas e bem consolidadas no mercado. A escolha de cada ferramenta foi orientada pela necessidade de garantir robustez, portabilidade e eficiência ao sistema, desde o versionamento do código até a sua execução em contêineres.

2.6.1 Git e GitHub

O *Git* consolidou-se como o padrão de referência para sistemas de controle de versão distribuídos, assegurando a integridade e a rastreabilidade do histórico de alterações do código-fonte. Baseado nos conceitos apresentados por Chacon e Straub (2014), a ferramenta desempenha papel fundamental na segmentação de funcionalidades e na mitigação de conflitos durante o ciclo de desenvolvimento. Complementarmente, a plataforma *GitHub* atua como o repositório remoto estratégico, provendo não apenas a hospedagem segura do projeto, mas também mecanismos de integração contínua e gestão organizada das metas de desenvolvimento.

2.6.2 Docker

A adoção do *Docker* demonstrou-se indispensável para garantir a paridade entre os ambientes de desenvolvimento e produção. Por meio da tecnologia de containerização, foi possível isolar a aplicação e suas dependências em unidades leves e portáteis. Essa abordagem mitigou inconsistências operacionais derivadas de diferentes sistemas hospedeiros, eliminando a necessidade de configurações manuais complexas. Ademais, o *Docker* configurou-se como a solução de containerização predominante no mercado global, representando um pilar fundamental nas práticas de *DevOps*. O domínio desta ferramenta assegurou a consistência técnica do projeto e alinhou o desenvolvimento aos padrões contemporâneos da indústria, onde a escalabilidade e a agilidade na implantação via contêineres constituem requisitos mandatórios (Docker, Inc., 2025).

2.6.3 Python

Python foi a linguagem escolhida para moldar a lógica do servidor, sendo responsável pelo desenvolvimento dos algoritmos de seleção aleatória de localizações, pela criação das classes de domínio (salas, mapas, usuários) e pela orquestração geral do sistema. A linguagem é amplamente reconhecida por sua clareza e produtividade, características que justificaram sua utilização neste projeto. Além disso, sua riqueza em bibliotecas e *frameworks* atendeu perfeitamente às necessidades do escopo atual. Desse ecossistema, as bibliotecas mais relevantes empregadas foram:

- **Flask:** *microframework* utilizado como base para construção do *backend*. Sua abordagem minimalista permite que o desenvolvedor escolha apenas as extensões desejadas, resultando em uma aplicação mais enxuta que outros *frameworks* do mercado (Grinberg, 2018). Nesta aplicação, o Flask foi responsável pela criação das rotas HTTP (*Hypertext Transfer Protocol*), pelo gerenciamento de requisições e pela integração entre os componentes do sistema;
- **SQLAlchemy (com Flask-SQLAlchemy):** ORM (*Object-Relational Mapping*) empregado para abstrair as operações com o banco de dados, permitindo a manipulação de dados por meio de objetos Python (SQLAlchemy Authors and Contributors, 2026). As ferramentas **Flask-Migrate** e **Alembic** foram utilizadas para gerenciar as migrações de esquema, garantindo versionamento e rastreabilidade das alterações na estrutura de dados;
- **Flask-Login:** extensão responsável pelo gerenciamento de sessões de usuário, implementando o fluxo de autenticação e controle de acesso;
- **WTForms (com Flask-WTF):** biblioteca utilizada para criação e validação de formulários web, garantindo a segurança e a integridade dos dados enviados pelos usuários (WTForms Team, 2026);
- **GeoAlchemy2 e Shapely:** bibliotecas empregadas para suporte a dados geoespaciais, estendendo o SQLAlchemy para trabalhar com tipos geográficos (GeoAlchemy Contributors, 2026) e oferecendo operações geométricas essenciais por meio do Shapely;

- **Jinja2**: mecanismo de templates utilizado pelo Flask para renderizar as páginas HTML (*Hypertext Markup Language*) de forma dinâmica, combinando marcadores especiais com dados fornecidos pelo backend (Pallets Projects, 2024);
- **Requests**: biblioteca para requisições HTTP a APIs (*Application Programming Interface*) externas, empregada nas comunicações com a *Google Street View Static API* para obtenção das imagens panorâmicas;
- **Bibliotecas auxiliares**: complementam o funcionamento do sistema com finalidades específicas: **Python-dotenv** para gerenciamento de variáveis de ambiente, **Email-validator** para validação de endereços de e-mail no cadastro de usuários e **Pillow** para operações básicas de processamento de imagens.

2.6.4 HTMX

O *HTMX* consolidou-se como uma abordagem diferenciada para o desenvolvimento de interfaces *web* interativas. Em contraste com os tradicionais e mais pesados *frameworks* JavaScript, como React, Vue ou Angular, o *HTMX* propõe uma arquitetura minimalista que estende as capacidades do próprio *HTML*. Por meio de atributos específicos, como `hx-get` ou `hx-post`, é possível realizar requisições assíncronas ao servidor e atualizar partes da página de forma dinâmica, reduzindo drasticamente a necessidade de escrever JavaScript customizado (Big Sky Software, 2026).

Essa abordagem revelou-se particularmente adequada à natureza do *Geolacta*: um jogo de estilo *GeoGuessr* que exige interações ágeis e leves como o envio de palpites, o carregamento de novos mapas e a atualização de placares, mas sem a complexidade de uma *single-page application*. Com o *HTMX*, foi possível manter a lógica de negócio majoritariamente no servidor, no ambiente *Flask*, utilizando o *front-end* apenas como uma camada de apresentação responsiva. As principais vantagens observadas no uso do *HTMX* foram:

- **Simplicidade de desenvolvimento**: a interatividade é declarada diretamente no *HTML*, o que reduz significativamente a quantidade de código JavaScript escrito manualmente e facilita a manutenção do sistema;
- **Integração nativa com Flask**: uma vez que o *HTMX* consome *endpoints* que retornam fragmentos *HTML*, ele se alinha perfeitamente com a arquitetura de templates *Jinja2*, dispensando a construção de APIs dedicadas para cada interação;

- **Experiência de usuário fluida:** permite atualizações parciais da interface, como carregar um novo mapa ou exibir o resultado de um palpite, sem a necessidade de recarregar a página inteira;
- **Redução de complexidade:** evita a sobrecarga de gerenciar estado no cliente, transferindo essa responsabilidade para o servidor, onde o modelo de domínio do jogo já está definido.

Desta forma, o *HTMX* contribuiu para que o desenvolvimento da interface se mantivesse ágil e alinhado com o restante da arquitetura, permitindo que o foco permanecesse na lógica do jogo e nas integrações geoespaciais, sem sacrificar a qualidade da interação com o usuário.

2.6.5 Tailwind CSS

O *Tailwind CSS (Cascading Style Sheets)* é um *framework CSS* utilitário que permite a construção de interfaces diretamente nos templates *HTML*, por meio de classes de baixo nível (*utility-first*). Diferentemente de abordagens tradicionais que exigem folhas de estilo extensas, o *Tailwind* oferece classes como `bg-white`, `rounded-lg`, `shadow-lg` e `grid-cols-1`, que foram utilizadas para definir o layout responsivo do jogo. Essa estratégia acelerou o desenvolvimento e garantiu consistência visual em diferentes resoluções de tela (*desktop*, *tablet* e *celular*), sem a necessidade de escrever *CSS* customizado em larga escala (Tailwind Labs, 2025).

2.6.6 D3.js

D3.js é uma biblioteca JavaScript amplamente utilizada para manipulação de documentos baseada em dados. Ela permite vincular dados arbitrários a elementos do *DOM* (Document Object Model) e aplicar transformações orientadas por dados, facilitando a criação de visualizações interativas, dinâmicas e personalizáveis para a web (D3.js Contributors, 2025).

A biblioteca *D3.js* foi empregada para criar visualizações dinâmicas e interativas dos dados de desempenho do usuário. Especificamente, implementaram-se dois componentes principais:

- **Gráfico de dispersão sequencial (*scatter plot*):** exibe a evolução da pontuação ao longo das partidas, com eixos representando o número da tentativa e a pontuação obtida (de 0 a 10.000 pontos). Cada ponto exibe um *tooltip* com informações adicionais (data, distância), permitindo ao jogador identificar tendências de melhora ou piora;

- **Mapa de calor de atividade (*heatmap* diário):** representa, em uma grade de 90 dias, a quantidade de palpites realizados por dia. A intensidade da cor azul indica maior frequência de atividade, fornecendo uma visão rápida dos períodos de maior engajamento.

Ambos os gráficos foram construídos com a versão 7 da biblioteca, aproveitando sua capacidade de vincular dados diretamente a elementos do *DOM* e de calcular escalas, eixos e cores de forma programática (D3.js Contributors, 2025).

2.6.7 Leaflet

Para a exibição de mapas interativos no *Geolacta*, adotou-se a biblioteca *Leaflet*, reconhecida por sua leveza e facilidade de integração em aplicações *web*. Um mapa na *web* é geralmente composto por um conjunto de imagens quadradas (*tiles*) organizadas em grades, formando um *tileset*, cada *tile* corresponde a uma porção do espaço geográfico em um dado nível de zoom. O *Leaflet* gerencia o carregamento e a disposição desses *tiles* de forma eficiente, permitindo ao desenvolvedor concentrar-se na lógica cartográfica.

No *Geolacta*, o *Leaflet* assumiu duas funções principais:

- **Representação de estatísticas espaciais:** na página de desempenho do usuário, a biblioteca renderiza um mapa-múndi onde cada palpite realizado aparece como um marcador circular. A cor do marcador (variando do vermelho para baixa pontuação ao azul para alta pontuação) permite ao jogador visualizar rapidamente as regiões onde obteve melhor ou pior desempenho. Ao interagir com cada marcador, um *tooltip* exibe detalhes como pontuação, distância ao local real e data da tentativa;
- **Interface ativa do jogo:** durante a partida, o *Leaflet* é empregado para apresentar o mapa no qual o usuário deve clicar para dar seu palpite. A biblioteca captura as coordenadas geográficas (latitude e longitude) do clique, alimentando o mecanismo de cálculo de distância e pontuação. Essa integração direta com a lógica do jogo dispensa a construção de componentes cartográficos externos e mantém a fluidez da experiência.

A escolha do *Leaflet* justificou-se, portanto, por sua capacidade de conciliar simplicidade de implementação com recursos essenciais (camadas de *tiles*, marcadores, *tooltips* e captura de eventos de clique), atendendo tanto à necessidade de visualização analítica quanto à interatividade exigida pela jogabilidade (Leaflet Contributors, 2025).

2.6.8 GeoJSON Maps e Natural Earth

Para a obtenção das geometrias cartográficas utilizadas na renderização dos continentes e países presentes no *Geolacta*, empregou-se a plataforma *GeoJSON Maps*. O serviço disponibiliza uma interface simplificada para geração e exportação de mapas vetoriais no formato *Geographic JavaScript Object Notation (GeoJSON)*, permitindo selecionar regiões específicas do globo e diferentes níveis de resolução cartográfica (KYD, 2026);

Os arquivos disponibilizados pela plataforma são derivados do projeto *Natural Earth*, uma base de dados cartográficos amplamente utilizada em aplicações geoespaciais e visualizações científicas. O *Natural Earth* fornece dados vetoriais em domínio público, abrangendo fronteiras políticas, massas continentais, hidrografia e demais elementos geográficos globais (Natural Earth, 2026).

No contexto do *Geolacta*, os arquivos *GeoJSON* foram empregados principalmente para:

- **Renderização de limites geográficos:** os polígonos vetoriais foram utilizados para desenhar continentes e países em mapas interativos exibidos na interface da aplicação;
- **Redução de processamento:** a possibilidade de selecionar apenas regiões específicas permitiu diminuir significativamente o tamanho dos arquivos transferidos ao navegador, reduzindo o consumo de banda e o tempo de carregamento das páginas;
- **Compatibilidade com bibliotecas cartográficas:** o formato *GeoJSON* integra-se diretamente com bibliotecas como o *Leaflet* e o *D3.js*, possibilitando manipulação dinâmica dos elementos geográficos através de *JavaScript*.

A adoção dessa solução justificou-se pela simplicidade de integração com aplicações *web* modernas, pela leveza dos arquivos gerados e pela disponibilidade de dados cartográficos abertos e em domínio público, adequados ao desenvolvimento de aplicações educacionais e experimentais (KYD, 2026; Natural Earth, 2026).

2.6.9 Google Maps Platform

A *Google Maps Platform* fornece um robusto conjunto de *API's*, consolidando-se como a infraestrutura geoespacial de referência para sistemas que exigem alta precisão cartográfica. No escopo deste projeto, a utilização desse ecossistema justificou-se pela necessidade de

renderização de ambientes imersivos em tempo real e pela integração de dados geográficos dinâmicos (Google for Developers, 2026b).

- ***Street View Static API e Maps JavaScript API***: a renderização das imagens panorâmicas em 360 graus ocorreu por meio do serviço *Street View*, que permite a exploração visual de localidades mapeadas globalmente. A interação do usuário com o mapa e a captura de coordenadas geográficas (latitude e longitude) foram processadas pela *Maps JavaScript API*. Essa ferramenta possibilitou a manipulação programática do *DOM* para a inserção de marcadores, o cálculo de trajetórias e a personalização de eventos de clique, elementos fundamentais para a mecânica do jogo proposta (Google for Developers, 2026c);
- **Metadados do *Street View* e processamento de imagens**: a integração com o serviço de metadados da *Street View Static API* foi empregada para verificar a disponibilidade de imagens panorâmicas antes de sua requisição, garantindo a fluidez da experiência do usuário. O sistema utilizou requisições seguindo o *HTTPS (Hypertext Transfer Protocol Secure)* padronizadas, seguindo as melhores práticas de segurança e tratamento de erros, conforme documentação oficial do Google para a plataforma (Google for Developers, 2026d; Google for Developers, 2026a).

A escolha por essa tecnologia fundamentou-se, por fim, na escalabilidade da infraestrutura do Google, que garante tempos de resposta reduzidos e uma base de dados cartográficos constantemente atualizada, fatores críticos para a experiência do usuário final em aplicações de geolocalização.

3 METODOLOGIA DE DESENVOLVIMENTO

O desenvolvimento do sistema e todas as suas funcionalidades seguiu as etapas de desenvolvimento estudadas na engenharia de software. Segundo Sommerville (2016, p. 44), o ciclo de desenvolvimento de sistemas tem 4 etapas iniciais, sendo elas:

- Especificação de software: etapa que consiste na definição do escopo do projeto, assim como na elicitação de seus requisitos funcionais e não funcionais. Nessa etapa também é decidido o *design* e as respectivas arquiteturas que o sistema deve utilizar;
- Desenvolvimento de software: etapa na qual todos os *designs* e protótipos feitos durante a especificação devem ser realizados;
- Validação de software: aplicação de testes realizados para conferir e garantir que a aplicação está ocorrendo conforme os requisitos;
- Evolução de software: etapa reservada para a introdução de novas funcionalidades e aprimoramento de funções anteriores.

3.1 Especificação de software

Por meio de uma reflexão sobre os tópicos abordados na fundamentação teórica, realizou-se um levantamento informal de funcionalidades junto a um pequeno grupo de jogadores conhecidos do autor, todos com experiência prévia no *GeoGuessr*. O grupo foi composto por três indivíduos, com idades entre 20 e 27 anos, que relataram já ter jogado o *GeoGuessr* com frequência variada.

A abordagem consistiu em perguntas abertas sobre quais recursos consideravam úteis ou ausentes no jogo original, bem como sugestões para uma plataforma educacional semelhante. As respostas obtidas foram consolidadas e serviram como insumos iniciais para a definição dos requisitos funcionais, posteriormente priorizados conforme o escopo do projeto.

O levantamento de requisitos é etapa fundamental do desenvolvimento de software, pois ajuda a definir precisamente o escopo de aplicação do sistema, a respeito disso Pressman e Maxim (2020) Definem:

Engenharia de requisitos é o termo que engloba o amplo espectro de tarefas e técnicas que levam à compreensão dos requisitos. Do ponto de vista do processo de software, a engenharia de requisitos é uma ação fundamental

que se inicia durante a comunicação e continua na fase de modelagem. Ela estabelece uma base sólida para o projeto e a construção. Sem ela, o software resultante tem uma alta probabilidade de não atender às necessidades do cliente. A engenharia de requisitos deve ser adaptada às necessidades do processo, do projeto, do produto e das pessoas envolvidas. É importante ressaltar que cada uma dessas tarefas é realizada de forma iterativa, à medida que a equipe do projeto e as partes interessadas compartilham informações sobre suas respectivas preocupações. (Pressman; Maxim, 2020, p. 103).

3.1.1 Requisitos funcionais

Quanto à divisão dos requisitos, é comumente aceita a separação entre dois tipos de requisitos que devem ser coletados pela equipe de desenvolvimento na maioria dos projetos. Sendo eles os requisitos funcionais e os requisitos não funcionais (Sommerville, 2016):

Requisitos funcionais são declarações dos serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em situações específicas. Em alguns casos, os requisitos funcionais também podem declarar explicitamente o que o sistema não deve fazer. (Sommerville, 2016, p. 105).

Com base nesse conceito a respeito dos requisitos funcionais, foram conceptualizados os seguintes requisitos:

Tabela 1 – Requisitos funcionais do sistema

Código	Descrição	Classificação
RF - 1	Desenvolver um sistema de login e autenticação para o registro de jogadores através de e-mail e senha.	Obrigatório
RF - 2	Desenvolver um sistema de geração de localizações pseudo-aleatório.	Obrigatório
RF - 3	Conectar as APIs corretas para a escolha e visualização de fotos no mapa.	Obrigatório
RF - 4	Criar diferentes tipos de mapa para diversificar o jogo.	Obrigatório
RF - 5	Criar uma página para a seleção de diferentes mapas.	Obrigatório
RF - 6	Criar uma página que suporte uma sala para demonstrar informações sobre o mapa a ser jogado.	Obrigatório
RF - 7	Criar uma identidade visual para o jogo (logotipos, <i>banners</i> , paleta de cores).	Desejável
RF - 8	Adicionar uma opção para a customização de perfil do usuário.	Desejável
RF - 9	Criar um sistema de pontuação e de soma de resultados.	Obrigatório
RF - 10	Implementar um serviço de coleta de dados de estatísticas e pontuações do usuário.	Obrigatório
RF - 11	Implementar um serviço de exibição do histórico de estatísticas para o usuário em cada mapa.	Obrigatório
RF - 12	Implementar um modo multijogador competitivo com placares em tempo real.	Fora de escopo
RF - 13	Criar um sistema dinâmico e interativo de criação de mapas.	Fora de escopo
RF - 14	Criar um sistema de filtro de localizações com características similares.	Fora de escopo

Fonte: Elaborado pelo autor (2026).

Observando a lista dos requisitos, percebe-se que eles possuem uma hierarquia entre si, cuidadosamente escolhida para que nenhum requisito atrapalhe o outro em seu desenvolvimento. Essa observação pode ser confirmada uma vez que os objetivos dependem uns dos outros para serem realizados. É inviável, por exemplo, desenvolver o sistema de pontuação sem antes ter ao menos o sistema necessário para que o jogador visualize as imagens e interaja com o mapa da forma que desejar.

A escolha dos requisitos também levou em conta os recursos possíveis de serem alocados para o desenvolvimento do projeto. Por mais que a adição de ferramentas de competição entre diferentes jogadores em tempo real possa parecer atraente para a confecção do jogo, as dificuldades relativas à implementação de tal sistema fazem com que esse requisito se desvie do tema central do projeto e de seu Mínimo produto viável (MPV), uma vez que a criação de um sistema capaz de comportar um jogador em alguns mapas, com os demais requisitos obrigatórios, é mais que o suficiente para a criação de uma experiência de jogo agradável.

3.1.2 Requisitos não funcionais

A respeito dos requisitos não funcionais, (Sommerville, 2016) declara:

São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de tempo, restrições ao processo de desenvolvimento e restrições impostas por normas. Os requisitos não funcionais geralmente se aplicam ao sistema como um todo, em vez de a recursos ou serviços individuais do sistema. (Sommerville, 2016, p. 105).

Seguindo essa definição, os requisitos não funcionais do sistema foram coletados por meio da observação e conceituação de um sistema funcional, os quais envolvem:

Tabela 2 – Requisitos não funcionais do sistema

Código	Descrição	Classificação
RNF - 01	O sistema deve garantir que apenas usuários autenticados possam acessar suas contas e dados, utilizando mecanismos seguros como <i>hash</i> de senhas.	Obrigatório
RNF - 02	A interface deve ser autoexplicativa. Um novo jogador deve ser capaz de iniciar uma partida e entender como dar um palpite em menos de um minuto.	Obrigatório
RNF - 03	O carregamento inicial do mapa e das imagens do <i>Street View</i> deve ser otimizado para ocorrer em poucos segundos.	Desejável
RNF - 04	O <i>front-end</i> deve ser modularizado para que seja mais fácil futuras adições ou manutenções.	Desejável
RNF - 05	O sistema deve ser eficiente na alocação de recursos para se preparar para um possível pico de jogadores.	Desejável

Fonte: Elaborado pelo autor (2026).

3.2 Desenvolvimento de software

3.2.1 Padrão de desenvolvimento

O padrão de desenvolvimento adotado na confecção do *Geolacta* foi uma variação do modelo de desenvolvimento incremental com alguns elementos de *Feature Driven Design* (FDD).

O desenvolvimento incremental baseia-se na ideia de desenvolver uma implementação inicial, obter *feedback* de usuários e outras partes interessadas e evoluir o software por meio de diversas versões até que o sistema desejado seja desenvolvido. As atividades de especificação, desenvolvimento e validação são intercaladas, em vez de separadas, com *feedback* rápido entre as atividades (Sommerville, 2016, p. 49).

Essa metodologia permitiu o desenvolvimento de diversas funcionalidades em uma ordem coesa, isolando o desenvolvimento de cada função do sistema e, em vez disso, permitindo um

maior foco e rapidez de desenvolvimento, mesmo com poucos recursos. Tornou-se viável a criação de uma versão mais simples e funcional, priorizando funcionalidades essenciais para o sistema inicialmente. A respeito dessa metodologia de desenvolvimento, (Sommerville, 2016) ainda confirma:

O desenvolvimento incremental de software, que é parte fundamental dos métodos ágeis de desenvolvimento, é mais adequado do que a abordagem em cascata para sistemas cujos requisitos provavelmente mudarão durante o processo de desenvolvimento. Este é o caso da maioria dos sistemas empresariais e produtos de software. O desenvolvimento incremental reflete a maneira como resolvemos problemas. Raramente elaboramos uma solução completa para um problema antecipadamente, mas avançamos em direção a uma solução em uma série de etapas, retrocedendo quando percebemos que cometemos um erro. Ao desenvolver o software incrementalmente, torna-se mais barato e fácil fazer alterações no software à medida que ele está sendo desenvolvido (Sommerville, 2016, p. 50).

3.2.2 Ordem de implementação dos requisitos

Conforme mencionado na seção de validação de software, a ordem de desenvolvimento das funcionalidades foi planejada para garantir que cada etapa criasse as bases necessárias para a seguinte. Dessa forma, optou-se por implementar uma funcionalidade por vez, sempre considerando sua relevância para as próximas entregas.

A sequência de implementação adotada foi a seguinte:

1. Configuração do banco de dados *PostgreSQL* com extensão *PostGIS* para suporte a dados geográficos;
2. Desenvolvimento das funcionalidades básicas do *back-end*, incluindo o sistema de autenticação;
3. Implementação dos componentes do *front-end* utilizando *HTMX*;
4. Integração com as APIs do *Google Street View* e do *Leaflet*;
5. Criação do algoritmo de geração de localizações aleatórias;
6. Adição de diferentes áreas geográficas para compor distintos modos de jogo;
7. Implementação do carregamento de arquivos *GeoJSON*;
8. Desenvolvimento de um mapa visual para exibir os palpites do jogador em um determinado mapa ou modo de jogo.

3.3 Validação de software

A fase de validação, conforme definida por Sommerville (2016, p. 44), tem como propósito fundamental verificar se o software em desenvolvimento está em conformidade com o que o cliente deseja. No contexto deste projeto, a validação foi conduzida de maneira contínua e predominantemente informal, alinhando-se às práticas comuns em metodologias ágeis e ao modelo de desenvolvimento incremental adotado.

O processo de validação consistiu, inicialmente, na execução de testes exploratórios, nos quais o desenvolvedor atuava simultaneamente como testador, buscando identificar *bugs* visuais, travamentos e anomalias no comportamento do sistema. Conforme destaca Kaner (2008), o teste exploratório é um estilo de teste de software que enfatiza a liberdade pessoal e a responsabilidade do testador, permitindo que a criação e a execução dos testes ocorram de forma simultânea e adaptativa. Essa abordagem mostrou-se particularmente adequada ao contexto de desenvolvimento individual, onde a agilidade e a capacidade de resposta imediata aos problemas são mais relevantes do que a formalização de planos de teste estruturados.

Adicionalmente, a validação foi complementada pela realização de testes de integração não sistemáticos, voltados a identificar conflitos entre funcionalidades desenvolvidas em momentos distintos do projeto. Tais conflitos, manifestados como lentidão ou falhas na geração de localizações aleatórias, foram tratados de forma corretiva imediata, sempre que identificados.

Por fim, reconhecendo a importância da perspectiva do usuário final no processo de validação, buscou-se, em momentos pontuais, o *feedback* de terceiros para avaliar a qualidade da experiência de usuário proporcionada pelo sistema. Como desdobramento futuro, pretende-se estruturar essa prática por meio da aplicação de um formulário de avaliação objetiva, permitindo a coleta sistemática de opiniões de diferentes usuários e a identificação de oportunidades de melhoria.

3.4 Evolução de software

A evolução do software, compreendida como a atividade de modificar o sistema para atender a novas demandas ou corrigir defeitos ao longo do tempo (Sommerville, 2016), esteve intrinsecamente associada ao processo de desenvolvimento adotado. A natureza incremental do projeto, na qual novas funcionalidades foram adicionadas sucessivamente ao sistema, fez com que as atividades de evolução ocorressem de forma praticamente contínua.

Cada novo incremento introduzido ao sistema representava uma evolução em relação à versão anterior, incorporando novas capacidades e, eventualmente, demandando ajustes nas funcionalidades previamente implementadas. Essa dinâmica é característica de sistemas em desenvolvimento ativo, nos quais a linha entre a construção inicial e a evolução contínua é frequentemente tênue, especialmente em projetos conduzidos por um único desenvolvedor.

A abordagem adotada, ainda que não tenha seguido um processo formal de gestão de mudanças, permitiu que o sistema evoluísse de maneira orgânica, respondendo rapidamente às necessidades identificadas ao longo do desenvolvimento. A flexibilidade proporcionada pela ausência de prazos fixos contribuiu para que as decisões sobre quais evoluções implementar fossem orientadas prioritariamente pelas dependências técnicas entre as funcionalidades, garantindo a consistência e a integridade do sistema.

4 PROJETO

A arquitetura de software adotada no desenvolvimento do *Geolacta* foi descrita neste capítulo, com ênfase na organização em camadas baseada nos padrões Modelo Visão Controle e Camada de Serviços. Foram apresentados o modelo de dados, o fluxo de controle da principal interação do jogo (submissão de palpite) e a estratégia de containerização utilizada para garantir reprodutibilidade e isolamento do ambiente.

Na sequência, o capítulo abordou os algoritmos geoespaciais desenvolvidos para o funcionamento da plataforma, incluindo o carregamento de fronteiras via grade de centróides, a geração hierárquica de localizações aleatórias com validação de imagens e o cálculo de distância geográfica baseado na fórmula de Haversine, integrado ao sistema de pontuação.

4.1 Visão Geral da Arquitetura

Na literatura de engenharia de software, o termo arquitetura abrange desde padrões de alto nível até estruturas modulares internas. De acordo com Pressman e Maxim (2021), a arquitetura de software de um programa ou sistema é a estrutura ou estruturas do sistema que compreendem os componentes de software, as propriedades externamente visíveis desses componentes e os relacionamentos entre eles.

Neste trabalho, a descrição das pastas, a separação em camadas e o fluxo de controle constituem a visão arquitetural. O padrão Modelo Visualização e Controle (*MVC*) com camada de serviços é, portanto, uma instância concreta dessa definição, sendo amplamente aceito em projetos acadêmicos e comerciais. Autores como Buschmann et al. (1996) denominam essa organização como *Layers pattern* (padrão de camadas), que por sua vez pode conter o padrão *MVC* em uma de suas camadas.

A arquitetura adotada no *Geolacta* fundamenta-se nos padrões (*MVC*) e *Service Layer*, adaptados ao contexto de desenvolvimento web com o *microframework Flask*. Esta seção descreve os fundamentos teóricos, a organização estrutural do código-fonte, o fluxo de requisições e a justificativa das decisões arquiteturais, correlacionando conceitos abstratos à implementação concreta nos diretórios e arquivos do projeto.

4.1.1 MVC e a Camada de Serviços

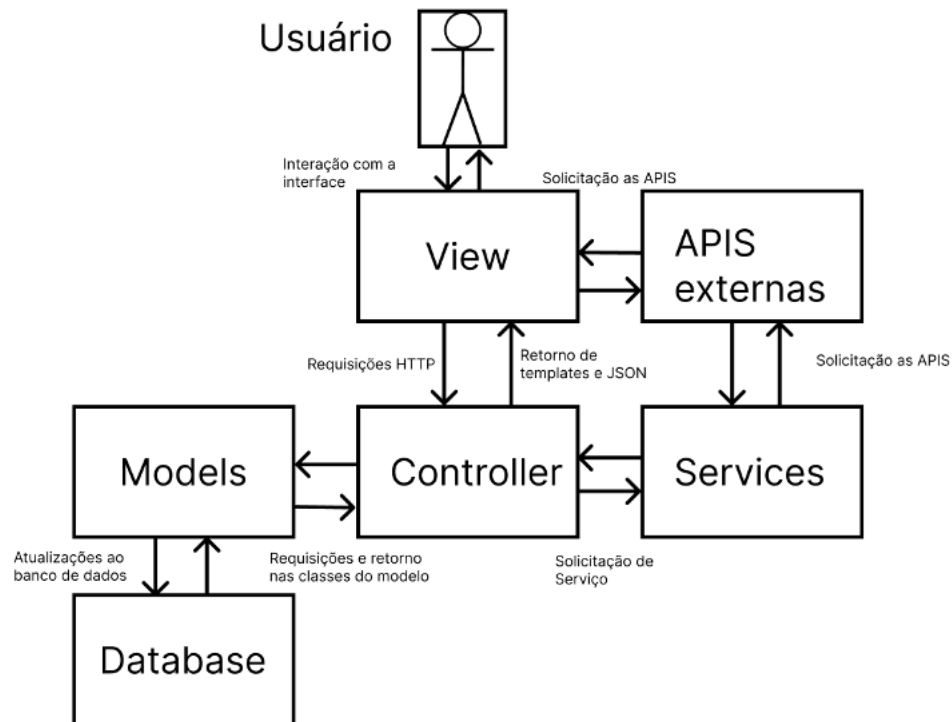
O padrão arquitetural *MVC*, descrito por (Pressman; Maxim, 2021), propõe a separação de um sistema interativo em três componentes principais: o **Model** (dados e regras de negócio), a **View** (apresentação e interação) e o **Controller** (orquestração das requisições). Em aplicações web, essa separação traz benefícios como manutenibilidade, reutilização e paralelismo no desenvolvimento.

No *Flask*, as funções associadas a rotas (decorador `@app.route`) atuam como a camada de controle: recebem a requisição, invocam serviços e retornam a resposta (renderização de *template* ou *JSON*). Os *templates Jinja2* constituem a camada de apresentação. No entanto, para sistemas que vão além de operações triviais de *CRUD* (*Create, Read, Update, Delete*), a simples tríade *MVC* tende a concentrar lógica de negócio nos controladores, fenômeno conhecido como *fat controller*. Para mitigar esse problema, Fowler (2002) introduziu o padrão **Service Layer**, que define uma camada intermediária contendo a lógica de aplicação.

Combinando ambos os padrões, este projeto adota uma arquitetura em camadas que separa claramente as responsabilidades de controle, serviços e modelo.

4.1.2 Organização em Camadas

A aplicação foi estruturada em camadas baseadas nos padrões *MVC* e *Service Layer*, adaptadas ao contexto do *Flask* e às demandas do sistema. A Figura 5 ilustra essa organização e as interações entre os componentes.

Figura 5 – Arquitetura em camadas do *Geolacta*

Fonte: Autor

A camada de visão (*View*) é responsável pela interação direta com o usuário, compreendendo os diretórios *Templates* e *Static*. O diretório *Templates* utiliza o mecanismo de renderização *Jinja2*, permitindo a criação de páginas dinâmicas e a reutilização de componentes visuais. O diretório *Static* armazena arquivos estáticos, como imagens, folhas de estilo *CSS* e scripts *JavaScript*.

A camada de controle (*Controllers*) atua como intermediária entre a interface e as regras de negócio. Sua principal responsabilidade consiste em receber requisições *HTTP*, processar os dados enviados pelo usuário, acionar os serviços necessários e retornar respostas apropriadas, seja por meio de páginas renderizadas ou estruturas *JSON*. Dessa forma, os controladores centralizam o fluxo de comunicação entre as diferentes camadas do sistema.

A camada de modelos (*Models*) representa as entidades persistidas no banco de dados, utilizando o *ORM SQLAlchemy* para abstrair operações relacionais.

A camada de serviços (*Services*) concentra as regras de negócio da aplicação, promovendo maior desacoplamento entre controladores e funcionalidades complexas. Os principais serviços implementados são:

- **grid.service**: manipulação e processamento de grades geográficas para geração e

organização dos pontos do jogo;

- **random_location**: geração de localizações aleatórias válidas dentro das regiões geográficas disponíveis;
- **streetview**: integração com a *Google Street View API*, validando e obtendo imagens panorâmicas;
- **scoring**: cálculo de pontuação com base na distância entre a localização real e o palpite do usuário;
- **image_processor**: operações de processamento e tratamento de imagens utilizadas pela aplicação.

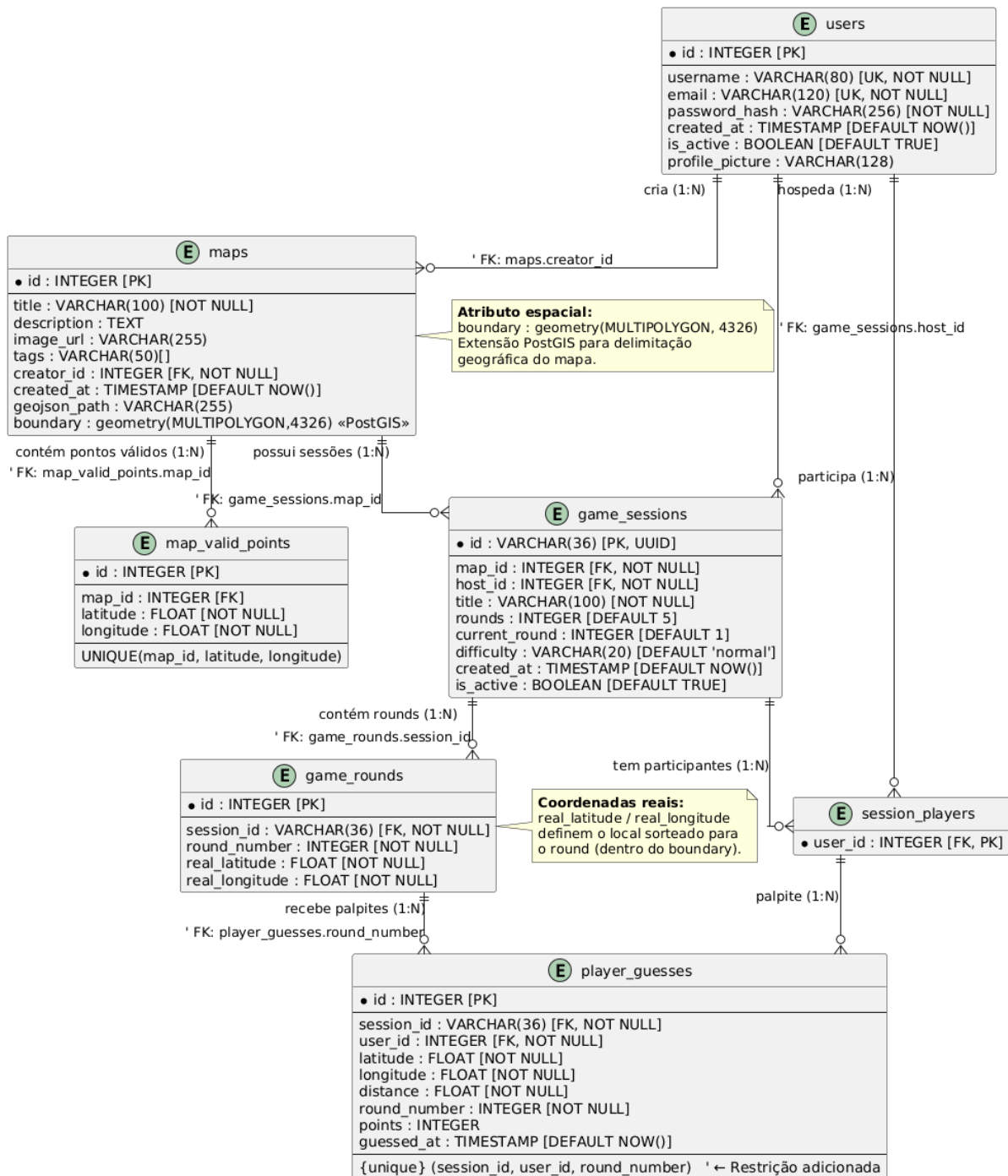
A persistência dos dados é realizada utilizando o sistema gerenciador de banco de dados *PostgreSQL* em conjunto com a extensão geoespacial *PostGIS*. Essa combinação possibilita o armazenamento e a manipulação eficiente de dados geográficos, permitindo consultas espaciais avançadas e operações geométricas como *ST_Contains* e *ST_DistanceSphere*, essenciais à mecânica do jogo. O sistema também integra a *Google Street View API* para obtenção de imagens panorâmicas, além das bibliotecas *Leaflet* (mapas interativos) e *D3.js* (visualizações de dados), já detalhadas na Seção 3.

O fluxo de funcionamento da arquitetura ocorre da seguinte maneira: inicialmente, o usuário interage com a camada de visão. As requisições são encaminhadas aos controladores, que processam os dados recebidos e acionam os serviços necessários. Os serviços realizam operações de negócio, consultas aos modelos e integrações externas. Em seguida, os resultados retornam aos controladores, que enviam respostas para a interface do usuário, seja por meio de páginas HTML renderizadas ou dados serializados em formato JSON. Essa separação em camadas proporciona maior organização estrutural, reduz o acoplamento entre componentes e facilita futuras expansões e manutenções da aplicação.

4.1.3 Modelo de Dados

O modelo de dados do sistema reflete os principais conceitos do domínio: usuários, sessões de jogo, rodadas e mapas. A figura 6 apresenta o Diagrama Entidade-Relacionamento (DER) completo.

Figura 6 – Modelo Entidade-Relacional do Geolacta



Fonte: Autor

As entidades são assim organizadas:

- **users:** armazena credenciais, perfil e indicadores de atividade;
- **maps:** modela os cenários de jogo, com destaque para o atributo; boundary do tipo geometry(MULTIPOLYGON, 4326) (sistema de referência EPSG:4326 - WGS84);

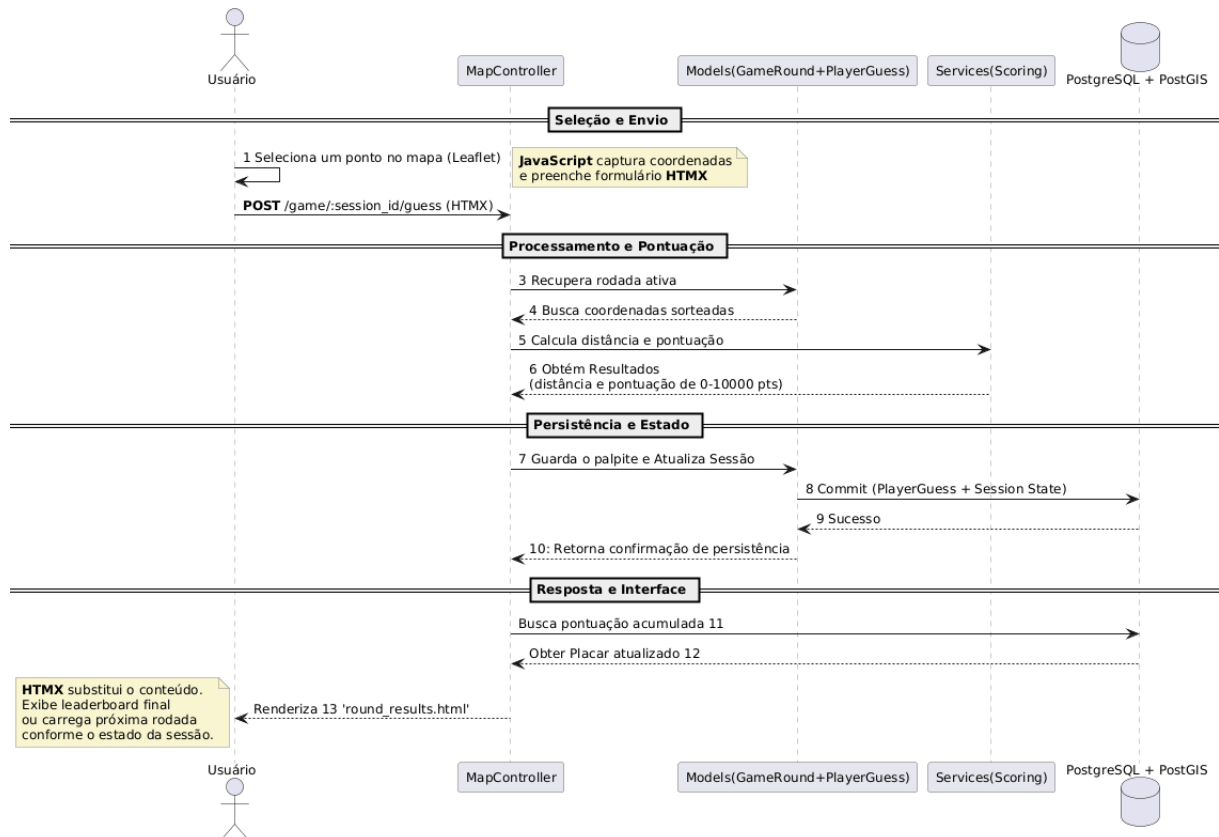
- `game_sessions`: representa partidas, referenciando um mapa (`map_id`) e um anfitrião (`host_id`);
- `session_players`: tabela associativa para relação muitos-para-muitos entre sessões e jogadores;
- `game_rounds`: armazena as coordenadas reais de cada rodada;
- `player_guesses`: registra palpites, distância calculada e pontuação;
- `map_valid_points`: permite pré-definir locais com cobertura de Street View.

As cardinalidades seguem a notação clássica (1:N, N:N). Índices foram definidos para atributos frequentemente consultados (`username`, `email`, chaves estrangeiras). A adoção do PostGIS justifica-se por operações espaciais como `ST_Contains` e `ST_DistanceSphere`, essenciais à mecânica do jogo.

4.1.4 Fluxo de controle: submissão de palpite

Para consolidar o entendimento da interação entre camadas, descreve-se o ciclo completo da ação central do jogo: o usuário submete um palpite em uma rodada. A Figura 7 ilustra esse fluxo por meio de um diagrama de sequência seguindo os padrões da *Unified Modeling Language* (UML).

Figura 7 – Diagrama de sequência para submissão de palpite



Fonte: Autor

As etapas representadas no diagrama de sequência da submissão de palpites descrevem o fluxo completo de interação entre as diferentes camadas arquiteturais do *Geolacta*, evidenciando a separação de responsabilidades entre interface, controle, serviços de negócio e persistência de dados. Essa divisão segue os princípios do padrão arquitetural MVC, permitindo maior organização, reutilização de código e facilidade de manutenção;

A camada de **visão** é representada pela interação direta do usuário com a interface web construída com *Leaflet*, *HTMX* e *Jinja2*. Nessa camada concentram-se os elementos responsáveis pela renderização visual do mapa, captura de eventos de clique e atualização dinâmica do conteúdo exibido ao jogador;

A camada de **controle** é representada pelo *MapController*, responsável por intermediar a comunicação entre a interface e as regras de negócio do sistema. O controlador recebe as requisições HTTP, valida os dados recebidos, coordena chamadas aos serviços internos e define qual resposta será retornada ao cliente;

A camada de **serviços** encapsula as regras de negócio da aplicação. Nesse fluxo, as funções dentro dos arquivos em *ScoringService* executa o cálculo da distância geográfica e

converte o resultado em pontuação, isolando a lógica matemática da camada de controle;

A camada de **modelo** é composta pelas entidades `GameRound` e `PlayerGuess`, manipuladas pelo ORM `SQLAlchemy`, além do banco de dados espacial *PostgreSQL/PostGIS*. Essa camada é responsável pelo armazenamento permanente das rodadas, palpites e pontuações da sessão.

O fluxo detalhado do processo ocorre da seguinte forma:

- 1 Seleção do ponto no mapa:** o usuário interage com o mapa renderizado pela biblioteca *Leaflet*, selecionando uma posição geográfica correspondente ao seu palpite. As coordenadas de latitude e longitude são capturadas via JavaScript e inseridas automaticamente em um formulário assíncrono controlado por HTMX;
- 2 Envio da requisição HTTP:** após a seleção do ponto, o navegador envia uma requisição do tipo POST para a rota `/game/:session_id/guess`. Essa comunicação ocorre de maneira assíncrona por meio do HTMX, evitando o recarregamento completo da página;
- 3–4 Recuperação da rodada ativa:** o `MapController` consulta a camada de modelos para recuperar os dados da rodada atual. Nesse momento, o sistema obtém as coordenadas reais associadas à localização sorteada da rodada em execução;
- 5–6 Processamento da pontuação:** o controlador delega ao `ScoringService` a responsabilidade pelo cálculo da distância geográfica entre o palpite do usuário e a localização correta. O serviço utiliza um algoritmo para calcular a distância sobre a superfície terrestre e converte o resultado em uma pontuação na faixa de 0 a 10.000 pontos;
- 7–10 Persistência e atualização do estado:** após o cálculo, o sistema cria um novo registro de `PlayerGuess`, armazenando coordenadas, distância e pontuação obtida. Em seguida, o estado da sessão é atualizado no banco de dados *PostgreSQL/PostGIS*, consolidando as alterações da rodada atual;
- 11–12 Cálculo do placar acumulado:** o controlador executa uma consulta adicional para recuperar a pontuação total acumulada pelo jogador durante a sessão. Essa etapa garante que o placar exibido na interface permaneça sincronizado com os dados persistidos no banco;
- 13 Renderização da resposta:** por fim, o controlador renderiza o `template round_results.html`, enviando ao cliente as informações da rodada, pontuação obtida e estado atualizado

da sessão. O HTMX substitui dinamicamente o conteúdo da interface, exibindo o resultado da rodada atual ou carregando automaticamente a próxima etapa do jogo.

Observa-se que o diagrama evidencia uma clara separação entre responsabilidades: a interface concentra-se exclusivamente na interação com o usuário; o controlador coordena o fluxo da aplicação; os serviços encapsulam regras matemáticas e operacionais; enquanto os modelos e o banco de dados garantem persistência e integridade das informações. Essa segmentação reduz o acoplamento entre componentes e favorece a escalabilidade e manutenção do sistema.

4.1.5 Ambiente de Desenvolvimento e Implantação com Containerização

Para garantir reprodutibilidade, isolar dependências e acelerar o ciclo de testes, o projeto foi containerizado utilizando Docker e gerenciado com Docker Compose. Dois contêineres principais foram definidos: um para o banco de dados e outro para a aplicação web.

O contêiner de banco de dados emprega a imagem `postgis/postgis`, que já incorpora a extensão geoespacial PostGIS sobre o PostgreSQL. Essa escolha evita a instalação manual de dependências específicas para operações com coordenadas geográficas, como cálculo de distâncias e testes de ponto em polígono. O contêiner é configurado com variáveis de ambiente para criação automática do banco, do usuário e da senha, além de um script de inicialização `init-db.sql` que prepara o esquema e as tabelas necessárias. Uma verificação de saúde (*healthcheck*) assegura que o banco esteja pronto antes de a aplicação tentar conectar-se.

O contêiner da aplicação web é construído a partir de um arquivo `Dockerfile` que instala as dependências listadas em `requirements.txt`. O comando de inicialização executa duas tarefas essenciais: popular o banco com mapas de exemplo via comando `flask seed-maps` e, em seguida, iniciar o servidor Flask em modo desenvolvimento com recarga automática. O mapeamento de volumes permite sincronizar o código-fonte entre a máquina hospedeira e o contêiner, dispensando reconstruções a cada alteração. Essa estratégia trouxe três vantagens principais:

1. **Reprodutibilidade:** qualquer desenvolvedor ou avaliador pode subir o ambiente completo com um único comando (`docker-compose up`), eliminando problemas de compatibilidade e instalação de dependências em diferentes dispositivos;

2. **Isolamento de dependências:** as versões do PostgreSQL, PostGIS e das bibliotecas Python ficam fixas no container, sem conflito com outros projetos no mesmo sistema operacional;
3. **Aceleração dos testes:** a inicialização do banco de dados é padronizada e rápida; os *healthchecks* garantem que a aplicação só tente conectar quando o banco estiver realmente pronto, reduzindo falhas.

4.1.6 Justificativa da arquitetura adotada

Três fatores principais motivaram a escolha por essa arquitetura, que combina o padrão MVC com uma camada de serviços explícita:

1. **Complexidade do domínio:** as operações geoespaciais, incluindo teste de ponto em polígono, amostragem aleatória e cálculo de distâncias, bem como a integração com APIs externas, por exemplo *Google Street View*, e o processamento de imagens exigem lógica sofisticada. Centralizar essa lógica em serviços separados evita a poluição dos controladores e facilita a manutenção;
2. **Manutenibilidade e evolução:** prevê-se a adição de novos modos de jogo, como mapa-múndi e desafio por tempo, além de diferentes fontes de dados como *OpenStreetMap*. Com a camada de serviços, tais extensões podem ser implementadas com impacto mínimo sobre as demais camadas;
3. **Testabilidade e isolamento:** a separação em camadas permite testar cada serviço independentemente, sem dependência de banco de dados real ou de APIs externas, aumentando a confiabilidade do sistema.

4.2 Algoritmos propostos

Para garantir o funcionamento correto da mecânica de jogo e da lógica geoespacial, três principais algoritmos foram desenvolvidos, com todos eles atuando na camada de serviços do jogo, compondo a lógica e os principais processos e mecanismos utilizados na plataforma. São eles: o algoritmo de Carregamento do mapa e das suas fronteiras (Seção 4.2.1), o algoritmo de Geração de localização aleatória válida (Seção 4.2.2) e o algoritmo de Cálculo da Distância Geográfica e Sistema de Pontuação (Seção 4.2.3).

4.2.1 Carregamento do mapa e das suas fronteiras

A geração de localizações é uma parte fundamental para jogos de adivinhação geográfica como o *GeoGuessr*. A escolha sobre como o *Geolacta* iria carregar e sortear as coordenadas geográficas fez parte de um processo longo, com várias etapas de tentativa e erro, e avaliações para conferir os potenciais pontos positivos e negativos de cada abordagem.

A última abordagem para o carregamento de fronteiras, essenciais para delimitar os limites geográficos que cada modo de jogo possui, foi carregada a partir de arquivos *GeoJSON*, armazenados no banco de dados espacial, que por sua vez extraiu as informações a respeito da fronteira de cada mapa por meio da coluna *boundary*. Essa coluna, por possuir o tipo espacial *geometry(MULTIPOLYGON, 4326)*, conseguiu definir perfeitamente os limites geográficos complexos de cada fronteira, que já estariam prontos para serem utilizados pela linguagem de programação através de uma chamada ao banco de dados.

Uma vez carregadas as fronteiras dos respectivos mapas no banco de dados geográfico, o sistema carrega os dados dessas respectivas fronteiras e busca dividi-las em uma grade com diferentes células. O objetivo deste processo é fragmentar o polígono político do mapa em uma matriz quadrada e extrair os pontos centrais (centróides) de cada quadrante que pertençam estritamente ao território jogável. A lógica de processamento e a estratégia de otimização via *cache* estão consolidadas no algoritmo 1.

Algoritmo 1: Geração de grade de centróides com filtro geométrico

Entrada: identificador *mapa_id*, número *tamanho_grade*

Saída: Estrutura com centróides válidos e dimensões das células

```

1 se os dados da grade para mapa_id já existem em cache então
2   | retorna dados_armazenados_no_cache;
3 fim
4 limites ← extrair_fronteras_retangulares(mapa_id);
5 celula ← calcular_dimensoes_da_celula(limites, tamanho_grade);
6 centroides_validos ← lista_vazia();
7 para cada célula gerada na interseção da grade faça
8   | ponto_central ← calcular_centroide_da_celula(célula, passos);
9   | se ponto_central está contido no polígono real do mapa então
10  |   | adicionar ponto_central em centroides_validos;
11  | fim
12 fim
13 x' resultado ← {centroides : centroides_validos, passos : passos};
14 armazenar_no_cache(mapa_id, resultado);
15 retorna resultado;

```

O funcionamento do algoritmo 1 baseia-se em um fluxo de processamento (*pipeline*) de geometria computacional otimizado para evitar processamento repetitivo, estruturado em quatro fases principais:

1. **Verificação de memória cache:** como o cálculo de intersecção geométrica dentro de polígonos complexos demanda alto custo de processamento, a rotina avalia primeiramente se o mapa em questão já teve sua grade computada anteriormente. Em caso positivo, os dados são retornados instantaneamente, poupando recursos do servidor;
2. **Delimitação da caixa envolvente:** caso o mapa seja processado pela primeira vez, a função *extrair_fronteras_retangulares* obtém os valores extremos de latitude e longitude do polígono político (caixa delimitadora). A partir dessas coordenadas máximas e mínimas, determina-se o tamanho de cada célula da matriz dividindo o espaço total pelo *tamanho_grade*;

3. **Filtragem topológica das células:** O algoritmo varre a matriz teórica gerada. Para cada quadrante, calcula-se a coordenada geodésica do seu centro exato. Antes de homologar a célula como utilizável para o jogo, o sistema verifica se esse ponto central está contido na área do território original. Pontos localizados em oceanos ou territórios internacionais adjacentes são descartados;
4. **Persistência e retorno:** ao final do mapeamento, a lista contendo apenas os centróides validados e as dimensões calculadas das células é empacotada, salva no *cache* global do sistema para consultas futuras e disponibilizada para os módulos de sorteio de partidas.

4.2.2 Geração de localização aleatória válida

O algoritmo 2 implementa uma estratégia de amostragem hierárquica para geração de localizações aleatórias com garantia de disponibilidade de imagens do *Google Street View*. A abordagem combina aleatoriedade controlada com validação espacial e externa, assegurando que os pontos gerados sejam jogáveis no contexto do *Geolacta*.

O processo inicia com a recuperação da grade de centroides previamente calculada através do algoritmo 1 descrito anteriormente. Cada célula da grade é subdividida em uma malha secundária, permitindo refinamento progressivo da busca. As células são processadas em ordem aleatória para evitar viés espacial, limitando-se o número máximo de células avaliadas por questão de performance.

Algoritmo 2: Estratégia de Amostragem Hierárquica de Alto Nível

```

1  celulas ← carregar_dados_das_celulas(mapa.id);
2  para até 3 células selecionadas aleatoriamente faça
3      sub_celulas ← gerar_subcelulas(celula);
4      para até 5 sub-células testadas aleatoriamente faça
5          ponto ← gerar_ponto_aleatorio(sub_celula);
6          se ponto está contido no polígono do mapa E possui Street View então
7              retorna ponto;
8          fim
9      fim
10 fim
11 retorna Exceção: Localização não encontrada;
```

O funcionamento do algoritmo 2 estrutura-se a partir de uma lógica de afunilamento espacial e resiliência, dividida em quatro etapas essenciais:

1. **Carregamento dos dados espaciais:** o procedimento inicia-se com a leitura da grade geográfica vinculada ao identificador do mapa. Essa função mapeia o território em macro-regiões denominadas células;
2. **Seleção limitada de células:** o sistema aplica um sorteio sobre as macro-regiões carregadas. Para limitar o tempo de processamento, o algoritmo restringe a busca a, no máximo, 3 células por execução. Caso uma célula falhe em fornecer um ponto válido, o fluxo avança automaticamente para a próxima das 3 células selecionadas;
3. **Subdivisão e teste de sub-células:** ao analisar uma célula ativa, o algoritmo a fragmenta em quadrantes menores (sub-células). O sistema compromete-se a testar de forma aleatória até 5 sub-células dessa região. Essa abordagem local evita que o algoritmo descarte uma macro-região inteira devido a uma inconsistência isolada;
4. **Geração de ponto e homologação dupla:** dentro de cada sub-célula testada, gera-se uma coordenada geográfica aleatória. O ponto passa por uma dupla validação eliminatória: primeiro, verifica-se em memória se as coordenadas pertencem ao polígono delimitador do mapa; segundo, consulta-se a infraestrutura externa para assegurar a existência de imagens do *Street View*.

Caso as 5 sub-células de uma determinada macro-região falhem em atender aos critérios de validação, o algoritmo realiza o recuo operacional *fallback*, saltando para a próxima célula macro. O processo repete-se até que uma localização utilizável seja encontrada ou que o limite de 3 células seja totalmente esgotado, situação na qual o sistema dispara uma exceção para impedir falhas na partida.

É importante notar, que a homologação do ponto candidato ocorre em duas camadas concêntricas de verificação. Primeiramente, aplica-se um teste de pertinência geométrica para assegurar que as coordenadas geradas estejam contidas no polígono delimitador do mapa, descartando-se instantaneamente pontos em áreas inválidas. Em seguida, submete-se o ponto válido espacialmente à API do *Google Street View* para confirmação da disponibilidade de imagens panorâmicas.

Esta consulta externa incorpora um mecanismo de controle de taxa (*rate limiting*) com intervalo mínimo de 250 ms entre requisições consecutivas, mitigando proativamente o erro HTTP 429 (*Too Many Requests*) imposto pelo provedor. Caso a API retorne status “OK” e coordenadas normalizadas do panorama, estas substituem o ponto aproximado original, garantindo precisão na localização final; caso contrário, o fluxo retorna *nulo*, acionando a próxima iteração do algoritmo hierárquico ou, se esgotadas as tentativas, propagando uma exceção controlada.

4.2.3 Cálculo da Distância Geográfica e Sistema de Pontuação

A determinação precisa da distância geográfica entre pontos na superfície terrestre desempenha um papel fundamental em diversas aplicações tecnológicas modernas, abrangendo desde sistemas de navegação por satélite e logística de transporte até plataformas de entretenimento baseadas em localização. Em jogos de adivinhação geográfica, como o *Geolacta*, essa métrica constitui o núcleo da mecânica do jogo, funcionando como o critério objetivo para avaliar a precisão das ações do usuário. Para viabilizar essa dinâmica, o sistema precisa processar as coordenadas esféricas fornecidas pela API de mapas e transformá-las em um indicador de desempenho, integrando diretamente o cálculo da distância física à atribuição de uma pontuação correspondente.

O algoritmo 3 unifica o processo de avaliação do palpite do usuário em uma única rotina computacional. Ele recebe a localização geográfica real do desafio e as coordenadas submetidas pelo jogador, aplica a fórmula do Haversine para determinar a distância esférica e, na sequência, computa a pontuação proporcional correspondente.

Algoritmo 3: Avaliação de Precisão Geográfica e Atribuição de Pontuação**Entrada:** Coordenadas (*lat_real*, *long_real*), Coordenadas (*lat_palpite*, *long_palpite*)**Saída:** Número inteiro representando a pontuação final do jogador

```

1  $R \leftarrow 6371$ ;
2 (lat_real, long_real, lat_palpite, long_palpite)  $\leftarrow$ 
   converter_para_radianos(lat_real, long_real, lat_palpite, long_palpite);
3  $dlat \leftarrow lat\_palpite - lat\_real$ ;
4  $dlong \leftarrow long\_palpite - long\_real$ ;
5  $a \leftarrow \sin^2\left(\frac{dlat}{2}\right) + \cos(lat\_real) \cdot \cos(lat\_palpite) \cdot \sin^2\left(\frac{dlong}{2}\right)$ ;
6  $c \leftarrow 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$ ;
7  $distancia \leftarrow R \cdot c$ ;
8  $PONTUACAO\_MAXIMA \leftarrow 10000$ ;
9  $pontuacao \leftarrow PONTUACAO\_MAXIMA \cdot \max\left(0, 1 - \frac{distancia}{2000}\right)$ ;
10 retorna inteiro(pontuacao);

```

O funcionamento do processo integrado de avaliação baseia-se em um fluxo de execução lógico estruturado nas seguintes etapas descritivas:

1. **Conversão espacial para radianos:** como as coordenadas de latitude e longitude obtidas na interface do usuário são expressas em graus decimais, o algoritmo inicia convertendo-as para radianos. Esta etapa é necessária para a compatibilidade com as funções trigonométricas da linguagem de programação utilizada;
2. **Cálculo do termo intermediário (*a*):** utilizando as diferenças obtidas entre as latitudes (*dlat*) e as longitudes (*dlong*), calcula-se a variável intermediária *a*, que compõe a primeira parte da equação matemática de Haversine:

$$a = \sin^2\left(\frac{dlat}{2}\right) + \cos(lat_real) \cdot \cos(lat_palpite) \cdot \sin^2\left(\frac{dlong}{2}\right)$$

3. **Determinação da distância linear (*distancia*):** com base no valor de *a*, determina-se a distância angular *c* por meio da função matemática *arctan2*. Na sequência, o valor da distância física final em quilômetros é consolidado multiplicando-se a distância angular *c* pelo raio médio aproximado da Terra ($R = 6371$ km);
4. **Cálculo proporcional da pontuação:** de posse da distância física em quilômetros, o algoritmo aplica a regra de negócio do jogo. A pontuação decresce linearmente a partir

de um teto predefinido de 10.000 pontos (*PONTUACAO_MAXIMA*) até um raio limite de 2.000 km. A função $\max(0, \cdot)$ assegura que qualquer palpite que exceda a distância limite de 2.000 km receba pontuação zero, evitando a atribuição de valores negativos;

5. **Discretização numérica:** por fim, o valor da pontuação obtido (originalmente em formato de ponto flutuante) é convertido para um tipo numérico inteiro através da operação *inteiro(pontuacao)*, adequando o dado ao formato esperado pela interface gráfica que renderiza os pontos para o usuário.

Dessa forma, o algoritmo estabelece um critério objetivo e automatizado para a validação dos palpites do usuário. Ao alinhar a fórmula de Haversine diretamente à lógica de pontuação do sistema, o *Geolacta* assegura um método consistente, replicável e computacionalmente estável para mensurar o desempenho dos jogadores.

5 RESULTADOS E DISCUSSÃO

Neste capítulo, são apresentados e analisados os principais resultados obtidos com a implementação do *Geolacta*. A discussão concentra-se nas interfaces gráficas desenvolvidas, nas funcionalidades implementadas e na aderência do sistema aos requisitos funcionais e não funcionais definidos na fase de projeto. Cada tela é descrita em termos de sua finalidade, componentes interativos e fundamentação teórica, destacando-se as contribuições para a experiência do usuário e para o monitoramento de desempenho eixos centrais da proposta.

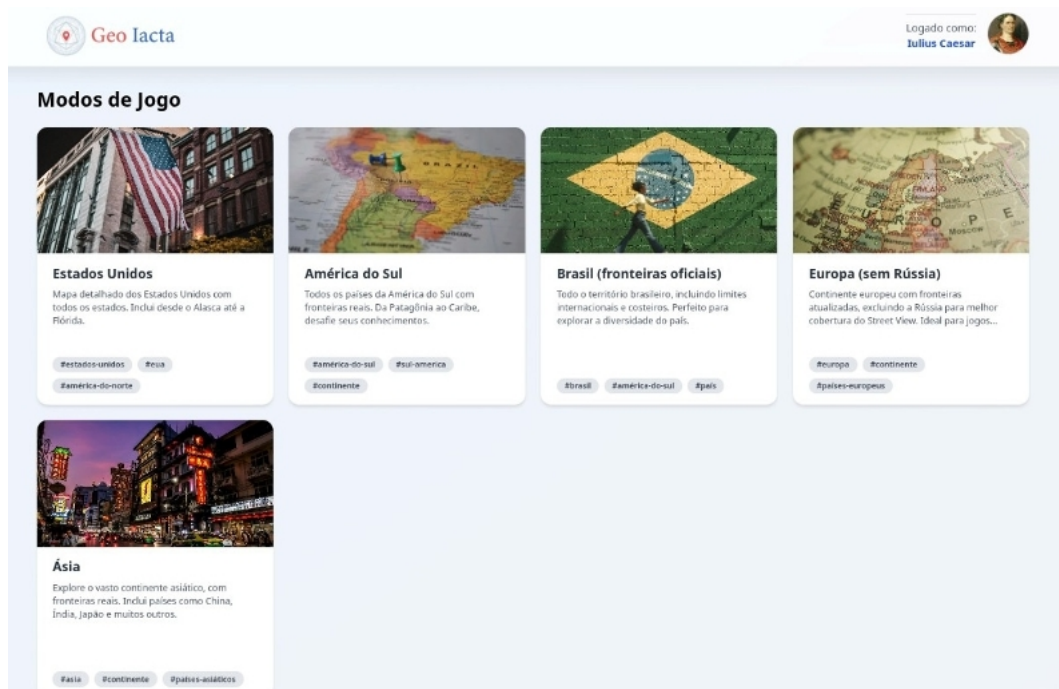
Ao final do capítulo, são explicitadas as principais limitações do protótipo atual, incluindo desafios técnicos, restrições algorítmicas e a impossibilidade de realização de testes controlados com usuários externos no escopo deste trabalho.

5.1 Telas desenvolvidas

Todas as interfaces do *Geolacta* foram projetadas seguindo os princípios de clareza, carga cognitiva mínima e obviedade visual propostos por Krug (2014), garantindo que o sistema seja autoexplicativo e de fácil navegação.

A interface de seleção de mapas (figura 8) permite ao usuário escolher o escopo geográfico da partida. Os cenários são organizados em cartões (*cards*) com imagem representativa, título e descrição, facilitando a identificação rápida das opções disponíveis.

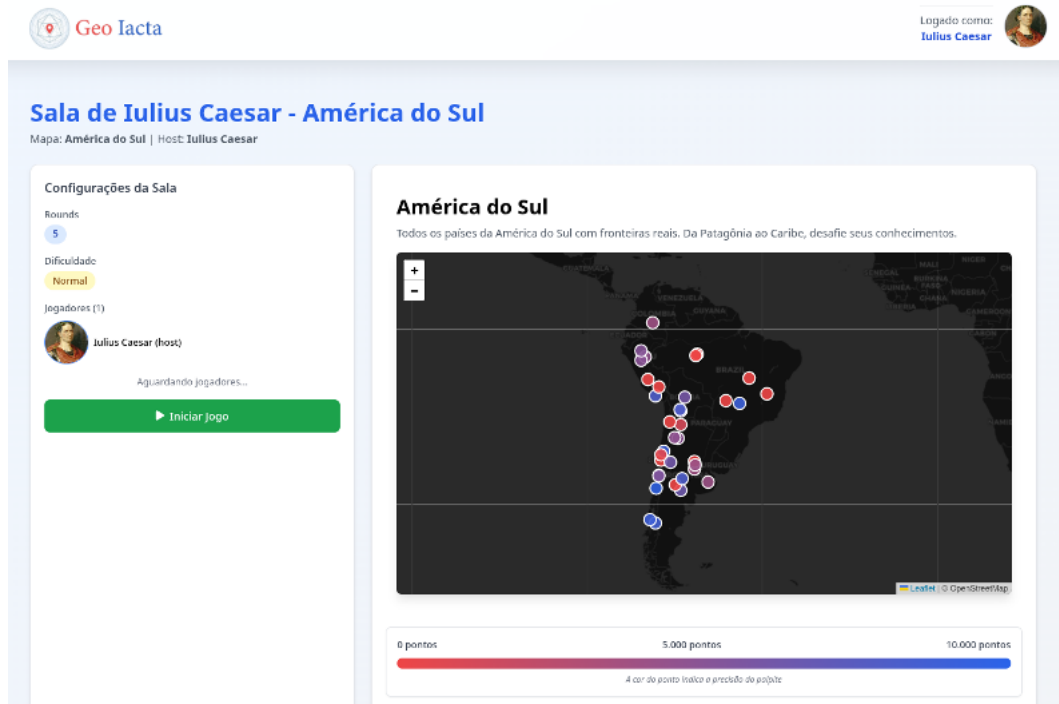
Figura 8 – Interface de seleção de modos de jogo



Fonte: Captura de tela realizada pelo autor

Na sala de preparação (*lobby*) (figura 9), o usuário visualiza parâmetros da partida (número de rodadas, dificuldade, jogadores conectados) e um botão verde de destaque para iniciar o jogo. A coluna direita exibe um mapa com o histórico de palpites do jogador na região escolhida, utilizando uma escala cromática (vermelho para baixa precisão, azul para alta precisão) para facilitar a análise de desempenho.

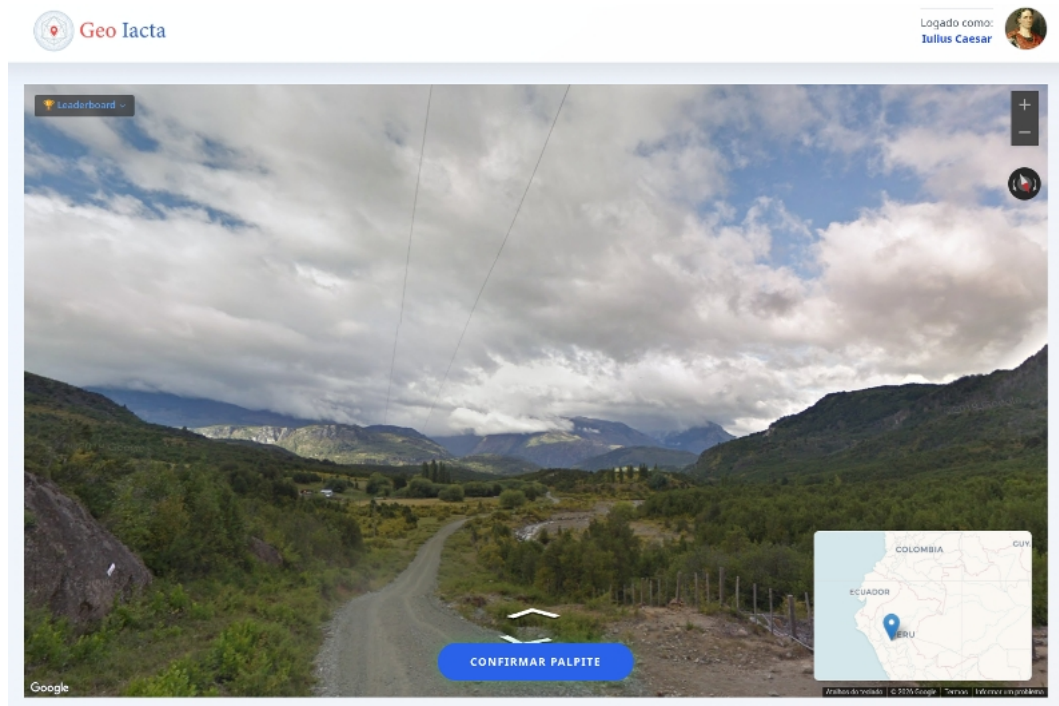
Figura 9 – Interface de carregamento de sessão



Fonte: Captura de tela realizada pelo autor

A tela de jogo (figura 10) integra uma visualização panorâmica do *Street View* em tela cheia com um minimapa flutuante. O jogador pode explorar a paisagem (relevo, vegetação, estradas) e, em seguida, clicar no minimapa para marcar sua estimativa. Um botão azul centralizado confirma o palpite e encerra a rodada.

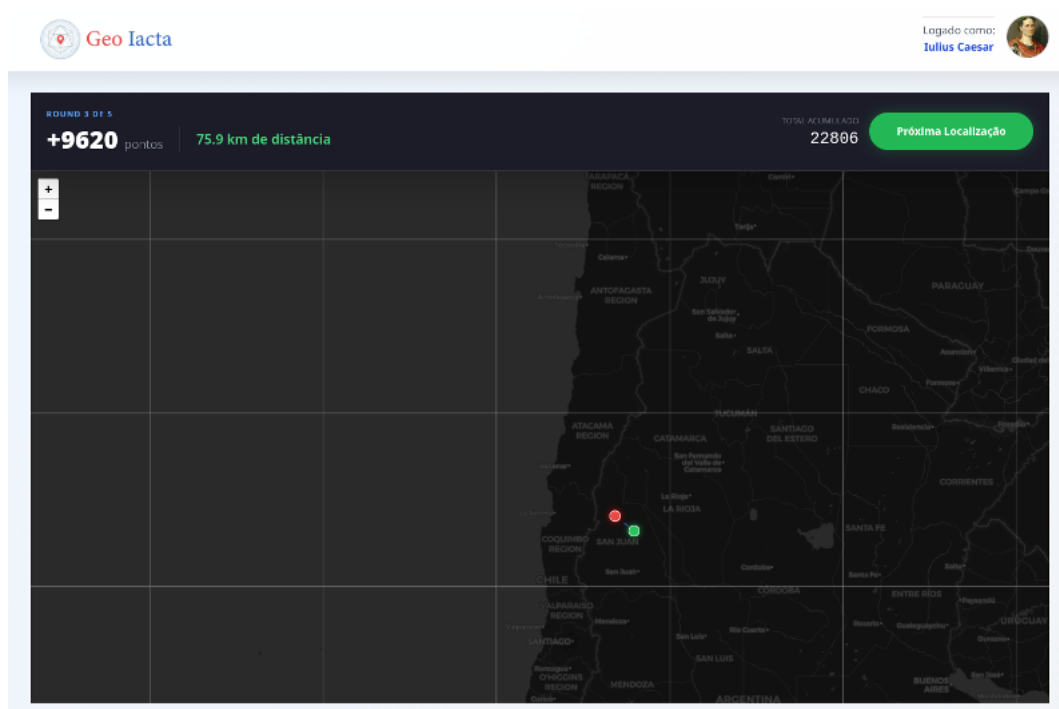
Figura 10 – Interface de Jogo



Fonte: Captura de tela realizada pelo autor

Após o palpite, a tela de resultados (figura 11) exibe um painel superior com a pontuação parcial, distância do erro e placar acumulado. O mapa central mostra dois marcadores (vermelho para o palpite, verde para a localização real) conectados por uma linha, permitindo comparação imediata.

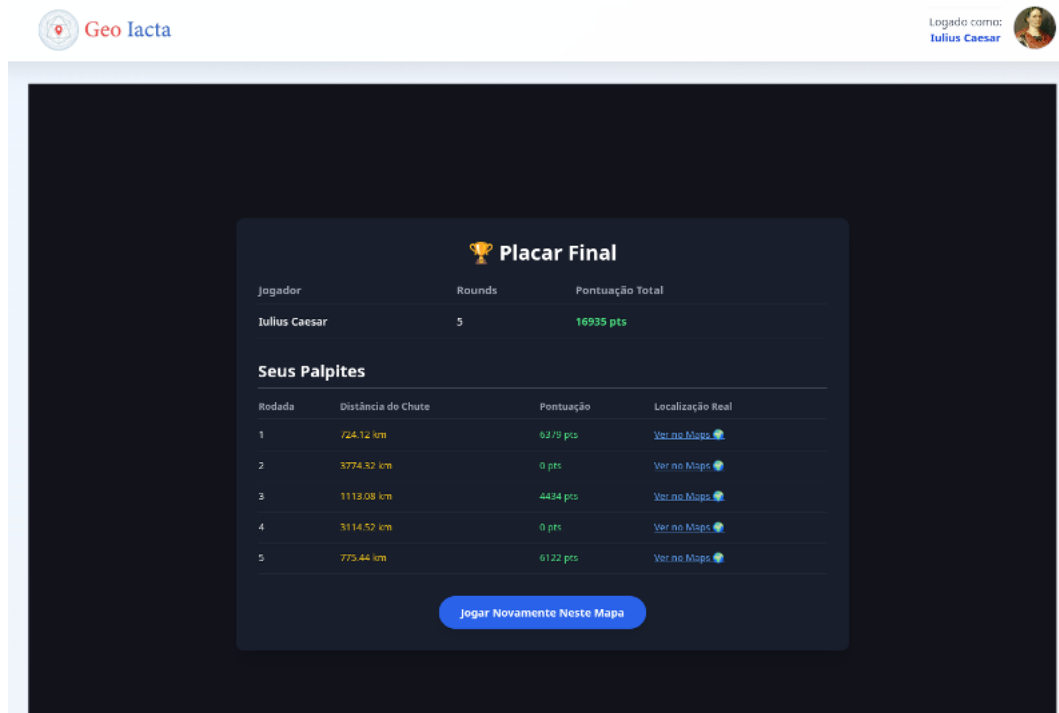
Figura 11 – Interface de resultados



Fonte: Captura de tela realizada pelo autor

Ao término das rodadas, a tela de placar final (figura 12) apresenta uma tabela detalhada com o desempenho do jogador rodada a rodada (distância e pontuação), além de atalhos para verificar a localização real em mapas externos, incentivando a revisão pedagógica.

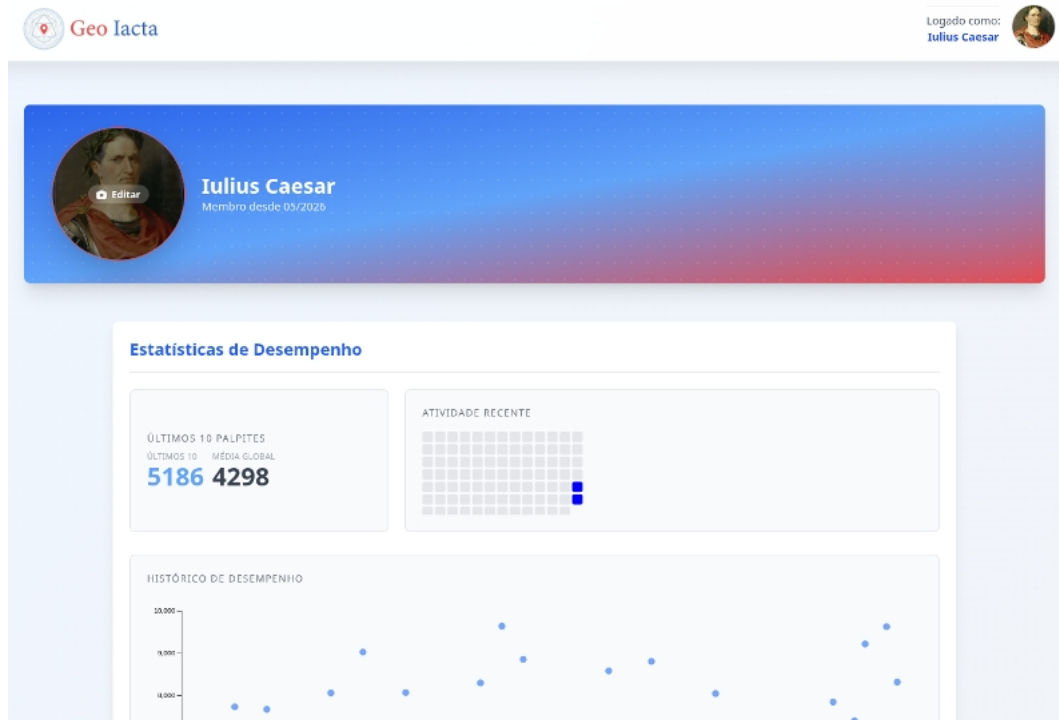
Figura 12 – Interface de resultados finais



Fonte: Captura de tela realizada pelo autor

A interface de perfil (figura 13) funciona como um painel analítico. Ela agrega um bloco de atividade recente (inspirado em gráficos de contribuição), um comparativo entre a média global e o desempenho recente do usuário, e um gráfico de dispersão com o histórico contínuo de pontuações. Esses elementos auxiliam na interpretação de padrões de aprendizagem e na autorregulação do jogador.

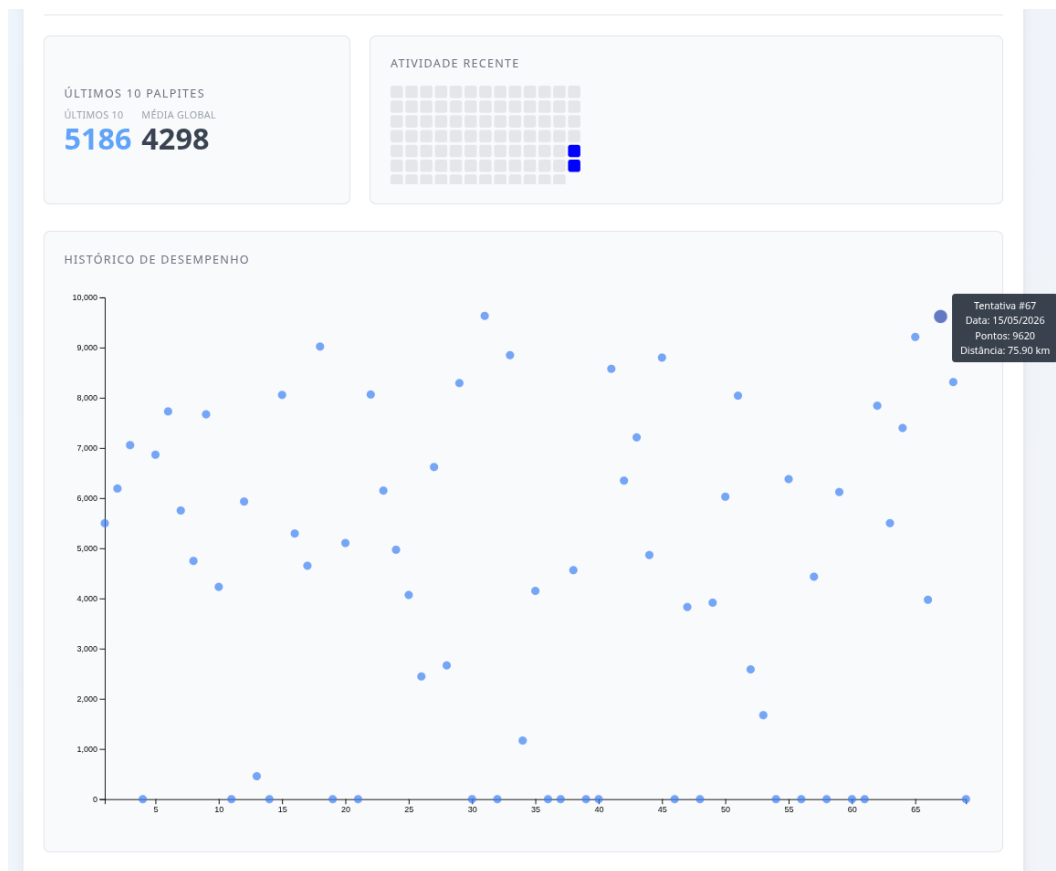
Figura 13 – Painel de estatísticas e histórico de perfil do usuário



Fonte: Captura de tela realizada pelo autor

Complementando o perfil, a figura 14 detalha as métricas numéricas: pontuação média dos últimos dez palpites versus média global, mapa de atividade recente em grade (quadrados coloridos conforme frequência) e gráfico de dispersão interativo com *tooltip* exibindo data, pontuação e distância de cada tentativa.

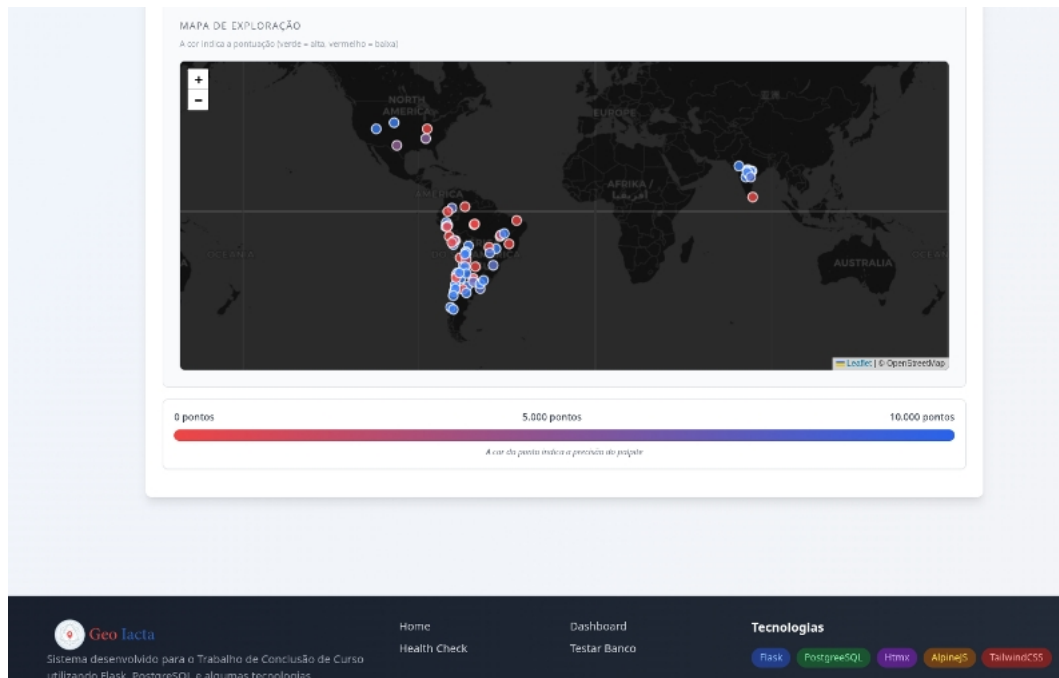
Figura 14 – Gráficos de perfil e estatísticas de desempenho



Fonte: Captura de tela realizada pelo autor

Por fim, o Mapa de Exploração Global (figura 15) carrega todo o histórico de palpites do usuário em todos os mapas, utilizando marcadores coloridos (mesma escala cromática) para indicar o nível de acerto em cada região. Diferente dos mapas de rodada, essa visualização cumulativa permite identificar, em nível global, quais áreas geográficas o jogador domina e quais necessitam de melhoria.

Figura 15 – Interface cartográfica de estatísticas de desempenho



Fonte: Captura de tela realizada pelo autor

5.2 Ressalvas e Limitações

Embora o protótipo desenvolvido cumpra satisfatoriamente os requisitos funcionais e não funcionais propostos, algumas limitações devem ser explicitamente registradas para qualificar os resultados e orientar trabalhos futuros.

5.2.1 Erro 429 em requisições à API do Google Street View

Durante os testes do sistema, observou-se a ocorrência esporádica do código de status *HTTP 429* (“Too Many Requests”) nas requisições à *Google Street View Static API*. Este erro indica que o servidor externo está recusando requisições devido à superação de uma cota ou taxa de solicitações por segundo. A imprevisibilidade do fenômeno, ora ocorrendo em momentos de baixa atividade, ora não se manifestando sob carga equivalente, sugere que fatores adicionais, possivelmente relacionados à política dinâmica de *rate limiting* do provedor ou à concorrência com outras aplicações que compartilham a mesma chave de API, podem estar atuando. A ausência de um mecanismo de *retry* com *backoff* exponencial na versão atual do código amplifica o impacto desse erro, podendo interromper temporariamente a geração de novas rodadas. Recomenda-se, em versões futuras, a implementação de uma camada de tratamento

robusta, incluindo filas de requisições e armazenamento em *cache* local das coordenadas com *Street View* confirmado.

5.2.2 Algoritmos de amostragem mais eficientes

O algoritmo de seleção aleatória de coordenadas utilizado no *Geolacta* baseia-se em uma grade hierárquica associada a polígonos GeoJSON, garantindo que os pontos sorteados estejam dentro dos limites do mapa. Contudo, esta abordagem não considera a distribuição real das vias de acesso ou a densidade da malha viária. Como consequência, há um viés intrínseco que pode levar à seleção de áreas excessivamente rurais, florestadas ou remotas, ambientes que, embora tecnicamente dentro do polígono, nem sempre oferecem imagens de *Street View* interessantes ou representativas para a experiência de adivinhação geográfica.

Na literatura técnica, soluções mais refinadas empregam dados de *OpenStreetMap* (OSM) para restringir a amostragem exclusivamente a trechos da malha rodoviária, ou ainda a pontos turísticos e marcos urbanos. A implementação de tal filtro exigiria o processamento prévio de grandes volumes de dados geoespaciais (por exemplo, baixar e consultar arquivos *Protocol Buffer Binary Format (PBF)* do OSM) e a integração de um banco de dados espacial capaz de realizar consultas de proximidade a vias com desempenho aceitável. Devido às restrições de tempo e escopo, sendo priorizada a finalização das funcionalidades de visualização de estatísticas e a estabilização do fluxo de jogo, essa otimização não foi incorporada ao protótipo atual. Reconhece-se, portanto, que o algoritmo atual pode subamostrar regiões urbanas e superamostrar zonas pouco povoadas, afetando a representatividade da experiência.

5.2.3 Ausência de avaliação com usuários externos

Por fim, cabe ressaltar que não foi possível realizar, no âmbito deste trabalho, uma avaliação sistemática com amostra de usuários externos (por exemplo, estudantes de geografia ou jogadores casuais). O projeto concentrou-se na fase de construção do protótipo funcional, garantindo que as funcionalidades de coleta e visualização de dados operassem corretamente. A validação da eficácia pedagógica, isto é, se o sistema realmente melhora o conhecimento geográfico ou a capacidade de localização espacial, depende de experimentos controlados com grupos de usuários, com etapas anterior e posterior ao teste, e análises estatísticas de significância. Tais atividades extrapolam o escopo definido para este trabalho, constituindo-se, contudo, como o passo natural de continuidade da pesquisa. Recomenda-se que futuros

trabalhos apliquem questionários de usabilidade como o *System Usability Scale* (SUS) e métricas de aprendizagem para quantificar o impacto real das visualizações de desempenho implementadas.

6 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo apresentar uma alternativa viável a sistemas que buscam criar uma experiência gamificada de aprendizagem geográfica, incorporando estatísticas para o monitoramento de desempenho dos usuários. Apesar de o produto final consistir em um *Mínimo Produto Viável* (MPV), é possível constatar que a experiência de desenvolvimento e os resultados alcançados foram bem-sucedidos, uma vez que o sistema atende a todos os requisitos funcionais e não funcionais definidos na fase de levantamento de requisitos, mesmo com uma quantidade limitada de mapas e funcionalidades, e problemas pontuais relacionados ao limite de requisições em torno da *Street View Static API*. Ademais, a arquitetura implementada possibilita a inserção de novas funcionalidades de maneira pouco custosa.

Entretanto, cabe ressaltar que o escopo deste trabalho concentrou-se exclusivamente na construção do sistema e na implementação das funcionalidades básicas. Avaliações sistemáticas com usuários reais, que poderiam validar a eficácia pedagógica e a usabilidade das ferramentas de visualização de progresso, **não fizeram parte do escopo atual**, constituindo-se como etapa necessária para trabalhos futuros.

Por fim, reforça-se a intenção de dar continuidade à pesquisa, avaliando a efetividade das visualizações de progresso e das mecânicas propostas, e adicionando novas funcionalidades. Somente com essa validação empírica será possível afirmar, com segurança, o potencial pedagógico do sistema desenvolvido.

6.1 Trabalhos Futuros

Embora o *Mínimo Produto Viável* tenha sido entregue, diversas funcionalidades podem ser exploradas em versões posteriores. Entre elas, destacam-se:

- Inserção de novos mapas prontos (países, regiões, biomas), expandindo a variedade de cenários disponíveis;
- Criação de uma ferramenta para selecionar ambientes que possuem características similares (ex.: florestas, desertos, áreas urbanas), permitindo ao jogador focar em tipos específicos de paisagem;
- Adição de um sistema *multiplayer* em tempo real, com salas competitivas onde jogadores possam disputar entre si;

- Implementação de soluções em nuvem (*Cloud Computing*) para que o sistema não fique restrito a uma máquina local, reduzindo drasticamente o risco de falhas e aumentando a disponibilidade;
- Hospedagem da aplicação em um domínio de fácil acesso e popularidade, facilitando a disseminação e o uso público;
- Adição de um sistema de visualização de ranking gamificados, com artifícios visuais para que os jogadores visualizem o seu progresso de maneira interativa;
- A adição de uma área para o administrador, caso seja aplicado em sala de aula, é ideal que um administrador, consiga monitorar o desempenho de uma turma como um todo, assim como as possíveis lacunas de conhecimentos gerais em uma sala;
- Implementação de modelos de inteligência artificial que analisem os resultados do jogador e forneçam dicas personalizadas sobre reconhecimento de ambientes e paisagens, promovendo aprendizado adaptativo;

A adição dessas funcionalidades conferiria ao sistema uma base mais robusta, com recursos que não apenas aprimorariam a experiência do usuário, mas também abririam portas para novas experiências que vão além das opções existentes no mercado. Além disso, tais melhorias preparariam e amadureceriam o projeto, tornando-o capaz de suportar uma carga significativamente maior de jogadores, de maneira segura e escalável.

Por fim, reforça-se a intenção de dar continuidade à pesquisa, avaliando a efetividade das visualizações de progresso e das mecânicas propostas. Somente com essa validação empírica será possível afirmar, com segurança, o potencial pedagógico do sistema desenvolvido.

REFERÊNCIAS

- Alam, M. I. et al. Investigating the impact of a gamified learning analytics dashboard: student experiences and academic achievement. **Journal of Computer Assisted Learning**, Wiley Online Library, v. 39, n. 5, p. 1436–1449, 2023.
- Ausubel, D. P. **A aquisição e retenção de conhecimentos: uma perspectiva cognitiva**. Lisboa: Plátano, 2003.
- Bennett, S.; Maton, K.; Kervin, L. The ‘digital natives’ debate: a critical review of the evidence. **British journal of educational technology**, Wiley Online Library, v. 39, n. 5, p. 775–786, 2008.
- Big Sky Software. **HTMX Documentation**. 2026. Disponível em: <https://htmx.org/docs/>. Acesso em: 05 maio 2026.
- Borges, T. da S. Gamificação na aula de geografia como estratégia de engajamento e motivação dos alunos. **REVISTA FOCO**, v. 16, n. 9, p. e2857–e2857, 2023.
- Borrella, I.; Ponce-Cueto, E. Design principles and impact of a learning analytics dashboard: evidence from a randomized mooc experiment. **Applied Sciences**, MDPI, v. 15, n. 21, p. 11493, 2025.
- Breda, T. V. Jogando com a geografia: possibilidades para um ensino divertido. **Giramundo: Revista de Geografia do Colégio Pedro II**, v. 5, n. 9, p. 55–63, 2018.
- Buschmann, F. et al. **Pattern-Oriented Software Architecture: A System of Patterns**. Chichester: Wiley, 1996.
- Castells, M. **A sociedade em rede**. 18. ed. São Paulo: Paz e Terra, 2017.
- CHACON, S.; Straub, B. **Pro Git**. 2. ed. New York: Apress, 2014.
- D3.js Contributors. **D3.js – Data-Driven Documents**. 2025. Disponível em: <https://d3js.org/>. Acesso em: 11 maio 2026.
- Docker, Inc. **Docker Documentation**. 2025. Disponível em: <https://docs.docker.com/>. Acesso em: 05 maio 2026.
- Felipe, B. A evolução da internet: uma perspectiva histórica. **Cadernos ASLEGIS**, v. 48, p. 35, 2013.
- Figlerowicz, M. Geoguessr’s digital pilgrimages. **Room One Thousand**, v. 3, n. 3, 2015.

Fowler, M. **Patterns of Enterprise Application Architecture**. Boston: Addison-Wesley, 2002. ISBN 9780321127426.

Freire, P. **Pedagogia da autonomia: saberes necessários à prática educativa**. São Paulo: Paz e Terra, 1996.

GeoAlchemy Contributors. **GeoAlchemy 2 Documentation**. 2026. Disponível em: <https://geoalchemy-2.readthedocs.io/>. Acesso em: 05 maio 2026.

Gomes. Gamificação e aprendizagem significativa: projetos interativos no ensino fundamental. **Cuadernos de Educación y Desarrollo**, v. 17, n. 4, p. e8080, 2025.

Google for Developers. **Best Practices Using Street View Static API**. 2026. Disponível em: <https://developers.google.com/maps/documentation/streetview/static-web-api-best-practices>. Acesso em: 05 maio 2026.

Google for Developers. **Maps JavaScript API Documentation**. 2026. Disponível em: <https://developers.google.com/maps/documentation/javascript/>. Acesso em: 05 maio 2026.

Google for Developers. **Services — Maps JavaScript API**. 2026. Disponível em: <https://developers.google.com/maps/documentation/javascript/services>. Acesso em: 05 maio 2026.

Google for Developers. **Street View Image Metadata**. 2026. Disponível em: <https://developers.google.com/maps/documentation/streetview/metadata>. Acesso em: 05 maio 2026.

Grinberg, M. **Flask Web Development: Developing Web Applications with Python**. 2. ed. Sebastopol, CA: O'Reilly Media, 2018.

Hamari, J.; Koivisto, J.; Sarsa, H. Does gamification work? – a literature review of empirical studies on gamification. In: **47th Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2014.

IBGE. **Internet chega a 74,9 milhões de domicílios do país em 2024**. 2025. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/44031-internet-chega-a-74-9-milhoes-de-domicilios-do-pais-em-2024>. Acesso em: 11 abr. 2026.

Kaner, C. **A Tutorial in Exploratory Testing**. Melbourne, 2008. Disponível em: http://www.kaner.com/pdfs/ET_stareast.pdf.

Krug, S. **Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability**. 3. ed. Berkeley, CA: New Riders, 2014. ISBN 978-0-321-96551-6.

Kuhn thomas s. **A ESTRUTURA DAS REVOLUCOES CIENTIFICAS**. 5. ed. Av. Brigadeiro Luís Antônio, 3025 01401-000 - Sao Paulo - SP - Brasil: EDITORA PERSPECTIVA, 1998.

KYD, A. **GeoJSON Maps of the Globe**. 2026. Acesso em: 25 janeiro 2026. Disponível em: <https://geojson-maps.kyd.au/>.

Leaflet Contributors. **Leaflet – an open-source JavaScript library for interactive maps**. 2025. Disponível em: <https://leafletjs.com/>. Acesso em: 11 maio 2026.

Libâneo, J. C. **Didática**. 2. ed. Rua Monte Alegre, 1074- Perdizes 05014-001- Sao Paulo - SP: Cortez Editora, 1999.

Montréal, A. do P. **Enhancing Student Engagement through Learning Analytics: A User-Centered Dashboard for LMS Platforms**. [S.l.], 2025. Disponível em: <https://publications.polymtl.ca/...> Acesso em: 11 abr. 2026.

Nascimento, I. do et al. Gamificação baseada em dados na educação básica: análise crítica das métricas de engajamento e sua influência sobre a aprendizagem significativa. **Missioneira**, v. 27, n. 2, p. 85–94, 2025.

Natural Earth. **Natural Earth Free Vector and Raster Map Data**. 2026. Dados cartográficos em domínio público. Acesso em: 25 janeiro 2026. Disponível em: <https://www.naturalearthdata.com/>.

Oliveira, M. E. M. de. Quebrando paradigmas: a educação em meio às tecnologias. **SCIAS-Educação, Comunicação e Tecnologia**, v. 1, n. 1, p. 62–78, 2019.

Oliveira, T. T. d. O raciocínio da localização como dimensão geográfica do jogo online geoguessr. 2022.

Pallets Projects. **Jinja — A very fast and expressive template engine**. 2024. Documentação oficial. Acesso em: maio de 2026. Disponível em: <https://jinja.palletsprojects.com/>.

Pei, B. et al. Learningviz: a dashboard for visualizing, analyzing and closing learning performance gaps—a case study approach. **Smart Learning Environments**, Springer, v. 11, n. 1, p. 56, 2024.

Policarpo, C.; Santaella, L. A estética do conhecimento nas redes digitais. **Dialogia**, p. 29–45, 2018.

Prensky, M. The emerging online life of the digital native. **Retrieved August**, v. 7, n. 2008, p. 253–263, 2004.

Pressman, R. S.; Maxim, B. R. **Software engineering**: a practitioner's approach. Ninth edition. New York, NY: McGraw-Hill Education, 2020. ISBN 978-1-260-54800-6 978-1-259-87297-6.

Pressman, R. S.; Maxim, B. R. **Engenharia de Software**: uma abordagem profissional. 9. ed. Porto Alegre: AMGH, 2021.

Rets et al. Exploring critical factors of the perceived usefulness of a learning analytics dashboard for distance university students. **International Journal of Educational Technology in Higher Education**, Springer, v. 18, n. 1, p. 46, 2021.

Ryan, R. M.; Deci, E. L. **Self-Determination Theory**: Basic Psychological Needs in Motivation, Development, and Wellness. [S.l.]: Guilford Publications, 2017.

Santos, M. **Metamorfoses do espaço habitado**. São Paulo: Edusp, 2007.

Schwab, K. **A Quarta Revolução Industrial**. São Paulo: Edipro, 2019.

Sena Ítalo Sousa de. **Visualização e valorização da paisagem a partir de geogame**. Tese (Tese (Doutorado em Geografia)) — Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, 2019. Disponível em: <http://hdl.handle.net/1843/32701>. Acesso em: Disponível em: <http://hdl.handle.net/1843/32701>.

Siqueira, B.; Moreno, F. G. O jogo geoguessr como metodologia ativa e interdisciplinar: uma proposta de integração entre geografia e matemática. **Terrae Didatica**, v. 17, p. e021051–e021051, 2021.

Sommerville, I. **Software engineering**. Tenth edition. Boston Columbus Indianapolis New York San Francisco Hoboken Amsterdam Cape Town Dubai London: Pearson, 2016. (Always learning). ISBN 978-1-292-09613-1 978-1-292-09614-8.

SQLAlchemy Authors and Contributors. **SQLAlchemy Documentation**. 2026. Disponível em: <https://docs.sqlalchemy.org/>. Acesso em: 05 maio 2026.

Stefenon, D. L. Entre paisagens e distâncias: o jogo geoguessr e seu potencial para a construção do pensamento conceitual nas aulas de geografia. **Giramundo: Revista de Geografia do Colégio Pedro II**, v. 3, n. 6, p. 30–40, 2016.

Tailwind Labs. **Tailwind CSS Documentation**. 2025. Disponível em: <https://tailwindcss.com/docs>. Acesso em: 11 maio 2026.

UIT. **Facts and Figures 2025**. 2025. Disponível em: <https://www.itu.int/itu-d/reports/statistics/facts-figures-2025/>. Acesso em: 11 abr. 2026.

Vianna, Y. et al. **Gamification, Inc. Como reinventar empresas a partir de jogos**. Rio de Janeiro: MJV Press, 2013. ISBN 978-85-65424-09-7.

Vieira, C.; Parsons, P.; Byrd, V. Visual learning analytics of educational data: a systematic literature review and research agenda. **Computers & Education**, v. 122, p. 119–135, 2018.

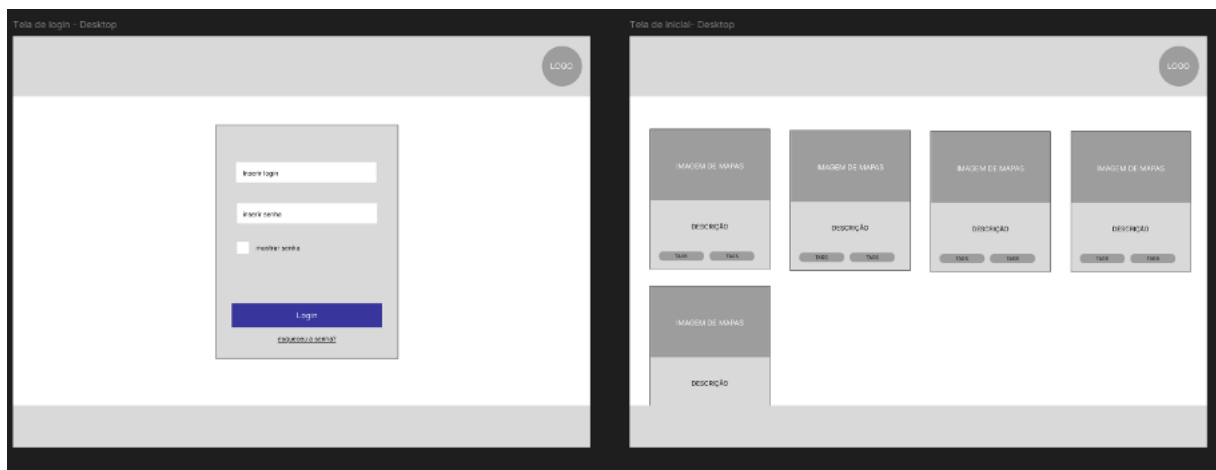
WTForms Team. **WTForms Documentation**. 2026. Disponível em: <https://wtforms.readthedocs.io/>. Acesso em: 05 maio 2026.

APÊNDICE A – PROTÓTIPOS

Um *website*, idealmente, deve ter seu *design* e funcionalidades bem definidos e otimizados com o fim de reduzir a carga cognitiva do usuário. Como dito em Krug (2014), um site mais simples permite que o usuário explore suas funcionalidades com satisfação e objetividade, elevando por si a qualidade do software desenvolvido.

A partir desse conceito e da filosofia incremental de desenvolvimento Pressman e Maxim (2021), foram feitos protótipos de interfaces, denominados *wireframes*. Tais interfaces são uma visão simplificada que passa por um processo de revisão antes do desenvolvimento prático, auxiliando posteriormente na programação das interfaces e seus componentes.

Figura 16 – Protótipos *desktop*: login e tela inicial.



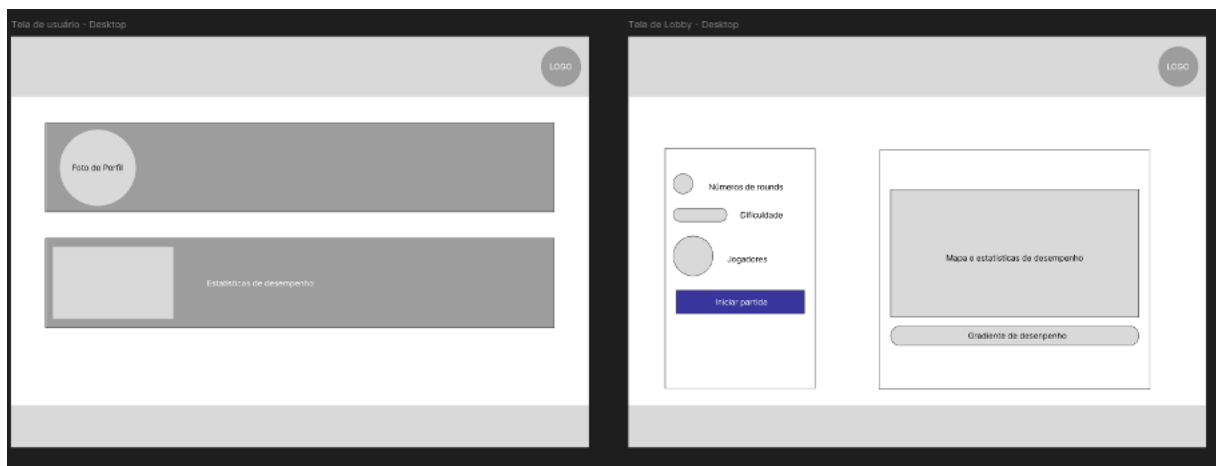
Fonte: Protótipos desenvolvidos pelo autor por meio do Figma

O protótipo apresentado na Figura 16 representa, inicialmente, uma interface voltada à segurança e ao controle de acesso, cuja funcionalidade principal é a autenticação de usuários. A tela de *login* é composta por campos específicos para a inserção das credenciais de acesso: “Inserir *login*” e “Inserir senha”. Para otimizar a experiência do usuário e evitar erros de digitação, foi implementada a opção “Mostrar senha”, que permite a conferência visual do código inserido. O fluxo de acesso é consolidado pelo botão de ação “*Login*”, destinado ao envio das credenciais para validação, contando ainda com o recurso “Esqueceu a senha?”, que direciona o usuário ao fluxo de recuperação de conta.

Dando continuidade à arquitetura visual, a tela inicial em modo *desktop* funciona como o painel principal de interação, tendo como objetivo a seleção de mapas pelo usuário. Nesta interface, os mapas são organizados de forma sistemática em uma estrutura de grade,

acompanhados por suas respectivas imagens ilustrativas e descrições contextuais. Para facilitar a navegabilidade e a organização do conteúdo, o sistema utiliza um método de categorização baseado em etiquetas (*tags*), o que permite filtrar as opções disponíveis de acordo com temas ou níveis de dificuldade, conferindo maior dinamismo à interface e auxiliando o usuário na tomada de decisão.

Figura 17 – Protótipos *desktop*: perfil e *lobby*.

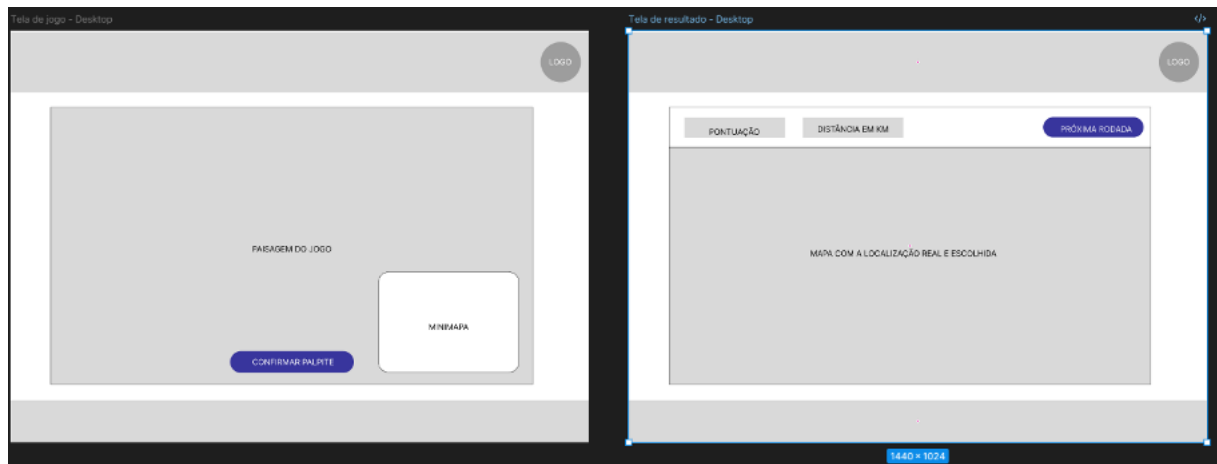


Fonte: Protótipos desenvolvidos pelo autor por meio do Figma

A interface dedicada ao usuário no ambiente *desktop* (Figura 17) tem como objetivo centralizar informações personalizadas e monitorar métricas individuais. No topo da página, um cabeçalho (*header*) organiza a identidade visual do jogador, exibindo a foto de perfil em conjunto com seus dados básicos de cadastro. Complementando essa visualização, a tela integra uma seção específica de “Estatísticas de desempenho”, projetada para permitir uma análise detalhada do progresso do usuário, facilitando a observação de sua evolução ao longo do tempo.

Em continuidade ao fluxo de interação, a tela de *lobby* atua como interface de transição e preparação para a experiência principal, permitindo a configuração personalizada de cada partida antes de seu início efetivo. Nesta etapa, o usuário possui controle sobre variáveis críticas, como o ajuste do número de rodadas, a seleção do nível de dificuldade e a visualização dos demais participantes presentes na sala. Adicionalmente, o sistema fornece uma camada de dados contextuais ao exibir as estatísticas prévias do mapa selecionado, conferindo ao jogador subsídios para sua estratégia. O ciclo de configuração é encerrado pelo botão “Iniciar partida”, que valida os parâmetros escolhidos e dispara o início da atividade.

Figura 18 – Protótipos *desktop*: *gameplay* e resultados.



Fonte: Protótipos desenvolvidos pelo autor por meio do Figma

A interface de jogo (Figura 18) constitui o núcleo da interação geográfica do sistema, priorizando a imersão do usuário por meio de uma paisagem exibida em tela cheia, o que facilita a exploração visual e a busca por pistas contextuais. Integrado a essa visualização, encontra-se um minimapa interativo posicionado de forma estratégica, que permite ao jogador inserir seu palpite geográfico com precisão. O fluxo de interação nesta tela é concluído pelo botão “CONFIRMAR PALPITE”, que registra a coordenada escolhida e encaminha o usuário para a etapa de validação.

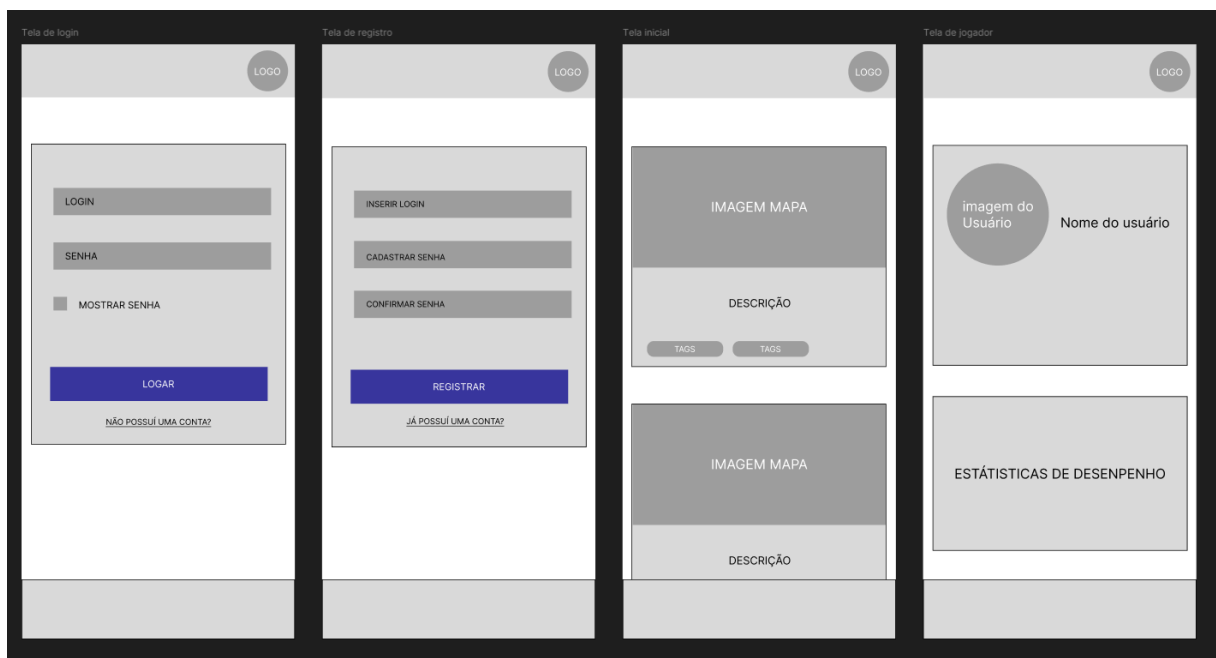
Após a submissão do palpite, o sistema apresenta a tela de resultados, cujo propósito é fornecer um *feedback* imediato e detalhado sobre o desempenho na rodada. Nesta interface, são exibidas métricas fundamentais, como a pontuação obtida e a distância exata (em quilômetros) entre o ponto selecionado e a localização real. Para enriquecer a análise do jogador, a tela oferece uma comparação visual direta no mapa, destacando ambos os pontos e o erro de proximidade. Por fim, o botão “PRÓXIMA RODADA” permite a continuidade do ciclo de jogo, conduzindo o usuário para o desafio subsequente de forma fluida.

O *design* responsivo é um princípio fundamental no desenvolvimento *web* contemporâneo. Como a aplicação desenvolvida se trata de um site acessível por múltiplos dispositivos, garantir a adaptabilidade da interface a diferentes tamanhos de tela não é uma questão de estética, mas uma necessidade estratégica e técnica. A importância do *design* responsivo reside, primeiramente, em dados de comportamento do usuário: estatísticas globais indicam que a maior parte do tráfego da *web* já provém de dispositivos móveis (*smartphones* e *tablets*), o que torna imperativo que os sites ofereçam uma experiência de qualidade nesses aparelhos. Além disso, motores de

busca, como o Google, adotaram práticas de indexação que priorizam versões *mobile*. Ignorar esse princípio resulta em experiências de navegação insatisfatórias, penalização nos rankings de busca e, conseqüentemente, perda de audiência e credibilidade.

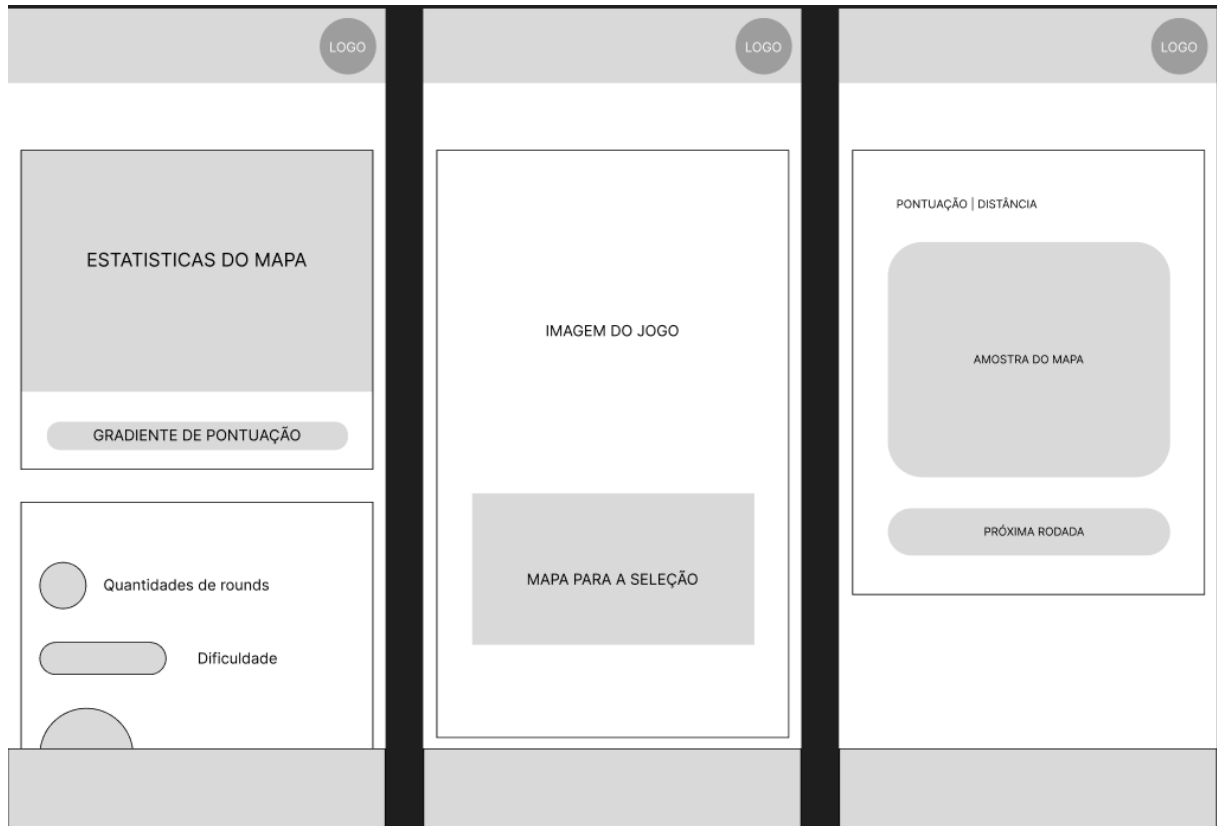
Com base nessas demandas, foram desenvolvidos protótipos de alta fidelidade para as telas do sistema em dispositivos móveis, utilizando a ferramenta Figma. Estes protótipos observados nas figuras 19 e 20 foram concebidos para garantir usabilidade e clareza em telas menores, respeitando as diretrizes de *design* e interação para dispositivos móveis.

Figura 19 – Protótipos *mobile*: telas de *login*, registro, seleção de mapas e perfil do usuário.



Fonte: Protótipos desenvolvidos pelo autor por meio do Figma

Figura 20 – Protótipos *mobile*: tela de características do mapa, tela de jogo e tela de exibição de resultados.



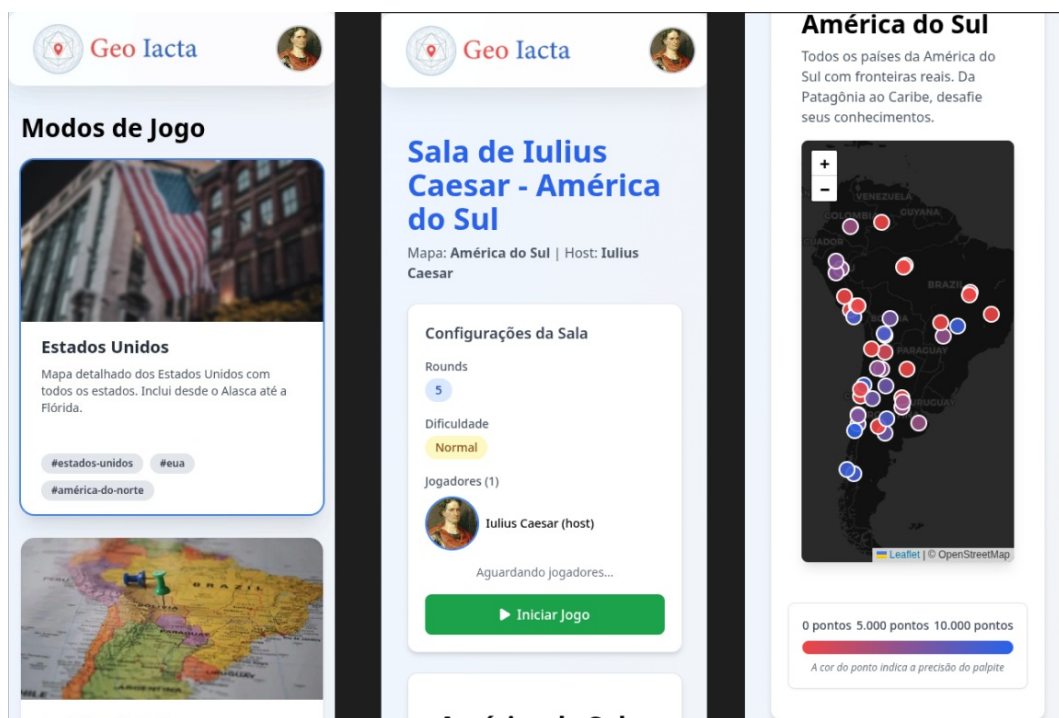
Fonte: Protótipos desenvolvidos pelo autor por meio do Figma

APÊNDICE B – INTERFACES MOBILE

O apêndice B apresenta as interfaces da aplicação em sua versão para dispositivos móveis, demonstrando a adaptação do *design* responsivo às telas de menor proporção. Foram desenvolvidas capturas de tela reais do sistema, abrangendo desde o fluxo de autenticação até a exibição dos resultados finais, com o objetivo de ilustrar como os elementos gráficos e as interações foram reorganizados para atender às especificidades dos *smartphones*.

As imagens evidenciam soluções como o empilhamento vertical dos componentes, o uso de sobreposições para o minimapa de palpites e a hierarquia visual por cores nas pontuações. O foco principal reside na manutenção da usabilidade e da clareza informacional, mesmo diante das limitações de espaço típicas dos dispositivos móveis, garantindo que a experiência de jogo permaneça fluida e intuitiva.

Figura 21 – Interface de seleção de mapas em dispositivos móveis



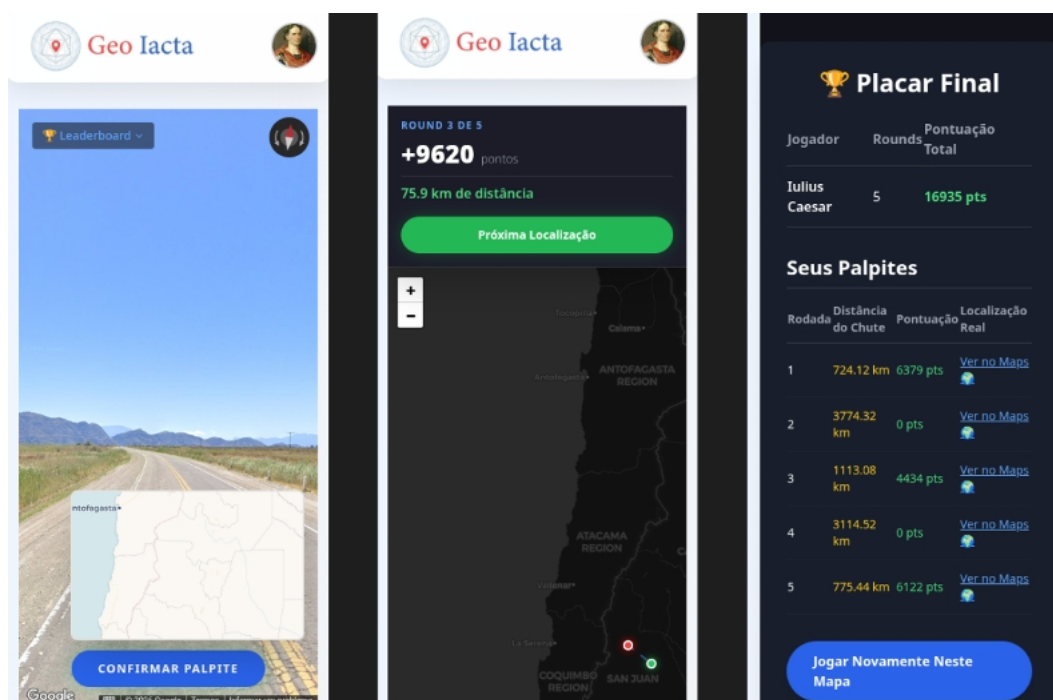
Fonte: Captura de tela realizada pelo autor

A imagem a esquerda da figura 21 ilustra a adaptação do protótipo para dispositivos móveis, utilizando *layout* de coluna única que prioriza a rolagem vertical. Os cartões de seleção de mapas foram redimensionados para ocupar a largura total da tela, mantendo a proporção das imagens de capa. Abaixo de cada imagem, o título e a descrição utilizam tipografia sem serifa para facilitar a leitura.

As duas interfaces restantes da figura 21 representam a página de preparação para a partida, adapta a complexidade da interface de carregamento de sessão para uma visualização vertical em coluna única. O contêiner “Configurações da Sala” organiza de forma modular os seletores de rodadas, o nível de dificuldade e a lista de jogadores ativos. Abaixo deste painel, o sistema apresenta um resumo do mapa selecionado com uma breve descrição textual, preparando a transição para o componente cartográfico.

Na sequência, o botão “Iniciar Jogo” é posicionado estrategicamente na base do primeiro bloco, utilizando a cor verde para sinalizar a ação de confirmação e início do fluxo de *gameplay*. Essa organização em duas partes visa facilitar a leitura em telas pequenas.

Figura 22 – Interface de jogo em dispositivos móveis.



Fonte: Captura de tela realizada pelo autor

A interface de interação principal observada na imagem a esquerda da figura 22 foca na maximização da experiência imersiva com *layout* vertical em tela cheia (*full-viewport*). O elemento central é o *canvas* do *Google Street View*, que permite exploração panorâmica por gestos de toque. O minimapa de palpites é implementado como sobreposição (*overlay*) na parte inferior, permitindo alternar entre observação da paisagem e marcação de coordenadas. Os controles secundários são posicionados discretamente nas extremidades, priorizando a navegação 360°.

A versão mobile da interface de resultados visualizada na interface central da da figura 22,

evidencia técnicas de *design* responsivo. Os elementos foram reorganizados em estrutura de coluna única, priorizando leitura vertical e interação por toque (*touch inputs*). Ícones como o globo terrestre economizam espaço, mantendo a semântica da ação. A soma da pontuação e o histórico de rodadas permanecem acessíveis, respeitando as limitações da *viewport*.

A versão para dispositivos móveis visualizada na interface a direita da figura 22 demonstra a aplicação de princípios de *responsive design*, empilhando verticalmente os elementos da listagem. O cabeçalho expandido é eliminado para otimizar o espaço de tela, mantendo o foco exclusivo no quadro de pontuação e nos detalhes das rodadas. O botão “Jogar Novamente Neste Mapa” ocupa toda a largura da base, facilitando a interação com o polegar. As colunas de dados mantêm hierarquia visual por cores (verde para pontuações, amarelo para distâncias), assegurando que a comparação e revisão dos resultados permaneçam íntegras e intuitivas.