

Tecendo Saberes: O desenvolvimento de uma *Application Programming Interface* (API) educacional para a preservação da cultura dos povos originários

Weaving Knowledge: The development of an educational *Application Programming Interface* (API) for the preservation of indigenous peoples' culture

Tejiendo Conocimientos: El desarrollo de una *Application Programming Interface* (API) educativa para la preservación de la cultura de los pueblos indígenas

Recebido: 15/01/2026 | Revisado: 23/01/2026 | Aceitado: 24/01/2026 | Publicado: 25/01/2026

Sebastian Henrique da Silva¹

ORCID: <https://orcid.org/0009-0003-9815-8649>

Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Brasil

E-mail: sebastian2dgir@gmail.com

Yara Regina Pereira Silva Menezes de Sá¹

ORCID: <https://orcid.org/0009-0000-4308-2534>

Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Brasil

E-mail: yara.regina@ifsertao-pe.edu.br

Thiago Luiz dos Santos¹

ORCID: <https://orcid.org/0009-0003-4370-9052>

Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Brasil

E-mail: thiagoluizst14@gmail.com

Resumo

O Brasil possui uma vasta diversidade étnica, contudo, a disseminação dos saberes indígenas historicamente enfrenta barreiras de acessibilidade e escassez de materiais digitais práticos. Este artigo tem como objetivo apresentar e analisar o processo de desenvolvimento e a arquitetura técnica da API educacional Tecendo Saberes, evidenciando suas contribuições para a preservação digital dos saberes dos povos originários brasileiros e para o fortalecimento de práticas educacionais alinhadas à Educação 4.0 e à Educação para as Relações Étnico-Raciais. Metodologicamente, adotou-se uma abordagem de desenvolvimento de software ágil gerenciada via Kanban, utilizando a linguagem TypeScript. A arquitetura do sistema foi construída com um backend em NestJS e Prisma ORM, operando sobre um banco de dados relacional PostgreSQL, integrado a uma interface frontend desenvolvida em Next.js e React.js para validação e experiência do usuário. Os resultados demonstram a implementação bem-sucedida das operações de gerenciamento de dados (CRUD) e uma modelagem que preserva o vínculo entre termos, territórios e etnias. Uma decisão estratégica relevante foi a remoção da obrigatoriedade de autenticação para consultas de leitura, mantendo a segurança via JWT apenas para operações de escrita, favorecendo a Ciência Aberta. Conclui-se que a ferramenta oferece uma infraestrutura escalável e modular, preenchendo uma lacuna tecnológica vital para o fortalecimento e a preservação digital da identidade cultural indígena.

Palavras-chave: API Educacional; Aplicação Web; Educação para as Relações Étnico-raciais (ERER); Povos Originários Brasileiros; Sustentabilidade Cultural; Tecnologia Educacional.

Abstract

Brazil has vast ethnic diversity, yet the dissemination of indigenous knowledge has historically faced barriers of accessibility and a scarcity of practical digital materials. This article aims to present and analyze the development process and technical architecture of the Tecendo Saberes educational API, highlighting its contributions to the digital preservation of the knowledge of Brazilian indigenous peoples and to the strengthening of educational practices aligned with Education 4.0 and Education for Ethnic-Racial Relations. Methodologically, an agile software development approach managed via Kanban was adopted, using the TypeScript language. The system architecture was built with a backend in NestJS and Prisma ORM, operating on a PostgreSQL relational database, integrated with a frontend interface developed in Next.js and React.js for validation and user experience. The results demonstrate the successful implementation of data management operations (CRUD) and a modeling that preserves the link between terms, territories, and ethnicities. A relevant strategic decision was to remove the authentication requirement for read queries, maintaining security via JWT only for write operations, favoring Open Science. It is concluded that the tool offers a

¹ Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Campus Floresta, Brasil.

scalable and modular infrastructure, filling a vital technological gap for the strengthening and digital preservation of indigenous cultural identity.

Keywords: Educational API; Web Application; Education for Ethnic-Racial Relations (ERER); Brazilian Indigenous Peoples; Cultural Sustainability; Educational Technology.

Resumen

Brasil cuenta con una gran diversidad étnica, sin embargo, la difusión del conocimiento indígena se ha enfrentado históricamente a barreras de accesibilidad y a la escasez de materiales digitales prácticos. El objetivo de este artículo es presentar y analizar el proceso de desarrollo y la arquitectura técnica de la API educativa Tecendo Saberes, destacando sus contribuciones a la preservación digital de los conocimientos de los pueblos originarios brasileños y al fortalecimiento de prácticas educativas alineadas con la Educación 4.0 y la Educación para las Relaciones Étnico-Raciales. Metodológicamente, se adoptó un enfoque de desarrollo de software ágil gestionado a través de Kanban, utilizando el lenguaje TypeScript. La arquitectura del sistema se construyó con un backend en NestJS y Prisma ORM, que opera sobre una base de datos relacional PostgreSQL, integrada a una interfaz frontend desarrollada en Next.js y React.js para la validación y la experiencia del usuario. Los resultados demuestran la implementación exitosa de las operaciones de gestión de datos (CRUD) y un modelado que preserva el vínculo entre términos, territorios y etnias. Una decisión estratégica relevante fue la eliminación de la autenticación obligatoria para las consultas de lectura, manteniendo la seguridad a través de JWT solo para las operaciones de escritura, lo que favorece la ciencia abierta. Se concluye que la herramienta ofrece una infraestructura escalable y modular, llenando un vacío tecnológico vital para el fortalecimiento y la preservación digital de la identidad cultural indígena.

Palabras clave: API educativa; Aplicación web; Educación para las Relaciones Étnico-Raciales (ERER); Pueblos Indígenas Brasileños; Sostenibilidad Cultural; Tecnología Educativa.

1. Introdução

O Brasil é um país multiétnico, com uma população composta por diferentes grupos indígenas que representam uma parte importante de sua diversidade cultural. A valorização das culturas, línguas e terminologias indígenas é essencial para a construção de uma sociedade mais justa e inclusiva, conforme indica a Base Nacional Comum Curricular (Brasil, 2018, p. 9-10). No entanto, a efetiva disseminação desses saberes historicamente enfrentou barreiras de acessibilidade e exclusão dos currículos, uma lacuna que a Lei nº 11.645/08 buscou mitigar ao tornar obrigatório o ensino da história e cultura indígena (Brasil, 2008). Apesar do avanço legislativo, a disponibilidade de materiais práticos, especialmente em meios digitais, muitas vezes permanece restrita. Essa lacuna é reconhecida pela UNESCO (2021), que alerta para a exclusão digital das comunidades indígenas e a urgência de desenvolver tecnologias que apoiem a preservação cultural desses povos. Nesse contexto, a criação de ferramentas tecnológicas torna-se imprescindível para democratizar o acesso a essas informações.

No contexto da Educação 4.0, muitas tecnologias digitais surgem como ferramentas poderosas capazes de atravessar essas barreiras. Elas oferecem novas possibilidades para documentar, organizar e, principalmente, disseminar o conhecimento de forma ampla e acessível. Para Vidal & Miguel (2020, p. 4) A educação 4.0 está inserida no contexto da Quarta Revolução Industrial, também conhecida como Indústria 4.0, onde tecnologias como a linguagem das máquinas, a Internet das Coisas, Inteligência Artificial, robôs, e outras tecnologias se tornam um conjunto dinâmico nos processos das mais variadas áreas da Indústria.

A partir da identificação da carência de ferramentas práticas e acessíveis para listar e disponibilizar informações sobre os povos originários que nasceu o projeto "TECENDO SABERES: Conexões entre Tecnologia e a Preservação da Cultura dos Povos Originários". O objetivo central do projeto é promover a valorização das línguas e terminologias indígenas através de uma solução tecnológica prática.

A solução proposta foi o desenvolvimento de uma API educacional. Para Torres (2021), a API (Interface de Programação de Aplicações) funciona como um conjunto de protocolos e padrões que viabilizam a comunicação e a integração entre diferentes sistemas de software. Esta aplicação fornecerá informações sobre povos, línguas, terminologias e aspectos culturais, garantindo também a preservação desses dados. As plataformas que consomem esses conhecimentos, como portais de pesquisa, aplicações educacionais e sites acadêmicos poderão utilizar essas informações para estudos e pesquisas.

Este artigo detalha o processo de desenvolvimento e a arquitetura técnica da API intitulada "TECENDO SABERES", tecnologia que visa disponibilizar informações sobre os povos originários, suas línguas e terminologias, assim como contribuir para o ensino e pesquisa no contexto da educação 4.0. Com isso, abordamos as escolhas metodológicas, a estruturação do banco de dados (PostgreSQL), a construção do backend (Node.js com Nest.js e Express.js) e a implementação de uma interface de uso para administradores e usuários (Next.js e React.js). Além disso, discutimos os principais resultados, desafios e decisões tomadas durante o ciclo de desenvolvimento, alinhando a solução tecnológica ao seu objetivo principal de disseminação do conhecimento.

Diante desse contexto, este artigo tem como objetivo apresentar e analisar o processo de desenvolvimento e a arquitetura técnica da API educacional Tecendo Saberes, evidenciando suas contribuições para a preservação digital dos saberes dos povos originários brasileiros e para o fortalecimento de práticas educacionais alinhadas à Educação 4.0 e à Educação para as Relações Étnico-Raciais.

2. Metodologia

Este estudo caracteriza-se como uma pesquisa aplicada, de abordagem qualitativa, com procedimentos de desenvolvimento tecnológico (Azevedo, 2018; Pereira et al., 2018; Gil, 2017). O trabalho envolveu a concepção, implementação e validação funcional de uma API educacional, orientada por princípios pedagógicos e éticos vinculados à Educação para as Relações Étnico-Raciais, visando à produção de uma solução tecnológica com finalidade social e educacional.

Dessa forma, o desenvolvimento da API foi norteado de maneira que pudesse evitar uma estrutura de software neutra ou descontextualizada, desenhada para preservar o conhecimento de uma forma que combate estereótipos e valoriza a diversidade cultural, reconhecendo a pluralidade de povos, línguas, territórios e saberes indígenas. Essa abordagem dialoga com a Política Nacional de Equidade (PNEERQ), instituída pela Portaria MEC nº 470/2024. O documento define como diretriz fundamental o enfrentamento das desigualdades étnico-raciais na educação e a equidade nas condições de oferta, reconhecendo que a igualdade formal não é suficiente sem políticas específicas para grupos vulneráveis (Brasil, 2024, Art. 2º, VIII). Seguindo essa premissa, a modelagem do banco de dados quebra com a visão generalista sobre os povos originários ao impor uma estrutura relacional que vincula estritamente cada dado a povos e territórios específicos, demonstrando que não existe uma "cultura indígena" única, mas sim uma pluralidade de povos com línguas, territórios e saberes distintos.

2.1 Planejamento e Definição do Escopo do Projeto

A definição do escopo e o planejamento do sistema seguiram práticas padronizadas da indústria. Inicialmente, realizou-se o levantamento dos requisitos funcionais e não funcionais, etapa descrita pelo Guia SWEBOK v4 (IEEE Computer Society, 2025) como fundamental para estabelecer as fronteiras do sistema e as restrições de design. Com o escopo delimitado, a estrutura do banco de dados foi modelada para suportar as funcionalidades previstas. Para a gestão do fluxo de trabalho, adotou-se a metodologia Kanban. Conforme o Guia Oficial (Kanban University, 2020), essa abordagem permite a visualização transparente das tarefas e o equilíbrio do fluxo de entrega, facilitando as interações incrementais e a evolução contínua da documentação.

2.1.1 Objetivos Gerais do Projeto

Desenvolver uma API Educacional eficiente, documentada e segura para integrar e disponibilizar informações sobre os povos originários, suas línguas e terminologias, contribuindo para o ensino e pesquisa no contexto da Educação 4.0.

2.1.2 Objetivos Específicos do Projeto

- Criar um banco de dados estruturado que contenha informações detalhadas sobre os povos originários brasileiros.
- Desenvolver endpoints que permitam consultas flexíveis sobre línguas, terminologias e aspectos culturais.

- Garantir a interoperabilidade da API com plataformas educacionais e de pesquisa.
- Oferecer uma documentação detalhada e amigável para desenvolvedores e educadores.
- Incorporar medidas de segurança para proteger os dados sensíveis.
- Validar a API com educadores e pesquisadores, incorporando feedback para melhorias.

2.1.3 Requisitos

Os requisitos essenciais do projeto foram definidos separadamente em dois quadros. O primeiro (Quadro 1) define os requisitos funcionais, ou seja, o que a API deve ser capaz de realizar como mecânica programada. Já o segundo (Quadro 2) define o que a API deve ser capaz de sustentar, mesmo não indicando essa capacidade como função realizável mecanicamente.

Quadro 1 – Requisitos Funcionais.

ID	Requisitos F	História do Usuário	Descrição
1	CRUD de Dados	Como usuário (administrador), eu quero cadastrar, ler, editar e remover/deletar informações sobre os povos originários, línguas e terminologias.	Cadastro e Atualização: Permitir o cadastro, edição e remoção de informações. Consulta Avançada: Implementar opções que possibilitem buscas de dados.
2	Endpoints Especializados	Como usuário, eu quero a partir do domínio, realizar consultas específicas para obter informações sobre aspectos culturais, línguas, terminologias... de uma determinada comunidade.	Endpoints Temáticos: Criar endpoints dedicados para consultas específicas. Endpoints de Relacionamento: Disponibilizar informações correlacionadas, como a relação entre línguas e povos, ou entre terminologias e contextos culturais.
3	Autenticação e Autorização	Como usuário, eu quero a garantia de segurança dos meus dados através de um sistema de autenticação.	Controle de Acesso: Implementar mecanismo de autenticação (com níveis de acesso: administrador, pesquisador, usuário comum, etc.) garantindo que apenas usuários autorizados realizem operações de escrita ou acessem dados sensíveis.
4	Interoperabilidade e Integração	Como usuário, eu quero integrar o sistema com outras plataformas, como portais acadêmicos.	APIs de Terceiros: Oferecer endpoints que permitam a integração com sistemas externos através de protocolos e padrões amplamente aceitos (REST, Json, etc).
5	Guias de Uso	Como usuário, eu quero visualizar tutoriais e exemplos de uso para me orientar durante o uso da ferramenta.	Tutoriais e Exemplos: Disponibilizar exemplos de chamadas e casos de uso práticos para orientar a integração.
6	Validação e Feedback	Como usuário, eu quero utilizar um canal de comunicação para reportar problemas com a qualidade e a pertinência dos dados, assim como a sugestão de melhorias, se necessário.	Validação: Realizar validações com educadores e pesquisadores para garantir a qualidade e a pertinência dos dados. Feedback: Disponibilizar um canal (ou endpoint) para que os usuários possam reportar problemas e sugerir melhorias.

Fonte: Autoria Própria (2025).

O Quadro 1 apresenta os requisitos funcionais da API listados na coluna 2 (Requisitos F). A terceira coluna (História do Usuário) exibe uma breve história que reflete como o usuário utiliza ou gostaria de utilizar a funcionalidade. Já a quarta coluna (Descrição) descreve o que deve ser realizado a partir disso.

Quadro 2 – Requisitos Não Funcionais.

ID	Requisitos NF	Descrição
1	Desempenho e Escalabilidade	Baixa Latência: Garantir que as respostas da API sejam rápidas e eficientes, mesmo em situações de alta demanda. Escalabilidade: Projetar a arquitetura de forma a suportar o aumento do volume de requisições e dados, possibilitando a expansão conforme necessário. Compatibilidade com Diversos Dispositivos: Garantir que a API possa ser consumida por diferentes tipos de clientes (web, mobile, etc.) sem prejuízo na performance ou segurança.
2	Segurança	Proteção dos Dados: Implementar criptografia (em repouso e em trânsito) para proteger informações sensíveis e garantir a conformidade com legislações de proteção de dados.
3	Usabilidade e Acessibilidade	Consistência na API: Manter uma estrutura de endpoints consistente e intuitiva para facilitar a integração por desenvolvedores e educadores.
4	Manutenibilidade	Código Modular: Estruturar o código de forma modular e documentada para facilitar a manutenção e futuras atualizações.
5	Confiabilidade e Disponibilidade	Alta Disponibilidade: Projetar a infraestrutura para minimizar o tempo de inatividade. Backup e Recuperação: Estabelecer políticas de backup e recuperação de dados para evitar perdas de informações.
6	Conformidade e Padrões	Aderência a Normas: A API deve estar em conformidade com as normativas e padrões aplicáveis, como LGPD ou outros regulamentos relacionados à proteção de dados e direitos culturais.

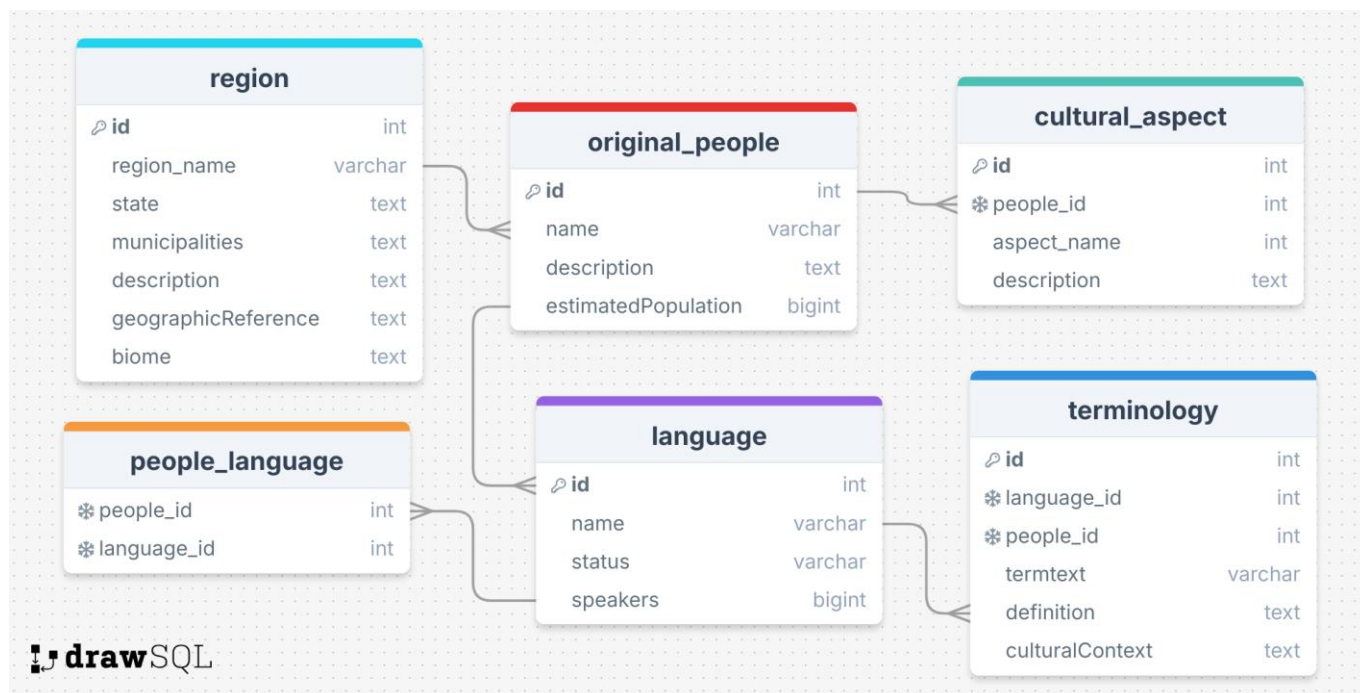
Fonte: Autoria Própria (2025).

O Quadro 2 apresenta os requisitos não funcionais da API listados na coluna 2 (Requisitos NF). A terceira coluna (Descrição) descreve o que deve ser desenvolvido para o bom funcionamento e manutenção geral do sistema, envolvendo boas práticas de desenvolvimento e segurança.

2.1.4 Estrutura do Banco de Dados

A modelagem visual do banco de dados foi feita a partir da ferramenta de design de banco de dados DrawSQL, que permitiu que criássemos diagramas de esquemas de entidades de forma prática e colaborativa, conforme na Figura 1. Ela ajudou a projetar os quadros e relacionamentos essenciais previstos para a API.

Figura 1 – Modelo da Estrutura do Banco de Dados.



Fonte: Autoria Própria (2025).

A Figura 1 apresenta como os módulos se relacionam, demonstrando seus vínculos. É também exibido suas características e atributos individuais. Cada atribuição (exceto o id), é um dado que pode ser visualizado na API, desde que haja o seu cadastro.

2.2 Estruturação do Banco de Dados

Para o gerenciamento de dados, a tecnologia selecionada foi o Prisma. Trata-se de um ORM (Object-Relational Mapping) de código aberto. Para contextualizar, a Amazon Web Services (2024) define o ORM como o processo de abstrair a conexão entre os objetos do código e o banco de dados, criando uma camada de software que converte os dados automaticamente. Isso permite a criação de aplicações modulares e fáceis de manter, sem que o programador precise lidar com os detalhes complexos das tabelas subjacentes.

Aplicando esse conceito, o Prisma se destaca por oferecer segurança de tipagem (type safety) e alta performance na comunicação com bancos de dados como o PostgreSQL (Prisma, 2025b). A escolha dessa ferramenta acelerou o desenvolvimento, garantindo uma estrutura de banco robusta e pronta para integração com a API. O funcionamento é prático: conforme descrito na documentação oficial (Prisma, 2025a), todo o trabalho começa em um arquivo de esquema (schema). Nesse arquivo, define-se o formato dos dados de maneira simples, e a própria ferramenta se encarrega de criar as tabelas no banco e gerar o código (o Prisma Client) que a aplicação usará para buscar e salvar as informações.

O banco de dados aplicado foi o PostgreSQL. Segundo a documentação (PostgreSQL, 2025), trata-se de uma solução open source robusta que combina a linguagem SQL com recursos avançados, garantindo o armazenamento seguro e a escalabilidade necessária para operações complexas.

2.3 Backend da API

O backend do projeto foi desenvolvido com o framework NestJS, utilizando também algumas funcionalidades do framework Express para preencher lacunas simples no desenvolvimento. No entanto, a ênfase está na tecnologia NestJS, o centro

do desenvolvimento, que por sua vez proporcionou um código padronizado e de boa manutenção. De acordo com Johnson (1997), frameworks funcionam fundamentalmente como uma técnica de reutilização orientada a objetos, compartilhando princípios comuns a outras estratégias de reaproveitamento de software. Conforme a documentação oficial (NestJS, 2025), o NestJS é um framework progressivo voltado para a construção de aplicações backend em Node.js eficientes e escaláveis. A ferramenta se destaca por oferecer suporte nativo ao TypeScript e por combinar elementos de Programação Orientada a Objetos (OOP), Programação Funcional (FP) e Programação Reativa Funcional (FRP).

A linguagem de programação escolhida para o desenvolvimento foi o TypeScript, já que sua tipagem oferece maior segurança para o desenvolvimento lógico da aplicação. Segundo a Microsoft (2025), o TypeScript é uma linguagem fortemente tipada baseada em JavaScript, projetada para oferecer ferramentas aprimoradas para o desenvolvimento em qualquer escala.

2.3.1 Ferramenta de Testes Insomnia

O desenvolvimento do backend exigiu uma interface capaz de realizar testes com os dados da API. Por isso, utilizamos uma ferramenta de auxílio essencial: o Insomnia. Conforme a documentação (Insomnia, 2025), o Insomnia é uma aplicação desktop de código aberto projetada para simplificar o design, a depuração e os testes de APIs. A ferramenta oferece recursos essenciais para o desenvolvimento, como auxiliares de autenticação, geração de código e gestão de variáveis de ambiente. Essa aplicação possibilitou o acesso a métodos envolvendo manipulação de dados, validando esses processos antes da evolução do código.

2.4 Frontend da API

O frontend da API foi desenvolvido com as tecnologias Next.js e React.js, as quais foram especialmente criadas para o desenvolvimento frontend. O Next.js é o responsável por permitir definir cada arquivo como rota localizada na pasta de aplicação no frontend. De acordo com a documentação (Vercel, 2024), o Next.js é um framework React voltado para a construção de aplicações web full-stack. Ele permite utilizar componentes React para criar interfaces de usuário, enquanto fornece recursos adicionais e otimizações de desempenho essenciais para o projeto. Já o React.js baseia-se no conceito de componentes. Conforme a documentação (React, 2025), trata-se de unidades da interface que possuem lógica e visual próprios, permitindo a construção desde pequenos elementos isolados até estruturas completas de página. Com isso, foi construída a estrutura visual do projeto.

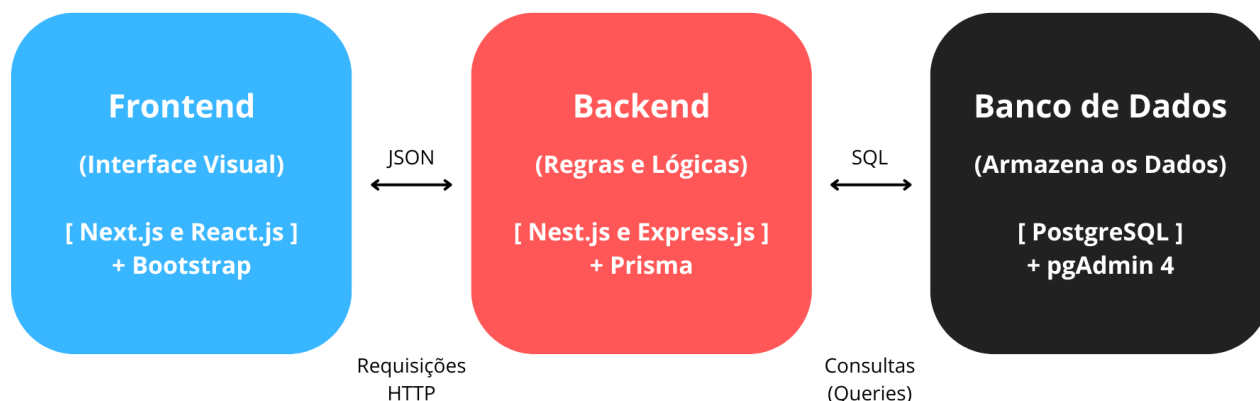
O Bootstrap foi a ferramenta de estilização selecionada para aprimorar o design visual da API no frontend. Ele é definido em sua documentação (Bootstrap, 2023) como um conjunto de ferramentas frontend poderoso e rico em recursos, servindo como base para a estilização do sistema. Através dele, aceleramos o desenvolvimento visual com a adição de recursos prontos para o nosso sistema.

Por fim, a linguagem de programação escolhida para o desenvolvimento também foi o TypeScript, considerando os mesmos motivos pelo qual foi utilizado no Backend.

2.5 Diagrama de Relacionamento

Uma ilustração de relacionamento ajuda a entender melhor o conjunto frontend, backend e banco de dados do projeto. Para isso, nós temos a seguinte Figura 2, descrevendo a ação do cliente no frontend para o servidor/API no backend.

Figura 2 – Diagrama de Relacionamento.



Fonte: Autoria Própria (2025).

Funcionamento do Diagrama:

1. O usuário acessa o Frontend (Next.js).
2. O Frontend envia um pedido (ex: "Buscar as terminologias") para o Backend.
3. O Backend (Nest.js) processa o pedido e usa o Prisma para falar com o Banco de Dados.
4. O Banco de Dados (PostgreSQL) devolve os dados brutos.
5. O Backend envia esses dados formatados (em JSON) de volta para o Frontend mostrar na tela de forma organizada.

2.6 Github: Controle de Versão e Armazenamento em Nuvem

Para o controle de versões e hospedagem do código, utilizou-se o GitHub. A ferramenta funciona como uma plataforma em nuvem para armazenamento e colaboração entre desenvolvedores (GitHub, 2025).

2.7 Documentação Técnica

A documentação técnica da API foi desenvolvida de forma contínua e interativa, acompanhando a metodologia ágil Kanban adotada no projeto. Mais do que um registro de metas e resultados, o documento abrange o levantamento detalhado de requisitos funcionais e não funcionais, a modelagem do banco de dados (incluindo diagramas de entidade-relacionamento) e a descrição da arquitetura do sistema. Além disso, foi elaborado um guia de uso voltado para educadores e desenvolvedores, assegurando a manutenção do código, boa usabilidade e a facilidade de integração futura do sistema por terceiros.

3. Resultados e Discussão

O principal resultado deste trabalho foi o desenvolvimento e a validação bem-sucedida da API Educacional "TECENDO SABERES", uma ferramenta tecnológica estruturada para a preservação e disseminação da cultura dos povos originários. A implementação seguiu o cronograma combinado, cumprindo os requisitos funcionais de cadastro, leitura, atualização e criação (CRUD).

Existem várias dificuldades no ensino indígena sendo uma delas a distância entre aldeias e os locais de ensino e, no caso de ensino superior e, mesmo no ensino em outros níveis educacionais, como consideram Silva, Scheibel e Shitsuka (2020). Acreditamos que a API que desenvolvemos, pode apoiar o processo de ensino em todos os níveis educacionais tanto da educação

básica quanto do nível superior, uma vez que por meio de recursos eletrônicos pode-se incentivar o aprendizado a distância, o auto aprendizado e, o aprendizado ativo. Por meio da API pode-se aproximar os conteúdos da BNCC (Brasil, 2018) dos estudantes indígenas e, também fazer parte da inclusão digital reconhecida pela UNESCO (2021).

3.1 Infraestrutura e Backend

A arquitetura do sistema foi consolidada utilizando o framework NestJS em conjunto com Express.js, operando sobre um ambiente Node.js. Esta escolha garantiu a modularidade do código e facilitou a escalabilidade da aplicação. Para a persistência dos dados, foi implementado um banco de dados relacional PostgreSQL, gerenciado através do ORM Prisma.

A modelagem do banco de dados resultou em um esquema relacional eficiente, composto por entidades interligadas: *indigenous_people* (povo), *language* (língua), *terminology* (terminologia), *cultural_aspect* (aspecto cultural) e *region* (região). Essa estrutura permitiu não apenas o armazenamento dos dados, mas a preservação do contexto significativo de cada termo, associando-o à sua origem étnica e geográfica.

3.2 Interface de Uso (Frontend)

Para validar o consumo da API e permitir a interação de usuários, foi entregue uma interface web utilizando Next.js, React.js e estilização com Bootstrap. A interface permite a visualização clara dos dados e oferece formulários intuitivos para administradores gerenciarem o conteúdo. A implementação deste frontend é importante para a utilização, identificação de bugs e para a validação da experiência do usuário, resultando em possíveis otimizações como a separação dos componentes de formulário para arquivos independentes, melhorando a organização e desempenho do código.

3.2.1 Visualização Geral e Individual da API

As figuras numeradas de 3 a 5 abaixo mostram os resultados do desenvolvimento da Aplicação Web, referente ao Design visual e exibição geral dos dados no frontend. Todo e qualquer cadastro de termos, povos originários, línguas, etc, foram temporariamente cadastrados para apresentação de funcionalidades.

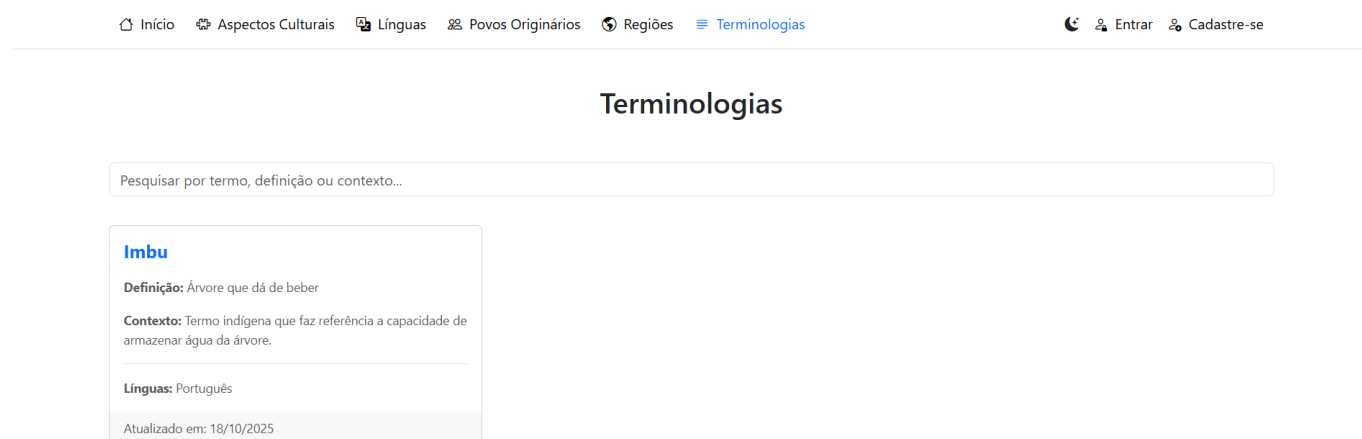
Figura 3 – Página Inicial da API.



Fonte: Autoria Própria (2025).

A página inicial do projeto conta com algumas informações básicas sobre a API, seus objetivos, e também consta com alguns links para a documentação, guia de uso e o GitHub do projeto.

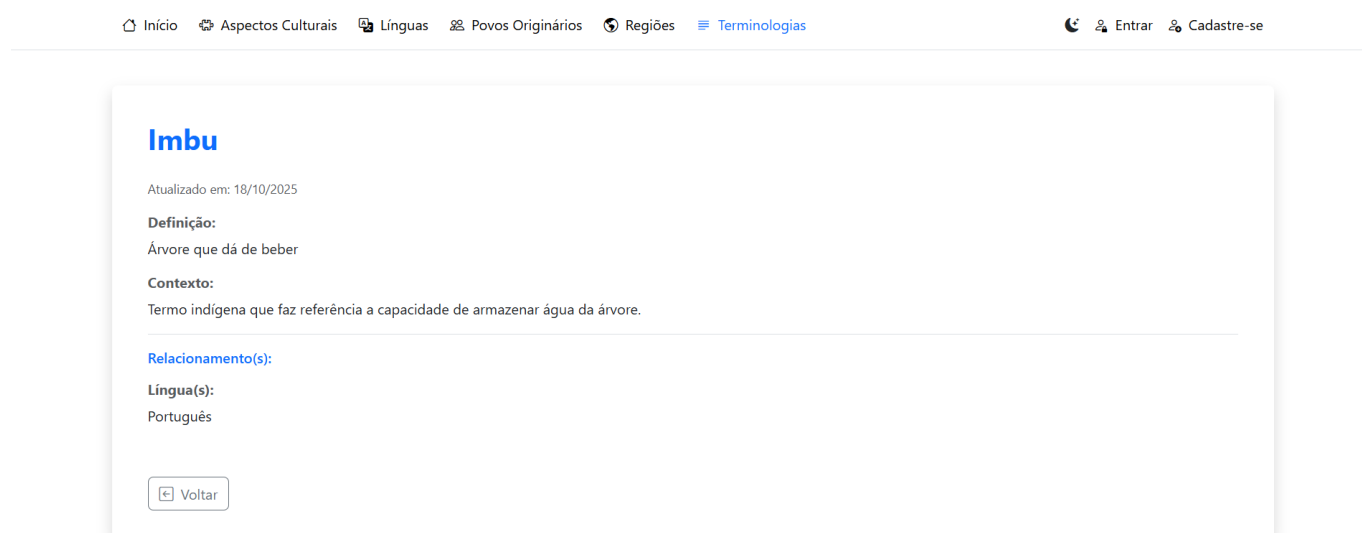
Figura 4 – Página de Terminologias.



Fonte: Autoria Própria (2025).

A página da Figura 4 é o lugar onde todas as terminologias cadastradas no sistema devem ser exibidas, exatamente como no formato mostrado na imagem. Cada cartão possui uma breve descrição dos seus atributos, e cada nome é um link que redireciona o usuário para a página da terminologia em questão. Ao acessar, é possível visualizar os dados completamente.

Figura 5 – Página de Terminologia por ID.



Fonte: Autoria Própria (2025).

Essa figura demonstra o acesso do usuário a alguma terminologia específica, tendo assim uma visão geral. Essa ação se aplica também a todas as outras entidades já mencionadas, como línguas, povos originários, etc.

3.2.2 Formulário de Edição e Criação

As figuras numeradas de 6 a 10 representam os formulários de Criação e Edição de uma entidade e as suas características. O acesso e gerenciamento dos dados só é possível através de contas com a função administrador.

Figura 6 – Formulário de Edição da Página de Terminologias.

Formulário de Edição da Página de Terminologias. O formulário contém campos para: Termo (Imbu), Definição (Árvore que dá de beber), Contexto (Termo indígena que faz referência a capacidade de armazenar água da árvore.), Vincular Língua(s) (Opcional) (Português selecionado) e Vincular Povo(s) Originário(s) (Opcional) (Pankararu selecionado). Botões: Cancelar, Salvar Alterações.

Fonte: Autoria Própria (2025).

A Figura 6 apresenta o formulário de Edição de uma determinada terminologia, sendo possível alterar nome e as suas características individuais, assim como os seus vínculos, se houver.

Figura 7 – Formulário de Criação da Página de Línguas.

Formulário de Criação da Página de Línguas. O formulário contém campos para: Nome (Ex: Pankará...), Atividade da língua (Ativa, Inativa, Não sei), Nº de falantes (Estimativa) (0), Vincular Povo(s) Originário(s) (Opcional) (Pankararu selecionado) e Vincular Terminologia(s) (Opcional) (Imbu selecionado). Botões: Cancelar, Cadastrar.

Fonte: Autoria Própria (2025).

A Figura 7 apresenta o formulário de Criação para registro de novas Línguas. Assim, é possível nomear e informar as suas características individuais, assim como os seus vínculos, se houver.

Figura 8 – Formulário de Criação da Página de Povos Originários.

Fonte: Autoria Própria (2025).

A Figura 8 apresenta o formulário de Criação para registro de novos Povos Originários. Assim, é possível nomear e informar as suas características individuais, assim como os seus vínculos, se houver. Como todos os módulos giram em torno de algum povo originário, nota-se que essa é a entidade com a possibilidade de ter todos os vínculos. O formulário foi alinhado para exibir todos os campos.

Figura 9 – Formulário de Criação da Página de Regiões.

Fonte: Autoria Própria (2025).

A Figura 9 apresenta o formulário de Criação para registro de novas Regiões. Assim, é possível nomear e informar as suas características individuais, assim como os seus vínculos, se houver. O formulário foi alinhado para exibir todos os campos.

Figura 10 – Formulário de Criação da Página de Aspectos Culturais.

A interface do sistema apresenta uma barra de navegação superior com links para Início, Aspectos Culturais, Línguas, Povos Originários, Regiões e Terminologias. No canto superior direito, há uma saudação 'Olá, Administrador'. O título principal da seção é 'Gerenciar Aspectos Culturais'. O formulário em foco, intitulado 'Novo Aspecto Cultural', contém os seguintes elementos:

- Nome:** Campo de texto com o exemplo 'Ex: Literatura de cordel...'.
- Descrição:** Campo de texto com o exemplo 'Características do aspecto cultural...'.
- Vincular Povo(s) Originário(s) (Opcional):** Seção que inclui um campo de busca 'Pesquisar povo...' e uma lista de 'Povos Originários' com a opção 'Pankararu' selecionada.
- Botões:** 'Cancelar' (em vermelho) e 'Cadastrar' (em azul) no canto inferior direito.

Fonte: Autoria Própria (2025).

A figura 10 apresenta o formulário de Criação para registro de novos Aspectos Culturais. Assim, é possível nomear e informar as suas características individuais, assim como os seus vínculos, se houver.

3.2.3 Login, Cadastro e Gerenciamento de Usuários

Finalizando os resultados da API, segue as figuras que apresentam os resultados do desenvolvimento dos campos de login e cadastro (Figuras 11 e 12), assim como o gerenciamento dessas contas (Figuras 13 e 14).

Figura 11 – Página de Login.

A interface da página de login possui uma barra de navegação superior idêntica à da Figura 10. O formulário centralizado, intitulado 'Fazer Login', contém:

- Email:** Campo de texto com o placeholder 'Endereço de email'.
- Senha:** Campo de texto com o placeholder 'Sua senha de 8 dígitos'.
- Botão ENTRAR:** Botão azul destacado.
- Links:** 'Esqueci a senha' (com ícone de lupa) e 'Cadastre-se' (em azul).
- Texto:** 'Não está registrado?' seguido pelo link 'Cadastre-se'.

Fonte: Autoria Própria (2025).

A página de login possui um formulário simples com dois campos, o primeiro exige o endereço de email da conta do usuário, e o segundo campo pede a senha de acesso. Logo abaixo, possui um link para troca de senha e um outro link de redirecionamento para cadastro de nova conta.

Figura 12 – Página de Cadastro de Usuário.

A interface da página de cadastro de usuário apresenta um formulário centralizado com o título "Criar Conta". O formulário possui os seguintes campos: "Nome" com o placeholder "Seu nome de usuário"; "Email" com o placeholder "Endereço de email"; "Data de nascimento" com o placeholder "dd/mm/aaaa" e um ícone de calendário; e "Senha" com o placeholder "Sua senha de 8 dígitos". Abaixo dos campos, há um botão azul "REGISTRAR" e um link "Já possui uma conta? Faça login". No topo da página, há uma barra de navegação com links: "Início", "Aspectos Culturais", "Línguas", "Povos Originários", "Regiões", "Terminologias", "Entrar" e "Cadastre-se".

Fonte: Autoria Própria (2025).

O cadastro de usuário é bem simples, o usuário apenas vai inserir o seu nome de usuário, endereço de email, data de nascimento e uma senha de 8 dígitos. Após a inserção dos dados, basta apenas registrar. Abaixo possui um link de redirecionamento para a página de login, caso o cliente já possua uma conta.

Figura 13 – Página de Gerenciamento da Conta.

A interface da página de gerenciamento de conta, intitulada "Minha Conta", exibe os dados do usuário atualizado. Os campos "Nome de usuário" e "Email" contêm os valores "Administrador" e "admin@tecendosaberes.com", respectivamente. O campo "Cargo" mostra "Administrador" em um botão. Abaixo, a seção "Alterar Senha" inclui uma mensagem de aviso: "Preencha o campo abaixo apenas se desejar alterar sua senha." e um campo "Nova senha". No canto inferior direito, há dois botões: "Cancelar" (em vermelho) e "Salvar Alterações" (em azul). A barra de navegação no topo é idêntica à da Figura 12, com o nome de usuário "Olá, Administrador" exibido.

Fonte: Autoria Própria (2025).

Essa página permite que o próprio usuário reescreva seu nome de exibição, troque o seu endereço de email ou altere a sua senha. A conta administradora não pode alterar o seu próprio cargo administrativo, isso exige uma outra conta com essa função.

Figura 14 – Página de Gerenciamento de Usuários para Administradores.



ID	Nome de Usuário	Email	Cargo	Ações
2	Sebastian	sebastian2dgir@gmail.com	ADMIN	✎ 🔑 🗑
3	Cleide	ahilda@gmail.com	USER	✎ 🔑 🗑
1	Administrador	admin@tecendosaberes.com	ADMIN	✎ 🔑 🗑

Fonte: Autoria Própria (2025).

O quadro da Figura 14 mostra uma lista dos usuários cadastrados. Através dessa página, o administrador pode adicionar uma nova conta ao banco de dados, editar os dados e resetar a senha de uma conta existente, assim como deletar algum perfil, caso haja necessidade.

3.3 Discussão: Acessibilidade e Decisões de Arquitetura

Durante o ciclo de desenvolvimento, a análise dos requisitos levou a uma mudança estratégica na regra de negócio da aplicação. Inicialmente, o sistema previa autenticação via Json Web Token (JWT) para todas as operações, exigindo login tanto para administradores quanto para usuários comuns. No entanto, observou-se que a obrigatoriedade de login para a simples consulta de dados criava uma barreira de acesso, dificultando a integração da API com outros portais de pesquisa e plataformas educacionais. Diante disso, optou-se por remover a restrição de autenticação para as requisições de leitura (GET). A autenticação JWT foi mantida estritamente para as operações de escrita (POST, PUT, DELETE), garantindo a segurança e integridade dos dados sem comprometer a democratização do acesso à informação.

Adicionalmente, o processo de validação permitiu a correção de falhas de usabilidade, como a correção de status de línguas e ajustes em campos de pesquisa que não respondiam adequadamente em determinadas páginas.

3.4 Limitações

Embora a arquitetura tenha sido projetada para alta performance, os testes de carga planejados com ferramentas como Loader.io não puderam ser executados em sua fase, devido à dependência de publicação da aplicação em um domínio digital, visto que o desenvolvimento foi realizado de forma local. Contudo, a utilização de tecnologias leves para o desenvolvimento web (Nest.js, Express.js, Next.js, React.js...), por si só já fornece uma base sólida para suportar a demanda prevista em trabalhos futuros.

4. Considerações Finais

O desenvolvimento da API Educacional "TECENDO SABERES" alcançou com êxito seu objetivo principal de criar uma ponte tecnológica entre a preservação cultural e a educação digital. A aplicação resultante apresenta-se não apenas como

um repositório de dados, mas como uma ferramenta estruturada capaz de dar visibilidade e organização às línguas e terminologias dos povos originários brasileiros.

Do ponto de vista técnico, a escolha de desenvolver em um modelo moderno baseado em Node.js (Nest.js e Express) e PostgreSQL provou-se acertada, garantindo uma arquitetura modular, escalável e de fácil manutenção. A implementação bem-sucedida das operações de CRUD e a modelagem relacional do banco de dados asseguram que as informações sejam armazenadas respeitando a complexidade e o contexto cultural de cada etnia.

A decisão de remover a obrigatoriedade de login para a consulta pública de dados é uma contribuição relevante que reflete o compromisso ético do projeto com a Ciência Aberta e a democratização do conhecimento, facilitando a integração futura com outras plataformas acadêmicas e educacionais.

Além do sistema, o projeto entrega um conjunto documental, incluindo guia de uso para educadores e especificações técnicas para desenvolvedores através da documentação, o que garante a longevidade e a possibilidade de evolução da ferramenta.

Como trabalhos futuros, recomenda-se a ampliação da validação da API em contextos educacionais reais, especialmente junto a educadores e comunidades indígenas, bem como a integração da plataforma com repositórios educacionais e ambientes virtuais de aprendizagem. Tais iniciativas poderão fortalecer ainda mais o impacto social e pedagógico da ferramenta, ampliando seu alcance e consolidando sua contribuição para a preservação e valorização dos saberes indígenas no ambiente digital.

Referências

- Amazon Web Services. (2025). O que é mapeamento objeto-relacional (ORM)? <https://aws.amazon.com/pt/what-is/object-relational-mapping/>
- Azevedo, C. B. (2018). Metodologia científica ao alcance de todos. Editora Manole. ISBN-13: 978-8520456385.
- Bootstrap. (2023). Get started with Bootstrap. [Getbootstrap.com. https://getbootstrap.com/docs/5.3/getting-started/introduction/](https://getbootstrap.com/docs/5.3/getting-started/introduction/)
- Brasil. (2008, 10 de março). Lei nº 11.645, de 10 de março de 2008. Altera a Lei nº 9.394, de 20 de dezembro de 1996, modificada pela Lei nº 10.639, de 9 de janeiro de 2003, para incluir no currículo oficial da rede de ensino a obrigatoriedade da temática “História e Cultura Afro-Brasileira e Indígena”. Presidência da República. https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/111645.htm
- Brasil. Ministério da Educação. (2024, 15 de maio). Portaria nº 470, de 14 de maio de 2024. Institui a Política Nacional de Equidade, Educação para as Relações Étnico-Raciais e Educação Escolar Quilombola – PNEERQ. Diário Oficial da União, Seção 1 - Extra B, p. 1. <https://www.in.gov.br/en/web/dou/-/portaria-n-470-de-14-de-maio-de-2024-559544343>
- Brasil. Ministério da Educação. (2018). Base Nacional Comum Curricular. https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC_EI_EF_110518_versaofinal.pdf
- Gil, A. C. (2017). Como elaborar um projeto de pesquisa. Editora Atlas.
- GitHub. (2025). GitHub e o Git. GitHub Docs. <https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>
- IEEE Computer Society. (2025). Guide to the Software Engineering Body of Knowledge (SWEBOK Guide) (Version 4.0a). IEEE. <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- Insomnia (2025). Kong Docs. <https://developer.konghq.com/insomnia/>
- Johnson, R. E. (1997). Frameworks = (components + patterns). Communications of the ACM, 40(10), 39–42. <https://doi.org/10.1145/262793.262799>
- Kanban University. (2020). The Official Kanban Guide. <https://kanban.university/kanban-guide/>
- Microsoft. (2025). TypeScript - JavaScript that scales. [Typescriptlang.org. https://www.typescriptlang.org/](https://www.typescriptlang.org/)
- NestJS. (2025). NestJS - A progressive Node.js framework. <https://docs.nestjs.com/>
- Pereira, A. S. et al. (2018). Metodologia da pesquisa científica. (Free ebook). Santa Maria. Editora da UFSM.
- PostgreSQL. (2025). PostgreSQL: About. [Postgresql.org. https://www.postgresql.org/about/](https://www.postgresql.org/about/)
- Prisma. (2025a). Getting started. Prisma. <https://www.prisma.io/docs/getting-started>
- Prisma. (2025b). What is Prisma? (Overview). <https://www.prisma.io/docs/orm/overview/introduction/what-is-prisma>

React. (2025). React.dev. <https://pt-br.react.dev/learn>

Silva, J. A. C.; Scheibel, M. F. & Shitsuka, R. (2020). Licenciatura indígena na formação de professores no Estado do Acre. Editora Poisson. ISBN: 978-65-86127-57-7. DOI: 10.36229/978-65-86127-57-7.

Torres, F. E. (2021). Desenvolvimento de API REST. Editora Senac São Paulo.

UNESCO. (2021). Plano de ação global da Década Internacional das Línguas Indígenas. <https://en.unesco.org/idi2022-2032/global-action-plan>

Vercel. (2024). Getting Started | Next.js. Nextjs.org. <https://nextjs.org/docs>

Vidal, A. S., & Miguel, J. R. (2020). As Tecnologias Digitais na Educação Contemporânea /Digital Technologies in Contemporary Education. ID on Line Revista de Psicologia, 14(50), 366–379. <https://doi.org/10.14295/online.v14i50.2443>