Insper

DESENVOLVIMENTO DE PLATAFORMA EXPERIMENTAL DE TECLADO VIRTUAL PARA LÍNGUAS INDÍGENAS EM ANDROID

Guilherme Augusto Chaves de Carvalho (guilhermeacc@al.insper.edu.br)
Guilherme Carneiro Lunetta (guilhermecl1@al.insper.edu.br)
José Rafael Martins Fernandes (josermf2@al.insper.edu.br)
Rafael Evangelista Monteiro (rafalem2@al.insper.edu.br)

Trabalho de Conclusão de Curso

Relatório
Versão Final
do Projeto Final de Engenharia

São Paulo-SP
Novembro de 2023

Guilherme Augusto Chaves de Carvalho
Guilherme Carneiro Lunetta
José Rafael Martins Fernandes
Rafael Evangelista Monteiro

DESENVOLVIMENTO DE PLATAFORMA EXPERIMENTAL DE TECLADO VIRTUAL PARA LÍNGUAS INDÍGENAS EM ANDROID

Relatório Final do Projeto Final de Engenharia

Relatório apresentado ao curso de Engenharia, como requisito para o Trabalho de Conclusão de Curso.

Professor Orientador: Prof. Dra. Graziela Simone Tonin

Mentor na Empresa: Julio Nogima

Coordenador TCC/PFE: Prof. Me. Raul Ikeda Gomes da Silva

São Paulo - SP
Novembro de 2023

Guilherme Augusto Chaves de Carvalho
Guilherme Carneiro Lunetta
José Rafael Martins Fernandes
Rafael Evangelista Monteiro

DEVELOPMENT OF EXPERIMENTAL PLATFORM OF VIRTUAL KEYBOARD FOR INDIGENOUS LANGUAGES IN ANDROID

Final Report on Final Engineering Project

Undergraduate Capstone Design Project Report submitted to the Engineering program (Computer/Mechanical/Mechatronic) in partial fulfillment of the requirements for the Bachelor's Degree in Engineering.

Advisor: Prof. Dra. Graziela Simone Tonin

Company Mentor: Julio Nogima

Course Coordinator TCC/PFE: Prof. Me. Raul Ikeda Gomes da Silva

São Paulo - SP
November of 2023

Guilherme Augusto Chaves de Carvalho
Guilherme Carneiro Lunetta
José Rafael Martins Fernandes
Rafael Evangelista Monteiro

DESENVOLVIMENTO DE PLATAFORMA EXPERIMENTAL DE TECLADO VIRTUAL PARA LÍNGUAS INDÍGENAS EM ANDROID

Projeto Final de Engenharia apresentado ao programa Graduação em Engenharia (Computação/Mecânica/Mecatrônica) como requisito parcial para a obtenção do título de Bacharel em Engenharia.

Orientador: Prof. Dra. Graziela Simone Tonin

Banca Examinadora

_____

Graziela Simone Tonin

Insper

_____

Maciel Calebe Vidal

Insper

_____

Luciano Silva

Insper

Sumário

RESUMO **6**

ABSTRACT **7**

1. INTRODUÇÃO **8**

1.1 ESCOPO DO PROJETO 9
1.2 RECURSOS 10
1.3 CRONOGRAMA 10
1.4 MAPEAMENTO DOS STAKEHOLDERS 12
1.5 RISCOS ENVOLVIDOS 13
1.6 QUESTÕES ÉTICAS E PROFISSIONAIS 14
1.7 REVISÃO DO ESTADO DA ARTE 14

1.7.1 TECNOLOGIAS E GESTÃO DE PROJETO 16

2. METODOLOGIA **17**

2.1 APLICAÇÃO WEB 21
2.2 APLICAÇÃO MOBILE 25
2.3 COLETA DE DADOS SOBRE O PROJETO 28
2.4 ANÁLISE DOS DADOS COLETADOS 31

3. RESULTADOS **32**

4. CONCLUSÕES E TRABALHOS FUTUROS **38**

REFERÊNCIAS **40**

RESUMO

A UNESCO oficializou o período de 2022 a 2032 como a Década Internacional das Línguas Indígenas, com o propósito de promover a reflexão sobre as ameaças enfrentadas por essas línguas e, principalmente, incentivar ações com o intuito de preservá-las e fortalecê-las. Nesse contexto, este projeto tem como principal objetivo o desenvolvimento de um aplicativo de teclado para Android que ofereça suporte à escrita de mais de 40 línguas indígenas. O projeto visa assistir a população índigena brasileira no auxílio à escrita de suas línguas nativas, a fim de contribuir com a acessibilidade, preservação cultural e aumento de registros escritos. Para isso, foi realizada uma parceria com a IBM Research que em conjunto com linguistas da USP (Universidade de São Paulo), possui uma série de estudos relacionados a essas línguas originárias, como o desenvolvimento de modelos de *machine learning* para traduções e completude de sentenças, baseados em bíblias traduzidas na época de colonização e catequização dos povos nativos brasileiros. Para desenvolvimento da solução, gestão do projeto e organização da equipe, utilizou-se o *framework* ágil *Scrum* e alinhamento semanais foram realizados com o cliente. Como protótipo foi desenvolvido um aplicativo em Flutter, um *framework open source*, que disponibilizará acesso à teclados com todos os caracteres presentes em cada uma das línguas. Adicionalmente, foi desenvolvida uma aplicação web que tornou possível aos gestores mapear os caracteres no teclado, facilitando a digitação e o uso do aplicativo pelos usuários finais. A versão final do protótipo foi testada e validada pelo representante do cliente, melhorias e sugestões futuras de evolução foram adicionadas ao projeto a partir deste testes

Palavras-chave: Aplicativo Android, Línguas Indígenas, Teclado virtual, *Framework Open Source,* Aplicação Web

ABSTRACT

UNESCO has officially designated the period from 2022 to 2032 as the International Decade of Indigenous Languages, with the purpose of promoting reflection on the threats faced by these languages and, mainly, encouraging actions to preserve and strengthen indigenous languages. In this context, this project aims to develop an Android keyboard application that supports the writing of more than 40 indigenous languages. The project's main goal is to assist the Brazilian indigenous population in writing their native languages, contributing to accessibility, cultural preservation, and an increase in written records. In order to achieve it, a partnership was established with IBM Research Brazil, that along with linguists from USP (University of São Paulo), have a series of studies related to those indigenous communities and their languages, like the development of machine learning models on translation and sentence completion, based on translated bibles written in the colonization and catechization period. For the development of the solution, project management, and team organization, the agile Scrum framework was used, and weekly alignments were conducted with the client. As a prototype, a Flutter application, an open-source framework, was developed, providing access to keyboards with all the characters present in each language. Additionally, a web application was created, allowing to arrange characters on the keyboard, which facilitated typing, and enhanced the usability of the application for end users. The final version of the prototype was tested and validated by the client, and improvements and suggestions for future evolution were incorporated into the project based on these tests.

Keywords: Android Application, Indigenous Languages, Virtual Keyboard, Web Application, Open Source Framework

1. Introdução

A preservação das línguas indígenas brasileiras é uma missão vital para manter viva a rica diversidade cultural e linguística do país. No entanto, a era digital apresenta desafios significativos, especialmente no que diz respeito à falta de suporte para caracteres específicos dessas línguas nos teclados convencionais.

Dessa forma, o desafio central deste projeto surge da falta de suporte para caracteres específicos presentes nas línguas indígenas brasileiras, que não são contemplados por outros idiomas e, consequentemente, não estão incorporados nos teclados convencionais. Essas línguas têm um conjunto único de caracteres que foram cuidadosamente selecionados pela IBM. No entanto, a ausência de implementação desses caracteres nos padrões atuais cria um obstáculo expressivo para tradução e uso adequado da escrita nativa por parte dos usuários. Essa lacuna não apenas dificulta o aprendizado dessas línguas, mas também representa uma ameaça à sua preservação cultural. A falta de um teclado que englobe esses caracteres especiais pode resultar em dificuldades substanciais na comunicação digital, limitando a expressão escrita, o compartilhamento de conhecimento e a participação das comunidades indígenas no ambiente online. Dessa forma, a IBM Research Brasil, ao conduzir projetos anteriores em colaboração com as populações indígenas, destacou a necessidade de utilizar a tecnologia como ferramenta fundamental na preservação desse valioso patrimônio linguístico brasileiro.

Este projeto é uma parceria com a IBM, empresa da área de tecnologia que fabrica hardware e software, oferece serviços de infraestrutura, hospedagem, consultoria, e tem forte atuação na área de inteligência artificial, contando com uma equipe de mais de 20 mil especialistas em IA. Sendo assim, uma das principais empresas de tecnologia da informação, com um longo histórico de pesquisa e inovação desde o século XIX.

Os usuários finais deste projeto são as comunidades indígenas, especialistas em linguagem, professores e qualquer pessoa interessada em apoiar a preservação e o uso dessas línguas. O projeto tem como principal objetivo o desenvolvimento de um aplicativo de teclado para Android que ofereça suporte à escrita de mais de 40 línguas indígenas. O projeto visa assistir a população índigena brasileira no auxílio à escrita de suas línguas nativas.

Esse projeto oferece os seguintes retornos e ganhos:

- Preservação Cultural e Impacto Social: A preservação das línguas indígenas é um dos principais objetivos do projeto, e a disponibilidade de ferramentas modernas de escrita está destinada a causar um impacto social positivo. Ao oferecer essas ferramentas, o projeto permite que os membros das comunidades indígenas não apenas usem e ensinem suas línguas, mas também contribuam significativamente para um aumento na documentação escrita. Isso, por sua vez, fortalece a identidade cultural e histórica dessas comunidades, criando uma base sólida para a preservação desses patrimônios linguísticos. Essa iniciativa também contribui para uma maior conscientização e consideração da diversidade linguística do país.
- Pesquisa e Desenvolvimento: A pesquisa e desenvolvimento de ferramentas e aplicativos voltados a línguas em risco não apenas beneficiam diretamente as comunidades indígenas, mas também têm potencial para impactar positivamente outros contextos linguísticos. Ao criar técnicas e soluções eficientes para o uso digital de línguas específicas, esse projeto pode fornecer contribuições valiosas que transcendem fronteiras culturais, promovendo avanços mais amplos na pesquisa e desenvolvimento de tecnologias linguísticas.
- Potencial de Expansão: As soluções desenvolvidas nesse projeto podem servir como base para iniciativas semelhantes em outras línguas indígenas.

A viabilidade desse projeto é promissora em vários aspectos:

- Regulamentação: Não há barreiras regulatórias significativas para o desenvolvimento de um aplicativo de apoio à escrita de línguas indígenas.
- Viabilidade Financeira: A IBM, com seus recursos e compromissos, tem a capacidade financeira para desenvolver e apoiar o desenvolvimento do protótipo da solução.

1.1 Escopo do projeto

Esse projeto tem como principal objetivo o desenvolvimento de um aplicativo de teclado para Android que ofereça suporte à escrita de mais de 40 línguas indígenas. Faz parte do projeto o desenvolvimento do MVP (*minimum viable product*) que consiste em uma versão inicial de uma aplicação *mobile* e uma aplicação web, o uso do banco de dados *Cloudant* pertencente à IBM Cloud, a coleta de *feedbacks* do cliente, a consulta ao colaborador do projeto acerca de quaisquer dúvidas sobre o conteúdo do aplicativo e a criação de uma

experiência simples e eficiente para os usuários. O que não faz parte do projeto é desenvolver *APIs REST* com conexão a um banco de dados de nossa escolha, disponibilizar o aplicativo publicamente na Play Store e tomar decisões sobre o conteúdo do aplicativo sem consultar o *stakeholder*.

O custo do projeto é absorvido através da parceria com a IBM que disponibiliza acesso gratuito a tecnologias e nuvem, como o *Cloudant* no plano *Lite* e acesso ao serviço de banco de dados da IBM Cloud. As restrições do projeto envolvem a impossibilidade de usar ferramentas que não sejam de código aberto (*open source*), tanto o *framework* escolhido para desenvolver o aplicativo e o site, como eventuais bibliotecas usadas durante o desenvolvimento. Além disso, o serviço de banco de dados usado, devido a uma solicitação do cliente, foi o *Cloudant*.

Como entrega final, foi desenvolvido um protótipo de um aplicativo e de uma aplicação web de acordo com os requisitos solicitados e validados pela IBM, utilizando as melhores práticas de desenvolvimento, realizando todos os testes e ajustes necessários e atendendo aos ajustes solicitados pelo *stakeholder*. No final, espera-se que o projeto seja utilizado por essas comunidades e auxilie no estudo e desenvolvimento dessas mais de 40 línguas, contribuindo para que nosso importante patrimônio histórico seja perpetuado.

1.2 Recursos

A IBM solicitou que para o desenvolvimento da solução fossem utilizadas apenas ferramentas de código aberto (*open source*) a fim de evitar custos tecnológicos, visto que tais ferramentas possuem licenciamento livre para o uso do produto. Nesse caso, tanto os *frameworks* (Flutter e React) quanto suas bibliotecas seguem esse padrão, possibilitando seu uso. Além disso, o único recurso necessário para a realização do projeto é o serviço de banco de dados *Cloudant*, da IBM Cloud.

1.3 Cronograma

O cronograma do projeto é uma etapa fundamental para o planejamento e acompanhamento de todas as atividades. Nele, definimos as etapas que serão executadas ao longo do processo, garantindo que cada passo seja cumprido de maneira organizada e eficiente. É importante destacar que o relatório foi desenvolvido incrementalmente ao longo do projeto, refletindo as mudanças e os avanços à medida que ocorreram.

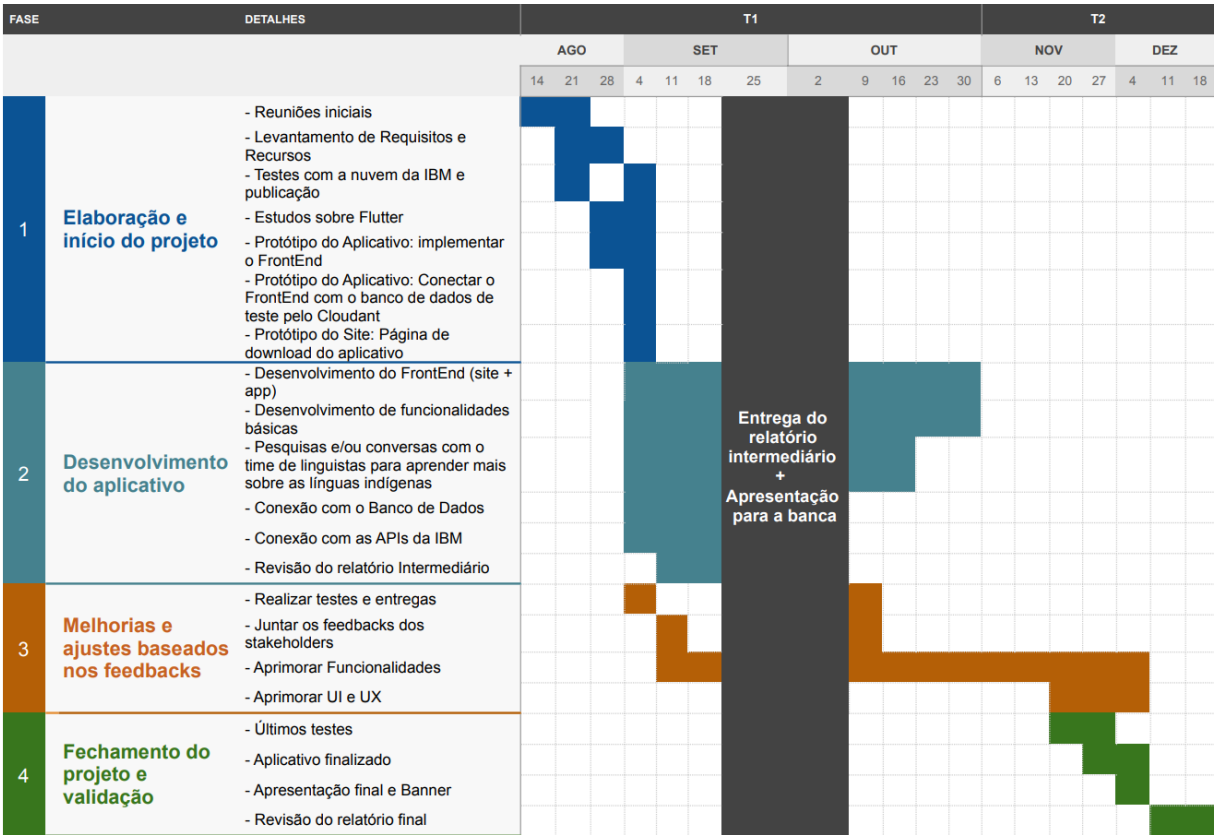
A elaboração e início do projeto foi o ponto de partida, onde ocorreram as reuniões iniciais para alinhar escopo, objetivo e metas. Em seguida, realizou-se o levantamento de requisitos e recursos necessários para o projeto. Para garantir a viabilidade e compreensão em relação à nuvem, testes com o Cloudant e a publicação do projeto foram realizados. Além disso, foi essencial adquirir conhecimento sobre os frameworks, principalmente o Flutter, e criar protótipos das aplicações, implementando a conexão do banco de dados.

A fase de desenvolvimento do aplicativo foi, de fato, a criação tanto da aplicação web, quanto mobile. Para isso, foram desenvolvidas as funcionalidades básicas e realizadas pesquisas ou conversas com especialistas em linguística indígena para garantir a precisão das informações linguísticas. Além disso, foi essencial estabelecer a conexão com o banco de dados e as APIs da IBM, garantindo a integração adequada da solução. Durante esse processo, a versão intermediária deste relatório foi revisada.

Na etapa de melhorias e ajustes baseados nos feedbacks, a equipe se concentrou em aprimorar a solução. Testes e entregas contínuas foram realizados, coletando feedbacks valiosos dos stakeholders e incorporando melhorias nas funcionalidades, bem como na interface do usuário (UI) e na experiência do usuário (UX).

Por fim, o fechamento do projeto e validação englobou os últimos testes para garantir que as aplicações estivessem prontos para uso. Após ajustes finais, a apresentação final e a criação de um banner para divulgação foram concluídas. Além disso, o relatório final foi revisado e finalizado como parte do encerramento do projeto. O cronograma pode ser visto na Tabela 1 a seguir:

Tabela 1 - Cronograma



Fonte: Elaborado pelos autores

1.4 Mapeamento dos stakeholders

Tabela 2 - Mapeamento dos stakeholders

Stakeholder	Posição	Papel no Projeto	Expectativas
Graziela Simone Tonin	Professora do Insper	Orientadora do Projeto	Auxiliar os alunos no desenvolvimento do projeto de modo a garantir seu bom desempenho nas tomadas de decisões.
Julio Nogima	Colaborador da IBM	Mentoria e fornecer suporte ao projeto	Contato e validação constante a fim de garantir um

			andamento satisfatório.
Guilherme Augusto Chaves de Carvalho Guilherme Carneiro Lunetta José Rafael Martins Fernandes Rafael Evangelista Monteiro	Alunos Insper	Desenvolvedores	Desenvolver o aplicativo para o auxílio à escrita de linguagens indígenas e um site para mapeamento de caracteres, de acordo com a IBM
Time de linguistas	Colaboradores da IBM	Fornecer suporte ao projeto	Realizar feedbacks e correções em relação às línguas indígenas e teste da aplicação web

Fonte: Elaborado pelos autores

1.5 Riscos Envolvidos

Após um mapeamento dos riscos envolvidos na realização do projeto, os seguintes foram destacados:

Riscos Técnicos: Um risco foi encontrado na aplicação web quando tenta-se fazer uma requisição específica para uma *API* da IBM. Basicamente, ao tentar fazer essa requisição por um navegador, ele é barrado por um serviço de proteção de recursos restritos na internet chamado *CORS (Cross-origin resource sharing)* e que apenas o dono do domínio dessa requisição pode permitir acesso. Esse risco tem grande impacto no projeto, visto que ele impossibilita o total funcionamento da aplicação web. Entretanto, esse risco foi solucionado com a criação de um backend que lida com todas as requisições feitas ao banco de dados.

Riscos de Cronograma: Durante o desenvolvimento do projeto uma grande mudança de escopo ocorreu, mudando a ideia de um aplicativo focado no auxílio da escrita da língua Guarani Mbya para um teclado virtual com caracteres de mais de 40 línguas indígenas. Desta forma, um atraso de cerca de 1 mês aconteceu por conta do difícil acesso a comunidade

indígena, e a dificuldade encontrada pela IBM em se comunicar com eles e assim fechar o escopo do projeto, pondo em risco o cronograma final. Outro risco identificado logo no início do projeto foi relativo ao impacto no cronograma e a realização dos testes das aplicações, tanto com os indígenas, quanto com os linguistas da USP. Isso porque, podem ocorrer conflitos de disponibilidade entre os participantes dos testes e a dificuldade de acesso aos indígenas, uma vez que a IBM não conseguiu trabalhar diretamente com essa população durante o semestre. Desse modo, ocorreu um atraso de desenvolvimento e falta de feedbacks. Por outro lado, esse risco foi mitigado mantendo conversas constantes com o representante da IBM e assegurando que o teste com os linguistas da USP ocorresse, mesmo que depois da entrega do relatório e antes da apresentação para a banca.

1.6 Questões Éticas e Profissionais

Por se tratar de um projeto relacionado à preservação das línguas indígenas brasileiras, é importante que tenha-se atenção à forma como o aplicativo final será apresentado para a população indígena.

Ainda, vale ressaltar que não existem questões econômicas envolvidas no projeto, uma vez que todo o desenvolvimento está sendo realizado com tecnologias *open source* e utilizando a infraestrutura de cloud já existente da empresa parceira. Por fim, não existe uma preocupação relacionada a dados sensíveis e confidenciais, pois o projeto não consome tais tipos de dados.

1.7 Revisão do Estado da Arte

O Brasil é um país diverso, com uma riqueza cultural que se reflete na presença de uma vasta quantidade de comunidades e línguas indígenas. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE, 2022), o Brasil possui uma população indígena de 1.693.535 pessoas, representando 0,83% do total de habitantes do país. O Censo 2010 identificou a existência de 274 línguas indígenas faladas por indivíduos pertencentes a 305 etnias diferentes (IBGE, 2010). Esses dados revelam a dimensão do problema da preservação e valorização dessas línguas no país. Solucionar isso se mostra crucial, não apenas por conservar a história originária do povo brasileiro, mas também por contribuir para a manutenção da identidade cultural de um número expressivo de pessoas em comunidades indígenas.

A complexidade da preservação das línguas indígenas no Brasil é acentuada pelo fato de que muitas delas são predominantemente orais, com poucos registros escritos e sem um padrão estabelecido. Historicamente, a colonização desempenhou um grande papel nesse contexto, como evidenciado no seguinte trecho:

> "O colonizador utilizou as traduções como uma maneira de oprimir o colonizado", embora, evidentemente, sempre reservando espaços de resistência para os povos subjugados, afinal "[...] o colonizado utilizou a tradução para manter a cultura e tradição indígenas"(DAS, 2008, p. 85).

Outro exemplo da influência religiosa na escrita das línguas indígenas, são os dados fornecidos pela IBM, isso porque, os caracteres foram em sua maioria extraídos de traduções da Bíblia.

Durante o período colonial, as traduções, principalmente da Bíblia, foram usadas tanto para doutrinação religiosa quanto para conservar aspectos da cultura indígena. Entretanto, esses aspectos linguísticos e culturais ainda são pouco representados na forma escrita, tornando a tarefa de preservação e até mesmo o uso prático das línguas indígenas um desafio. Principalmente no contexto atual, visto que a falta de teclados e recursos tecnológicos adaptados dificulta ainda mais sua escrita e disseminação.

Portanto, a falta de documentação adequada torna ainda mais desafiador o trabalho de linguistas, educadores e comunidades indígenas que buscam revitalizar e transmitir essas línguas para as gerações futuras. O fato de haver poucos dados disponíveis dificulta o estudo, a análise e a compreensão dessas línguas, o que pode levar à perda irreparável de informações linguísticas e culturais importantes do Brasil.

Nesse contexto, algumas iniciativas foram criadas para mitigar esse problema, com o intuito de contribuir para a revitalização dessas línguas e garantir que elas sejam mais acessíveis no mundo digital. Um exemplo notável é o aplicativo *"Linklado"*, que foi desenvolvido com o propósito de superar a falta de caracteres indígenas e facilitar a escrita e o uso prático dessas línguas. Diferentemente da solução descrita nesse relatório, em que vários teclados serão criados e mapeados com ajuda de linguistas e da IBM, esse aplicativo oferece um layout de teclado de tamanho maior e é adaptado com caracteres especiais e acentos específicos, permitindo combinações que são cruciais para a escrita em línguas indígenas.

Além de pequenos projetos, grandes empresas, como a Motorola, começaram a reconhecer a importância da diversidade linguística e cultural do Brasil. Ao incluir idiomas indígenas ameaçados de extinção, como o Kaingang e o Nheengatu, em seus smartphones com o sistema operacional Android 11, a Motorola começou a contribuir para ampliar o

alcance dessas línguas e promover sua preservação. Ainda que de forma lenta, tais iniciativas demonstram que a conscientização sobre a importância das línguas indígenas está crescendo e iniciativas nesse campo estão surgindo.

Em resumo, é perceptível a importância da preservação das línguas indígenas no Brasil e seus desafios históricos, como a falta de registros escritos e a deficiência tecnológica em relação aos caracteres. Por meio da implementação da plataforma composta por aplicações web e mobile, o projeto descrito por esse relatório tem o intuito de buscar novas soluções para revitalização dessas línguas digitalmente. A personalização do teclado para cada língua específica, que poderá ser feita por linguistas e pela IBM através da aplicação web, simplifica o uso prático e preserva particularidades das línguas. Além disso, com o uso de um teclado no celular, os indígenas poderão documentar, preservar e transmitir suas tradições linguísticas de maneira digital, incentivando o aprendizado e uso dessas línguas, assim, fortalecendo a identidade cultural dessas comunidades.

1.7.1 Tecnologias e Gestão de Projeto

A fim de atingir o objetivo mencionado anteriormente, alguns frameworks para desenvolvimento web foram testados, como o Angular e o Vue JS. Entretanto, o ReactJS foi escolhido por, além de ser um dos principais frameworks para desenvolvimento web, é eficiente na criação de plataformas interativas e responsivas, atendendo todos os requisitos do projeto. Para garantir uma arquitetura robusta e eficiente, optamos pelo uso do Express.js, que permite a implementação de um *backend* simples e funcional para a aplicação em React. A combinação de ambos permite um desenvolvimento facilitado por utilizarem uma linguagem popular, o Javascript, simplificando o fluxo de desenvolvimento.

Para o desenvolvimento mobile, foi escolhido o Flutter, framework que cada vez mais ganha destaque, principalmente por ser multiplataformas. O Flutter permite a criação de aplicativos para Android e iOS a partir de um único código-fonte, economizando tempo e recursos no desenvolvimento. Além disso, tanto o React quanto o Flutter são *open source*, o que significa que a comunidade pode contribuir e melhorar constantemente essas tecnologias, não sendo necessário custear uma licença específica de uso, alinhando-se com nossos objetivos de preservação e acessibilidade.

Além disso, a adoção de metodologias ágeis, amplamente utilizadas no mundo, e o framework de desenvolvimento *Scrum* também se alinham com o objetivo do projeto. As metodologias ágeis são abordagens colaborativas e iterativas para o desenvolvimento de um

projeto, possibilitando uma gestão otimizada e incorporação constante de feedbacks dos stakeholders. Dentro desse contexto, o Scrum, um framework ágil, se destaca como uma abordagem de gerenciamento que enfatiza a colaboração, a adaptação e a entrega contínua de valor por meio de sprints.

O *Scrum* divide o desenvolvimento em iterações chamadas sprints, geralmente de uma a quatro semanas, durante as quais as equipes trabalham para entregar funcionalidades específicas. Essas iterações possibilitam uma resposta rápida às mudanças nos requisitos e aprimoramentos do projeto. Além disso, existem papeis cruciais, como o *Product Owner*, que é responsável por representar os interesses do cliente, intermediando todos os contatos, e garantir o valor do produto. Já o *Scrum Master*, desempenha um papel de facilitador, promovendo a colaboração e a organização, evitando grandes obstáculos. Os desenvolvedores, por sua vez, são responsáveis pela implementação das funcionalidades conforme orientadas pelo *Product Owner*, colaborando de forma eficaz para garantir entregas bem-sucedidas durante os sprints. A aplicação das metodologias ágeis, em conjunto com o Scrum, ofereceu uma abordagem flexível e eficaz para o desenvolvimento desse projeto, assegurando que as aplicações atendessem de forma precisa e contínua às necessidades das línguas indígenas identificadas.

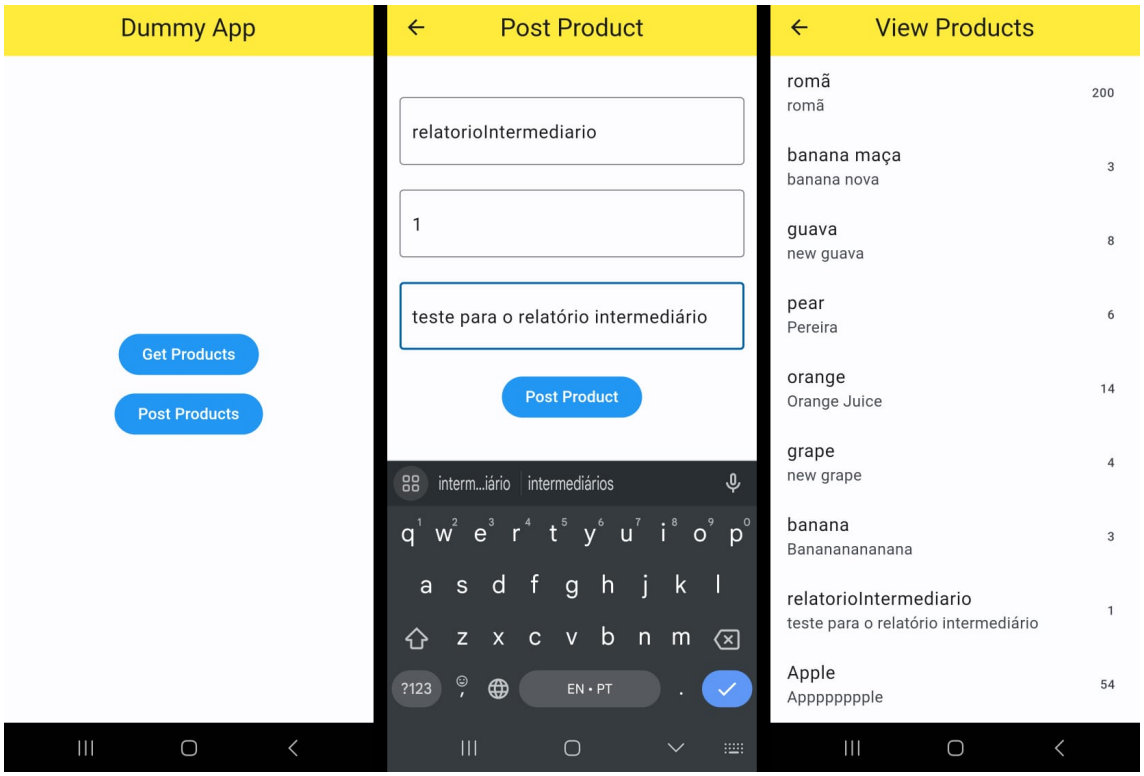
2. Metodologia

A metodologia adotada para o desenvolvimento, tanto da aplicação mobile quanto da web, desempenha um papel crucial no êxito de nossos objetivos. Inicialmente, o foco foi em compreender o escopo do projeto, as peculiaridades das línguas indígenas e, principalmente, os requisitos, ferramentas e funcionalidades solicitados pela IBM.

Com base nessa compreensão inicial, deu-se início à fase de prototipagem, criando um *dummy app*[1] que pode ser visto na Figura 1 abaixo. Esse protótipo foi feito com o intuito de entender e explorar recursos obrigatórios, como o sistema de banco de dados e caracteres unicode, e escolher os frameworks *open source* optativos, o Flutter e o React, para o desenvolvimento das aplicações. Essa fase inicial foi fundamental para testar conceitos e validar a viabilidade técnica do projeto.

[1] Aplicativo sem funcionalidade específica. Criado apenas para fins de teste.

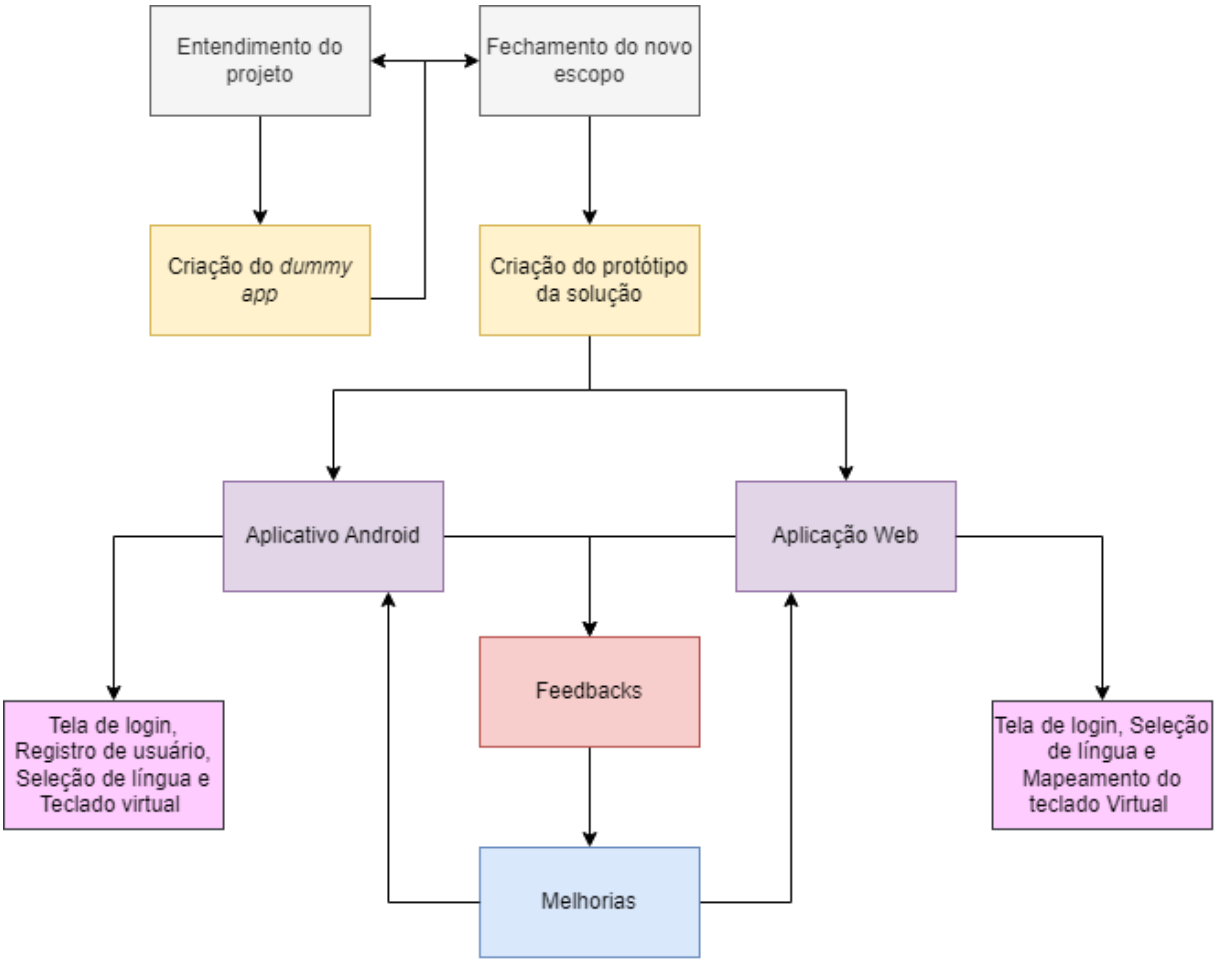
Figura 1 - Telas do *dummy app*



Fonte: Elaborado pelos autores

Durante todo o processo de desenvolvimento, optamos pela divisão de tarefas e a implementação de um framework ágil *Scrum*, utilizando sprints semanais visando entregas contínuas de valor e validação constante junto ao cliente por meio de alinhamentos semanais. Isso garantiu a constância e a organização de prazos, permitindo que a equipe fosse eficiente na resposta a mudanças e no aperfeiçoamento contínuo das aplicações. O fluxograma de todas as etapas pode ser visualizado no Diagrama 1, abaixo:

Diagrama 1 - Fluxograma de desenvolvimento



Fonte: Elaborado pelos autores

As descrições detalhadas de cada sprint, são:

- Sprint 1: Entendimento do escopo do projeto, escolha das ferramentas utilizadas e primeiros testes com a plataforma *Cloudant* (serviço de banco de dados da IBM). Além disso, os caracteres das línguas indígenas e seus respectivos unicodes foram estudados.
- Sprint 2: Após decidido as ferramentas para a construção das aplicações web e *mobile*, foi desenvolvido uma *Landing Page* e um *Dummy App*, respectivamente. Nesses protótipos, testou-se a conexão com os bancos de dados, criando uma base para o desenvolvimento futuro das aplicações. Ainda, durante essa sprint, houve uma mudança de escopo, gerando um risco (dificuldade encontrada pela IBM em se comunicar com os futuros usuários e assim fechar o escopo do projeto).

- Sprint 3: Após o fechamento do escopo do projeto e entendimento dos requisitos, o foco foi na implementação dos requisitos começando pelo desenvolvimento de uma primeira versão do design do sistema. Com o uso do figma, foi criado um protótipo visual, que a posteriore foi validado com o cliente. A partir do *Dummy App*, as funcionalidades básicas foram criadas, do aplicativo mobile: Registrar, *login* e escrever em algumas línguas indígenas, e da aplicação web: *Login* e mapeamento do teclado.
- Sprint 4: Como já havia sido feito bastante coisa do projeto, essa sprint foi dedicada a melhorar a UI (*User Interface*) das aplicações, desenvolvimento do relatório e apresentação para a banca intermediária.
- Sprint 5: Após a coleta de feedbacks da banca intermediária, o foco da Sprint foi a discussão em relação às melhorias das plataformas com a IBM. Nesse contexto, aderindo a uma sugestão da banca, foram escolhidas as línguas que seriam priorizadas, como seria possível melhorar as realocações dos caracteres indígenas e foi decidido que um *backend* seria necessário para evitar eventuais problemas de CORS, que é um mecanismo que, por meio de cabeçalhos HTTP, permite que uma aplicação web seja executada em uma origem e acesse recursos de outra origem com segurança.
- Sprint 6: O desenvolvimento do *backend* foi o foco da sprint. Para isso, foi escolhido o framework Express.js, a fim de atender necessidades do projeto, requisições da IBM e manter um padrão de linguagem para a aplicação web. Dessa forma, as requisições para o banco de dados do *Cloudant* deixaram de ser através do frontend e passaram a ser feitas pelo backend.
- Sprint 7: O primeiro aprimoramento que foi feito na aplicação web e foco durante essa sprint foi refazer o grid que simula um teclado. A fim de tornar o grid o mais parecido com um teclado virtual e ter todas as funcionalidades discutidas nas sprints anteriores, deixamos de usar bibliotecas e desenvolvemos um algoritmo específico para melhor atender a solicitação do usuário.
- Sprint 8: Com o grid feito, foi decidido que o usuário poderia não só adicionar os caracteres indígenas em espaços em branco do teclado, mas também colocar caracteres acentuados dentro de letras. Dessa forma, nessa sprint foi desenvolvido o sistema de *popup* para que essa ferramenta funcionasse.

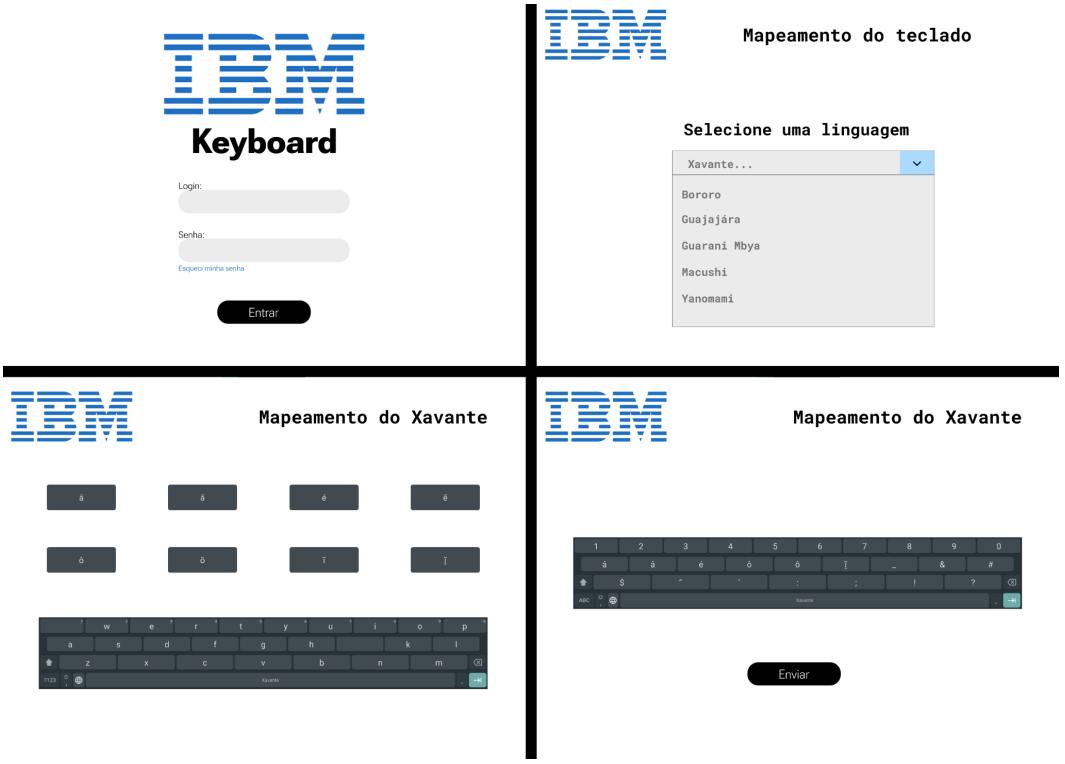
- Sprint 9: Após o mapeamento das línguas estar pronto, essa sprint envolveu o desenvolvimento do login e do aparecimento do teclado mapeado na aplicação web . Para isso houve um aprimoramento de funcionalidades do *backend*.
- Sprint 10: Esta sprint foi responsável pelo aprimoramento do teclado da aplicação mobile. Nela, o visual do teclado virtual foi refeito para que se parecesse com o disponibilizado pelo Android. Além disso, foi adicionada a funcionalidade de segurar a tecla para digitar caracteres com acentuação.
- Sprint 11: Por fim, a última sprint foi destinada a melhorar a UX (*User Experience*) das aplicações através de testes, desenvolvimento final do relatório e apresentação para a banca.

2.1 Aplicação Web

O primeiro passo para a realização da aplicação web foi o desenvolvimento de um protótipo. Para isso, iniciou-se alguns estudos sobre os frameworks *open source* mais utilizados para desenvolvimento, em que o ReactJS foi escolhido. O objetivo da prototipação era entender como a aplicação faria a conexão com o banco de dados do *Cloudant*, disponível na IBM Cloud. Isso envolveu a criação de um esboço inicial das funcionalidades e, principalmente, a configuração das requisições por meio da *API* da IBM.

Após estabelecer a base do projeto, foi feita a criação do design da aplicação web e a definição da experiência do usuário (UX). Para isso, utilizou-se o Figma para criar um *mockup* detalhado. Esse passo é essencial para visualizar como a aplicação funciona na prática, desde a navegação até a interação do usuário. Esse mockup pode ser visualizado nas seguintes imagens.

Figura 2 - *Mockup* visual da interface web



Fonte: Elaborado pelos autores

Posteriormente, iniciou-se o processo de desenvolvimento de um MVP *(minimum viable product*) da aplicação web, com as funcionalidades principais. Foi definido que seguiríamos a ideia de modularização trazido pelo React, tornando cada elemento visível na tela, em um componente. Assim, todos os itens descritos a seguir, seguem essas boas práticas.

A primeira fase de elaboração consistiu no desenvolvimento da versão inicial da aplicação web. O objetivo principal nesse estágio foi criar uma base para futuras iterações, ao mesmo tempo em que feedbacks foram sendo coletados. Com isso, criou-se um esboço do sistema de login sem a funcionalidade de verificação e conexão com o banco de dados. Em seguida, implementou-se a funcionalidade de seleção de língua através de um dropdown na barra de navegação. Outra implementação feita foi a capacidade de montar um teclado personalizado por meio de drag and drop dos caracteres em um grid que simula, ainda sem focar na UI (User Interface), a interface de um teclado. Para isso, utilizou-se a biblioteca react-grid-dnd para criar esse sistema interativo que permite os usuários dispor os caracteres de acordo com suas preferências. Quando o usuário finaliza tal mapeamento, surge um botão de envio para o banco de dados criado no *Cloudant* e consumido pelo aplicativo mobile. Para tornar isso possível, as requisições foram feitas através da biblioteca axios.

Por fim, a navegação entre páginas foi implementada utilizando a biblioteca React Router Dom, que com poucas funções torna possível a renderização e mudança de todos os elementos da página. Além disso, a estilização da aplicação foi feita com o framework Tailwind CSS, um método eficiente e que facilita a personalização da UI (User Interface) por possuir uma série de classes CSS prontas.

A segunda fase de elaboração do MVP consistiu no desenvolvimento da versão final da aplicação web. Nessa etapa, além de dar continuidade implementando funcionalidades requisitadas pela IBM, foi aprimorado o projeto de acordo com os feedbacks coletados. Nesse contexto, as requisições feitas a partir da biblioteca *axios*, passaram a ser feitas por um backend. Tal mudança foi realizada com dois principais objetivos: evitar qualquer problema de CORS e garantir a segurança da aplicação com os dados. O CORS impede que o frontend acesse recursos no banco de dados diretamente, devido a políticas de segurança do Cloudant. Portanto, o backend intermedia as requisições do frontend para o Cloudant, o que evita problemas de CORS e fortalece a segurança dos dados, já que o frontend não tem acesso direto ao banco de dados. Para garantir a consistência do desenvolvimento da aplicação web, optamos por usar o framework Express.js, já que é uma ferramenta open source e em JavaScript, o que cumpre requisições da IBM e mantém uma linguagem padrão.

Além disso, o backend funciona de forma simples e eficaz. Ele consiste em receber um objeto JSON do frontend, contendo uma linguagem e seu mapeamento. Em seguida, o backend encaminha as informações para o banco de dados através de um endpoint. Para garantir a segurança e a autorização adequada do Cloudant, utilizamos um arquivo .env que armazena a chave de API do banco de dados. A partir dessa chave, o código gera um token de autenticação, que é renovado automaticamente ao expirar, e o usa para autenticar as requisições válidas.

A partir do backend, outro elemento implementado foi a tela de login e suas funcionalidades, para garantir que apenas linguistas e funcionários da IBM tenham acesso à aplicação. Com o intuito de atingir esse objetivo, não será possível criar um usuário no frontend da aplicação web. Para isso, há um endpoint no backend que permite o envio de uma requisição que cria um novo usuário. A senha fornecida é salva e criptografada por uma biblioteca chamada bcrypt no banco de dados.

Já a realização do login é feita a partir de outro endpoint. Dessa forma, ao tentar fazer o login pelo frontend, os dados digitados nos campos de email e senha são enviados ao backend, onde uma requisição para o banco de dados é feita, com o intuito de conferir se

existe uma conta. Caso ela exista, o backend retorna o email e a senha para o frontend e, através da biblioteca bcrypt React, compara para ver se a senha criptografada do banco de dados é igual a senha inserida. Por fim, se as senhas forem iguais, o usuário consegue entrar na aplicação. Contudo, caso o e-mail não seja vinculado à uma conta e/ou as senhas forem diferentes, o usuário receberá uma notificação de erro.

Outra modificação significativa que a aplicação web passou, foi na abordagem do desenvolvimento do grid que simula a interface de um teclado. No processo de aprimoramento, foram incorporados os caracteres do teclado padrão Android para todas as linguagens, incluindo uma segunda página dedicada a pontuações, para que se assemelha ao máximo com um teclado virtual. Além disso, a principal mudança envolveu a escolha de uma nova biblioteca para a realização da funcionalidade de drag and drop, já que a anterior, chamada react-grid-dnd, não oferecia suporte adequado para as atualizações necessárias. Contudo, nenhuma outra biblioteca do React atendia aos requisitos específicos, o que levou à decisão de desenvolver a lógica do zero, utilizando somente o JavaScript no React.

A lógica para o drag and drop foi estruturada em três grids: o primeiro destinado a caracteres especiais das línguas indígenas que precisam ser posicionados no teclado, o segundo referente à primeira página do teclado, e o terceiro relacionado à segunda página do teclado. Cada elemento de cada grid é manipulável, permitindo sua movimentação para alterar a ordem. A implementação do drag and drop envolveu três funções distintas padrões do JavaScript: onDragStart, onDragEnter e onDragEnd. A primeira é acionada ao selecionar e a arrastar um caractere em um dos grids.Já onDragEnter, é chamada quando o caractere em movimento colide com outro, e abrange toda a lógica de substituição de caracteres. Por fim, o drag reinicia as variáveis inicializadas no onDragStart e é ativado quando o usuário solta o item, encerrando o movimento de arrastar.

Após análise de feedbacks e discussões com a IBM, uma nova funcionalidade foi implementada na aplicação, permitindo aos usuários inserir caracteres dentro de outros. Essa abordagem possibilita que uma letra acentuada tenha como origem a mesma letra não acentuada, seguindo o modelo dos teclados do Android. A implementação dessa característica envolveu a criação de um novo grid, acessível por meio de um popup que é ativado quando o usuário seleciona um item no teclado.

O novo grid interage exclusivamente com o grid dos caracteres especiais. Isso significa que, se a necessidade for incluir, além de um caractere indígena, uma letra do teclado padrão, a letra padrão deve ser movida para o grid dos caracteres especiais antes de ser

inserida no novo contexto. Essa abordagem proporciona uma experiência mais personalizável e adaptável aos diferentes requisitos de cada língua. Isso porque, é permitido que o usuário coloque os novos caracteres tanto nos espaços em branco nas duas páginas, quanto em cada letra do teclado.

Por fim, a última grande mudança em comparação à primeira versão, diz respeito aos mapeamentos anteriores que são armazenados e recuperados do banco de dados. Inicialmente, ao iniciar a aplicação, o teclado padrão e os caracteres especiais eram extraídos de um arquivo JSON. Contudo, essa abordagem foi aprimorada para que o frontend, ao iniciar, tente fazer uma requisição ao banco de dados para verificar se já existe um mapeamento da língua selecionada.

Para viabilizar essa funcionalidade, foi adicionado um novo *endpoint* no backend, permitindo recuperar o último mapeamento de uma língua específica. Assim, a lista de todos os caracteres e suas posições são armazenados no banco de dados, possibilitando que o *frontend* obtenha esses dados e construa o teclado para a visualização do usuário. Adicionalmente, para que o usuário entenda os mapeamentos anteriores, os caracteres especiais estarão em vermelho e, caracteres que têm outros dentro, estarão em amarelo.

Além disso, para criar o teclado padrão de línguas ainda não mapeadas, é utilizada uma função específica. Se o *frontend* fizer uma requisição ao banco de dados e nenhum mapeamento anterior for encontrado, essa função é acionada, extraindo os caracteres especiais e padrões do arquivo JSON. Outra situação em que essa função pode ser utilizada é quando a requisição retorna um mapeamento já existente. Nesse caso, surge um botão que permite ao usuário reiniciar as modificações realizadas e ao acioná-lo, a função é novamente chamada, remontando o teclado com base no arquivo JSON.

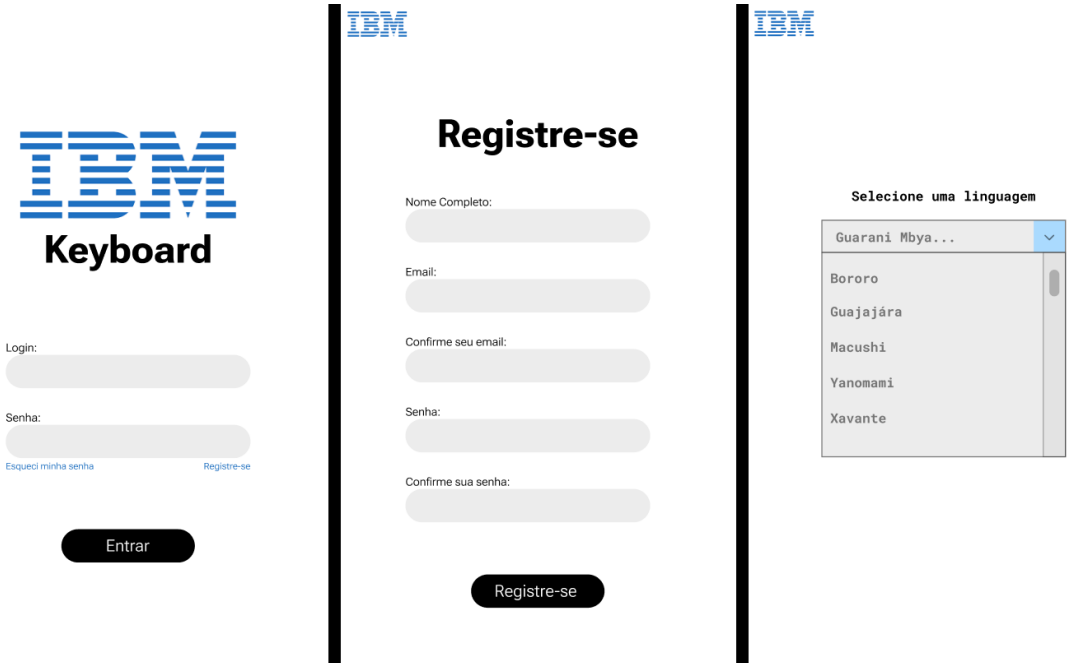
2.2 Aplicativo Mobile

O primeiro passo para a realização da aplicação *mobile*, como mencionado anteriormente, foi o desenvolvimento de um *dummy app*, que é basicamente um protótipo com o intuito de testar ferramentas, ignorando o design. O objetivo desse protótipo foi o teste da conexão com o banco de dados *Cloudant* e a validação com a equipe da IBM, assim, obtendo um resultado positivo ao final do desenvolvimento.

Após o sucesso do protótipo, iniciou-se a criação do design da aplicação *mobile* e a definição da experiência do usuário (UX). Para isso, utilizou-se novamente o Figma para criar um mockup detalhado. Esse passo é essencial para visualizar como a aplicação funcionaria na

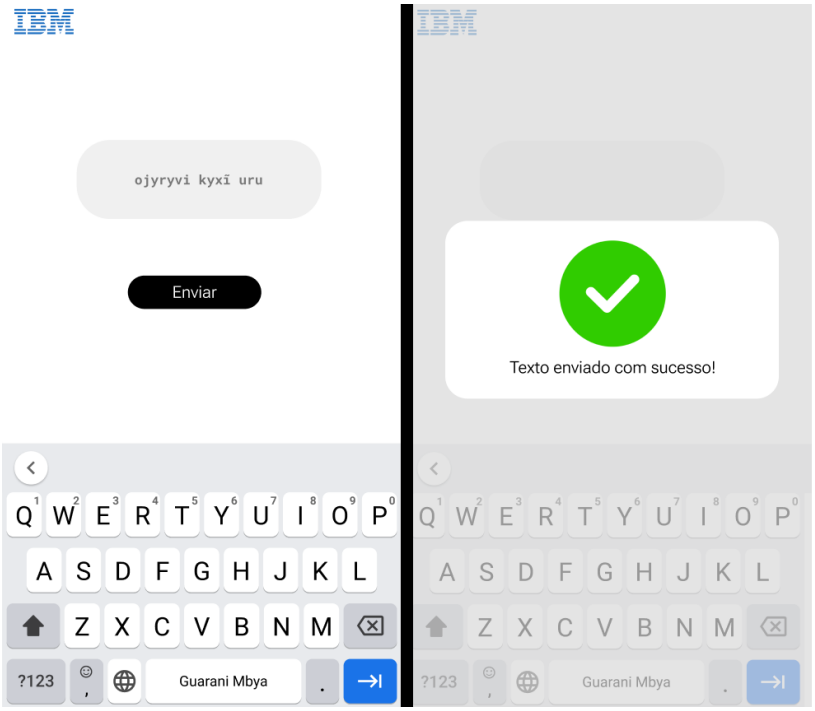
prática, desde a navegação entre telas até a interação do usuário com as funcionalidades. Esse *mockup* pode ser visualizado nas seguintes imagens:

Figura 3 - Primeiras telas do *mockup* visual do aplicativo Android



Fonte: Elaborado pelos autores

Figura 4 - Outras telas do *mockup* visual do aplicativo Android



Fonte: Elaborado pelos autores

Depois da criação do design, iniciou-se o desenvolvimento da versão inicial do MVP (*minimum viable product*) da aplicação *mobile*, seguindo o design criado anteriormente juntamente às funcionalidades estabelecidas. A primeira *feature* desenvolvida foi um sistema completo de *login* e cadastro de um usuário, requisito pedido pela IBM nas conversas anteriores. Para tornar esse sistema possível, foi criado um novo banco de dados que registra as informações do usuário ao fazer o cadastro e salva sua senha criptografada. Dessa forma, ao tentar acessar o aplicativo, é feita uma requisição para o banco de dados e as informações são comparadas, permitindo o login. Ainda, caso exista algum erro no cadastro ou login, como uma senha errada ou a tentativa de criação de um usuário repetido, uma notificação aparece no app, indicando qual foi o problema. Em seguida, foi criada uma tela com um *dropdown* para a seleção de uma das línguas indígenas, um campo de texto onde o usuário poderá escrever usando os caracteres padrões e especiais de cada língua e o teclado do idioma escolhido, vindo diretamente do banco de dados *Cloudant*, após os caracteres da língua escolhida serem mapeados pelos linguistas da IBM na aplicação web. Essa troca de dados foi possível através de requisições ao banco de dados a partir da biblioteca *http* do Flutter.

Por fim, a pedido da IBM, para a última funcionalidade da versão inicial, foi preparado um sistema de envio de mensagens que serão armazenadas em um banco de dados visando estudos futuros através de modelos de *machine learning*. Cada mensagem armazenada irá conter o texto escrito pelo usuário no aplicativo, a língua indígena escolhida e o usuário que enviou a mensagem. Esses 3 parâmetros enviados ao banco de dados irão possibilitar um avançado estudo no futuro pelos linguistas da IBM. No entanto, a empresa parceira deixou claro que essa funcionalidade não está completa, por isso, nosso objetivo é apenas enviar as mensagens para o banco de dados e, no futuro, a IBM vai consumir essas mensagens.

Após a coleta de feedbacks, a segunda fase do desenvolvimento do MVP consistiu no desenvolvimento da versão final da aplicação mobile. A primeira alteração diz respeito à interface do usuário (UI) e à apresentação do teclado. Nesta versão, as teclas foram redesenhadas com uma forma mais retangular, visando uma semelhança com teclados já existentes no Android. Adicionalmente, foi introduzida uma nova página de teclado para não só abranger todos os caracteres presentes em um teclado padrão, mas também garantir compatibilidade com o teclado mapeado pela aplicação web. Para essa implementação, foram utilizadas duas ferramentas do Flutter, o GridView para organizar as teclas do teclado e o PageView para gerenciar a paginação do teclado.

Outra modificação relevante envolve a capacidade de digitar caracteres contidos dentro de outros, como acentuação e caracteres indígenas. Para isso, foi utilizada uma lógica semelhante à aplicação web. Ao manter pressionada uma tecla, um popup é ativado, exibindo os caracteres previamente mapeados. O usuário pode então selecionar o caractere desejado, fechando o popup e inserindo o caractere na caixa de texto.

Finalmente, um último aprimoramento foi incorporado. Na caixa de texto, foi adicionado um recurso de rolagem, o scroll, assegurando que os usuários possam escrever textos sem limitações de espaço. Essas mudanças visam proporcionar uma experiência de usuário mais intuitiva e abrangente na versão final da aplicação mobile.

2.3 Coleta de dados sobre o projeto

Durante o projeto, reuniões semanais de alinhamento, explicação de conceitos e sobre dados a serem utilizados estão sendo realizadas com a IBM, a fim de guiar o desenvolvimento das aplicações e validar constantemente os requisitos e entregáveis com o cliente. O conteúdo principal delas está explicitado na Tabela 3 abaixo:

Tabela 3 - Tabela de dados coletados

Data	Local	Duração	Participantes externos	Pautas
18/08/2023	Microsoft Teams	45 minutos	Julio Nogima e Graziela Simone Tonin	Apresentação do projeto e agendamento das reuniões semanais
24/08/2023	Webex	1 hora	Julio Nogima	Discussão do escopo do projeto, ferramentas utilizadas, cronograma e protótipo das aplicações

31/08/2023	Webex	1 hora	Julio Nogima	Feedback do protótipo e mudança no escopo
06/09/2023	Webex	1 hora e 30 minutos	Julio Nogima e Graziela Simone Tonin	Explicação do novo escopo, dados das 40 línguas indígenas e seus respectivos números Unicode
13/09/2023	Webex	45 minutos	Julio Nogima e Graziela Simone Tonin	Discussão sobre dúvidas do projeto e início do desenvolvimento
21/09/2023	Webex	45 minutos	Julio Nogima	Discussão sobre o número de línguas no sistema e apresentação do MVP das aplicações
28/09/2023	Webex	30 minutos	Julio Nogima	Coleta de feedbacks sobre o relatório e apresentação intermediária

11/10/2023	Webex	1 hora	Julio Nogima e Graziela Simone Tonin	Discussão sobre os feedbacks coletados na apresentação.
19/10/2023	Webex	1 hora	Julio Nogima	Discussão sobre a versão final do MVP, principalmente o design da aplicação web
26/10/2023	Reunião Cancelada			
2/11/2023	Reunião Cancelada			

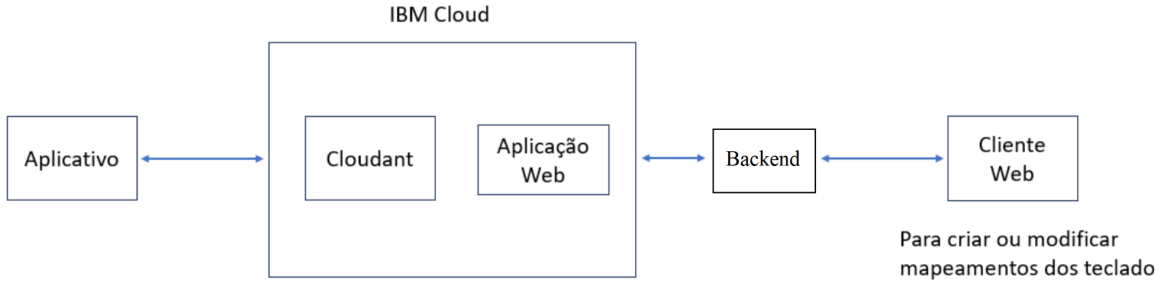
9/11/2023	Webex	1 hora	Julio Nogima e Graziela Simone Tonin	Apresentação da aplicação web e coleta de feedbacks
16/10/2023	Webex	45 minutos	Julio Nogima e Graziela Simone Tonin	Apresentação final da aplicação web e mobile

Fonte: Elaborado pelos autores

2.4 Análise dos Dados Coletados

As principais ferramentas a serem utilizadas são a IBM Cloud e o serviço de banco de dados Cloudant. Dessa forma, tanto a aplicação web, quanto os bancos de dados, serão hospedados em uma cloud proprietária da empresa, possibilitando a escalabilidade e atualizações futuras. O serviço Cloudant tem uma *API* que permite requisições aos bancos de dados criados para o projeto com informações de login e dos caracteres das línguas indígenas e, por meio dela, é possível inserir e extrair dados desejados. A fim de garantir o funcionamento da API, foi utilizado um backend, o que evita problemas de segurança. Abaixo, o diagrama 2 demonstra o funcionamento e arquitetura do projeto.

Diagrama 2 - Funcionamento e arquitetura do projeto



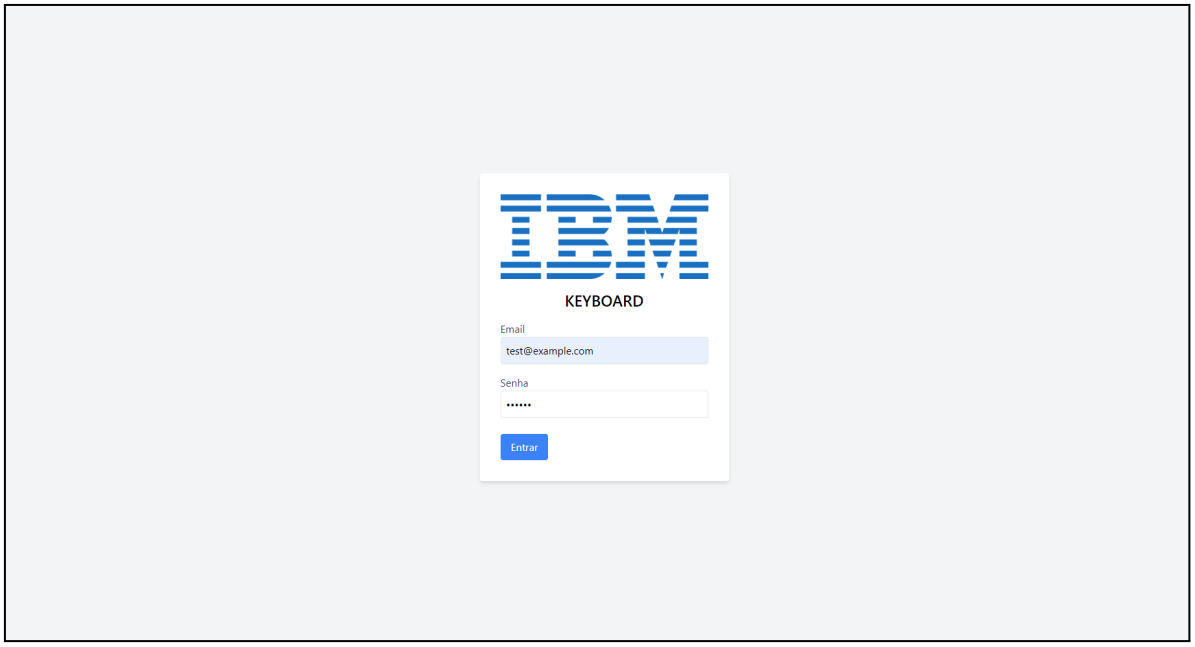
Fonte: Elaborado por Julio Nogima

3. Resultados

De modo a atingir o objetivo da IBM de construir uma plataforma experimental de teclado virtual para línguas indígenas em android e contribuir não só com a preservação cultural, mas também com o aumento de registros escritos, as aplicações web e mobile foram desenvolvidas. Na aplicação web, existem três principais funcionalidades: a tela de login, a seleção e o mapeamento das línguas.

Com o intuito de chegar a um resultado final satisfatório, foram realizadas duas iterações do projeto, uma versão inicial entregue na banca intermediária e outra na banca final. Na primeira, existia na aplicação web uma tela de login com o design adaptado do Figma, levando em consideração os requisitos propostos pela IBM, sem nenhuma conexão com o banco de dados. Já na segunda versão, apesar de nenhuma alteração visual, foi implementado a verificação dos dados inseridos pelo usuário. Nessa tela, o usuário deve informar um email e senha, previamente cadastrados pela IBM, a fim de acessar a aplicação. Caso ocorra alguma falha na verificação dos dados, um erro será exibido na tela. A figura 5 representa essa funcionalidade.

Figura 5 - Tela de login da aplicação web

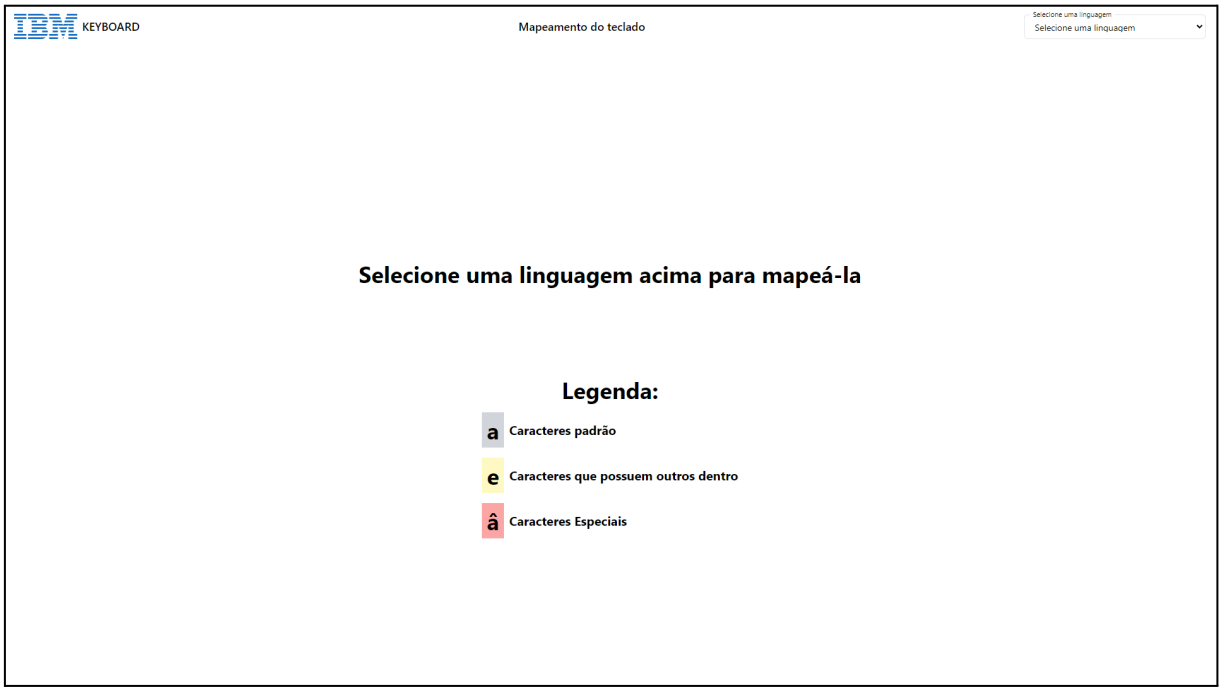


Fonte: Elaborado pelos autores

Além disso, na versão final, foi adicionado na tela de seleção da linguagem um texto com instruções para o usuário selecionar uma língua e a legenda das cores que aparecem na tela de mapeamento. Nessa página, o usuário deve escolher a linguagem que deseja mapear

através do menu *dropdown*, presente na barra de navegação. Tal funcionalidade com suas respectivas alterações pode ser vista na figura 6.

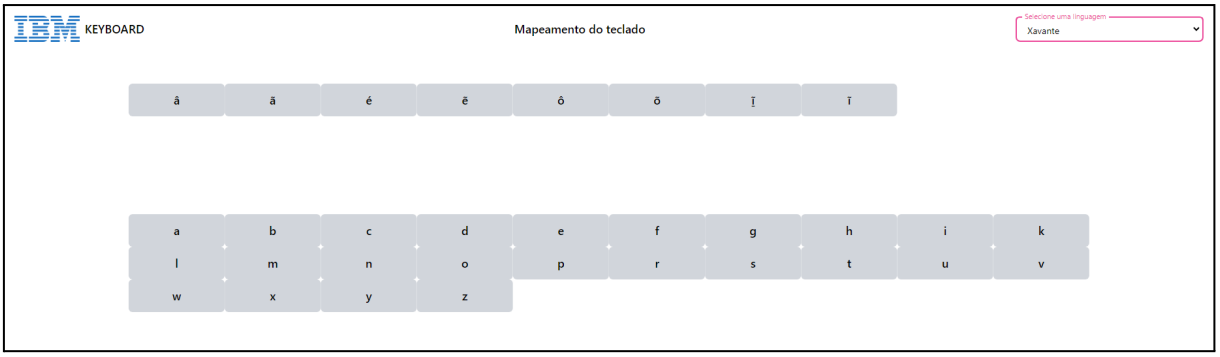
Figura 6 - Tela de seleção de línguas na aplicação web



Fonte: Elaborado pelos autores

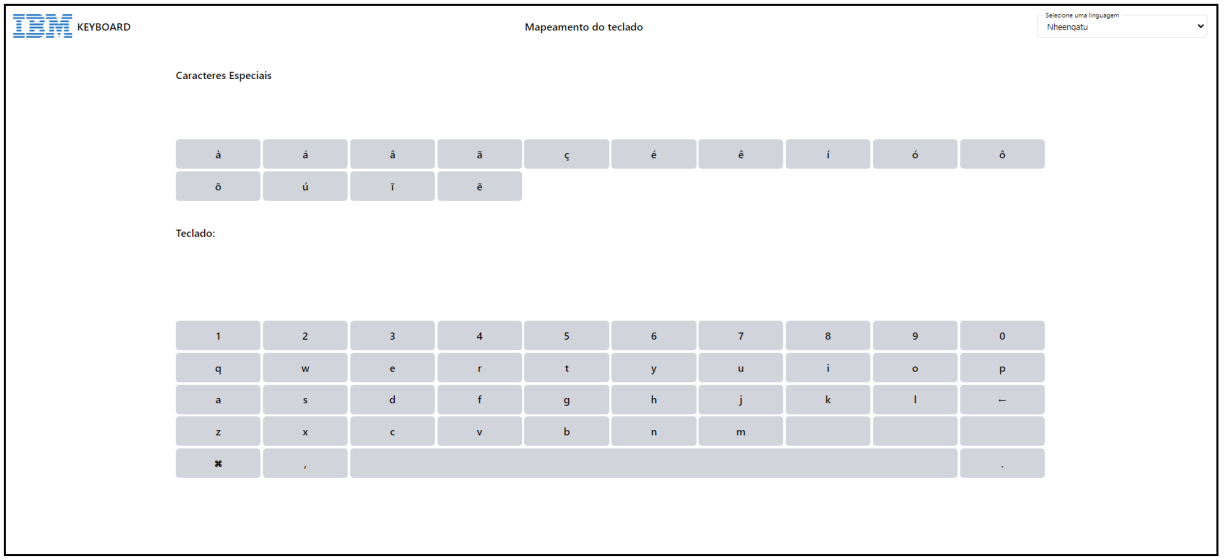
A tela do mapeamento é o local onde o usuário pode utilizar a *feature* de *drag and drop* para personalizar o teclado da maneira que desejar, seja rearranjando os caracteres, ou adicionando uns dentro de outros. Em comparação com a versão inicial, o teclado foi alterado de forma a parecer com o teclado nativo do Android. Isso consiste em sua ampliação para duas páginas, na adição de números e pontuação. Ainda, assim como na primeira iteração, os caracteres especiais aparecem em um grid no topo da tela. Esses aprimoramentos podem ser vistos ao comparar a figura 7, versão inicial, e figura 8, versão final.

Figura 7 - Tela de mapeamento da versão inicial da aplicação web



Fonte: Elaborado pelos autores

Figura 8 - Tela de mapeamento da versão final da aplicação web

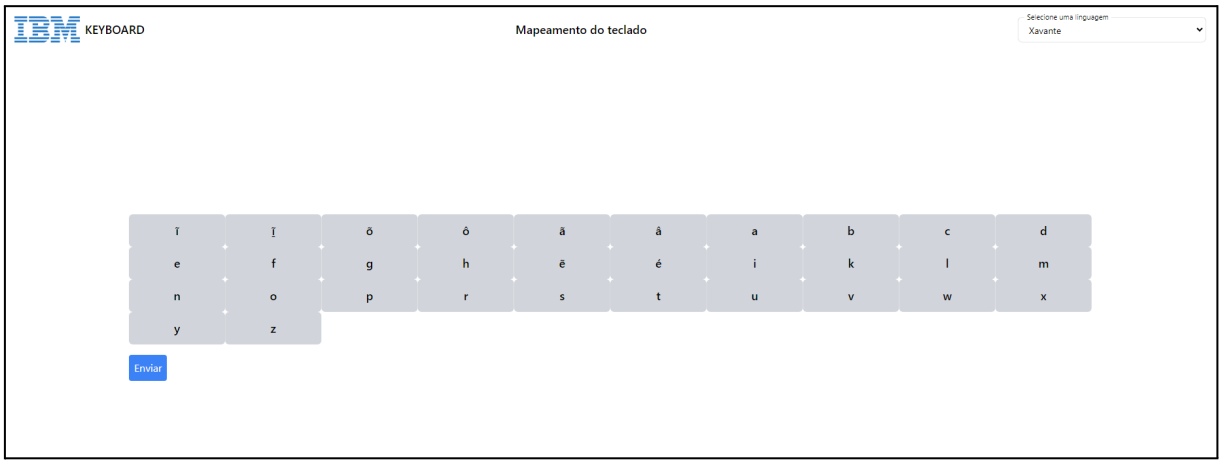


Fonte: Elaborado pelos autores

Por fim, a última mudança da aplicação web, envolve a adição de uma nova funcionalidade na tela de mapeamento. Ao clicar em uma das letras do teclado padrão, na versão final, um *popup* surge, permitindo que o usuário coloque caracteres especiais dentro da letra previamente selecionada. Além disso, um esquema de cores foi criado para facilitar o uso da aplicação. Os caracteres que aparecem em amarelo, são aqueles que possuem outros dentro deles, enquanto os que estão em vermelho, são os caracteres especiais que foram posicionados no teclado. Tais mudanças foram feitas para permitir que o usuário personalize o teclado da forma que considerar mais adequada, para facilitar o uso e garantir que o mapeamento seja mais intuitivo. Na figura 9, é possível ver o mapeamento finalizado na versão inicial, enquanto na figura 10, o mesmo mapeamento foi feito na versão final. Ao

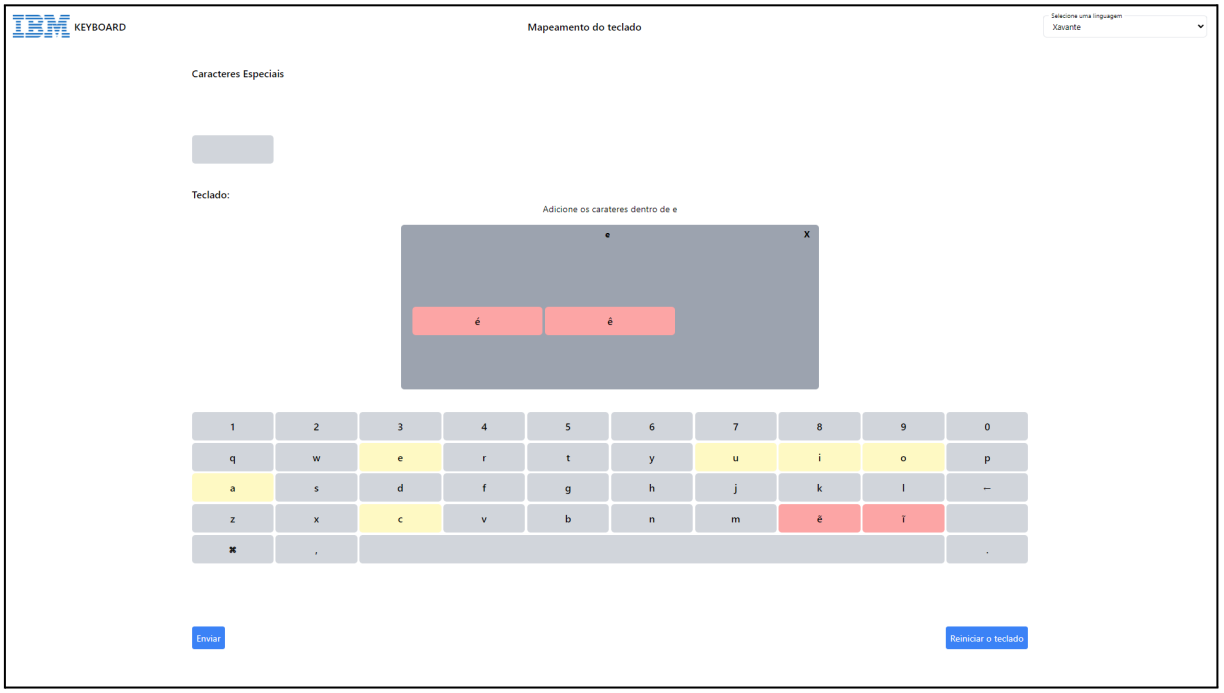
compará-los, é possível ver que as mudanças do teclado são mais perceptíveis, indicando ao usuário onde as teclas foram posicionadas.

Figura 9 - Exemplo de teclado mapeado na versão inicial da aplicação web



Fonte: Elaborado pelos autores

Figura 10 - Exemplo de teclado mapeado na versão final da aplicação web

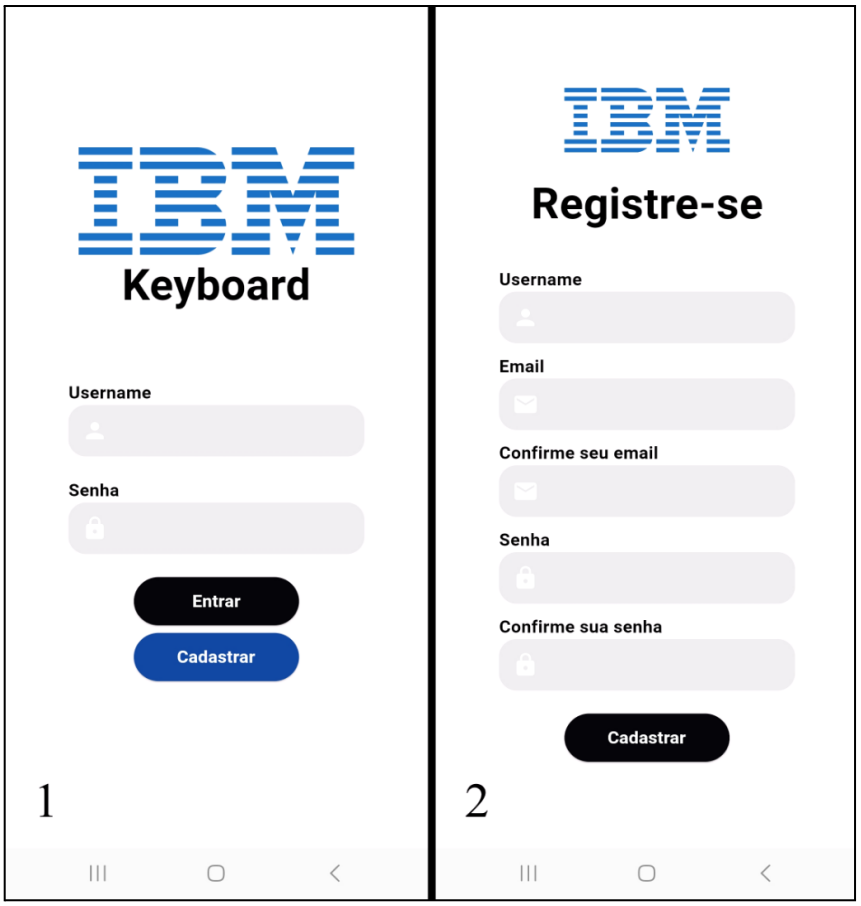


Fonte: Elaborado pelos autores

Já no aplicativo mobile, existem quatro principais funcionalidades: a tela de login, a tela de cadastro, a seleção da língua e o teclado mapeado. Os resultados, vistos na figura 11, são relacionados às duas primeiras features e não sofreram mudanças visuais após sua primeira implementação. O design da tela foi adaptado do *mockup* do Figma, levando em consideração os requisitos propostos pela IBM. Contudo, houve um aprimoramento na forma

em como a senha é salva no banco de dados, na versão final, ela passou a ser criptografada. Na primeira tela, é possível que o usuário crie uma conta caso ele não seja registrado, ou preencha seus dados e acesse o aplicativo, ainda, se ocorrer alguma falha na verificação dos dados, um erro é exibido. Em seguida, na segunda tela, entende-se o caminho de registrar um usuário, para que o erro mencionado anteriormente não aconteça. Desse modo, os espaços devem ser preenchidos com *username*, e-mail e senha.

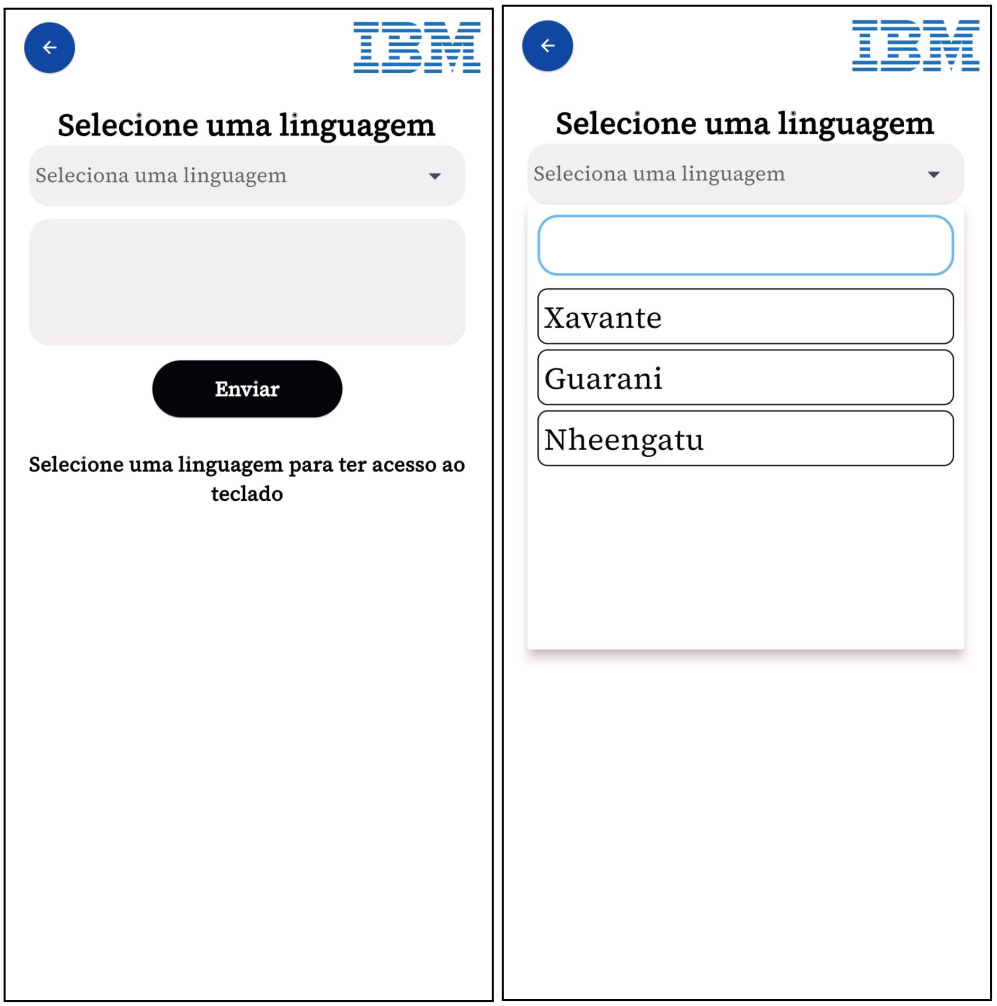
Figura 11 - Versão final da tela de login e de cadastro do aplicativo mobile



Fonte: Elaborado pelos autores

Além disso, a figura 12, representa outra funcionalidade, a tela de seleção das línguas. Nela, o usuário deve selecionar a linguagem cujo teclado deseja utilizar, por meio do menu de *dropdown*, onde é possível filtrá-las digitando o seu respectivo nome.

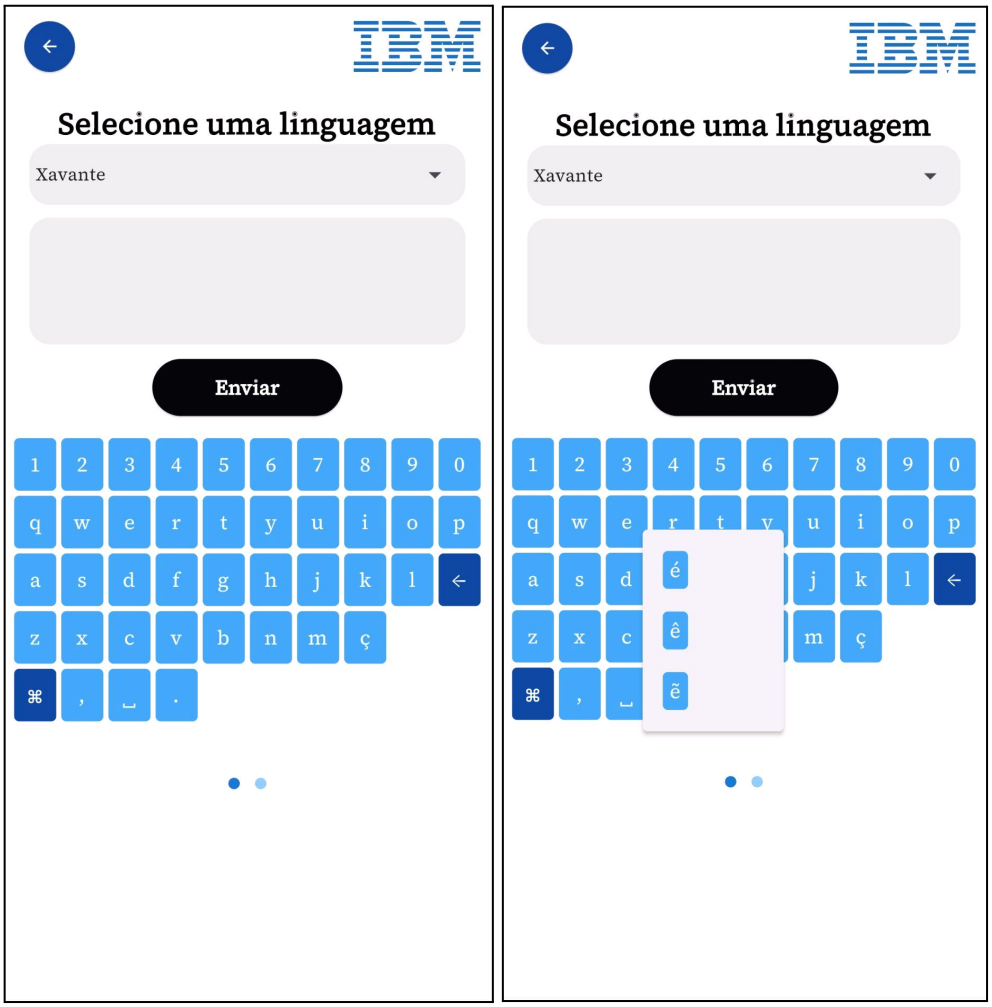
Figura 12 - Versão final da tela de seleção da língua no aplicativo mobile



Fonte: Elaborado pelos autores

Por fim, após completar os passos descritos anteriormente, o usuário terá acesso à última funcionalidade do aplicativo, o teclado mapeado pelos linguistas. Ao utilizá-lo, os caracteres digitados aparecem no campo de texto e ao selecionar o botão de enviar, o conteúdo escrito é enviado ao banco de dados, exibindo uma mensagem de sucesso. Os principais aprimoramentos relacionados à essa tela em comparação com a versão inicial são: o design das teclas e o acréscimo de uma nova página; para se assemelhar ao teclado nativo e dar suporte ao mapeamento feito na aplicação web; a adição do scroll no campo de texto, para que seja possível visualizar todo o texto digitado; e a funcionalidade de utilizar outros caracteres dentro das letras ao segurá-las. A tela e suas melhorias podem ser visualizadas na figura 13.

Figura 13 - Versão final da tela com o teclado mapeado no aplicativo mobile



Fonte: Elaborado pelos autores

4. Conclusões e Trabalhos Futuros

O projeto teve como objetivo o desenvolvimento de uma plataforma experimental de teclado virtual para línguas indígenas em dispositivos Android, com a finalidade de possibilitar a digitação para usuários indígenas na sua língua materna. A iniciativa visa oferecer uma solução para desafios de acessibilidade digital enfrentados por essa comunidade, promovendo o aumento de registros escritos das línguas indígenas e, principalmente, preservando a diversidade cultural dessas populações.

O produto entregue atendeu aos requisitos estabelecidos pela IBM, utilizando frameworks open source, o serviço Cloudant para o banco de dados e bibliotecas com licenças

permitidas para o desenvolvimento das aplicações web e mobile. A plataforma foi projetada para suportar a adição de novas línguas conforme necessário pela IBM , à medida que novos dados forem estudados e estiverem disponíveis. Ainda, é possível que a IBM no futuro implemente um de seus projetos idealizados, ao desenvolver modelos de IA e inseri-los no aplicativo mobile, criando um chatbot para a interação com os indígenas. Os próximos passos nas semanas seguintes à entrega envolvem o deploy das aplicações, com base nos requisitos a serem definidos pela IBM. Além disso, está previsto realizar testes com linguistas para avaliar as aplicações e incorporar melhorias com base nos feedbacks obtidos durante os testes e nas apresentações finais, tanto para a banca do Insper, quanto para a Falconi.

Como parte das melhorias planejadas, destaca-se a implementação dos teclados de forma nativa no sistema Android, a exploração de uma possível versão para iOS, a inclusão do aplicativo na loja na Google Play e a ampliação dos testes, abrangendo comunidades indígenas e mais linguistas. Essas ações visam aprimorar não só a usabilidade e experiência do usuário, mas também a disponibilidade do teclado virtual, ampliando seu impacto e alcance entre os usuários indígenas.

Referências

ASSESSORIA DE COMUNICACÃO/FUNAI. Brasil registra 274 línguas indígenas diferentes faladas por 305 etnias. 27/10/2022. Disponível em: https://www.gov.br/funai/pt-br/assuntos/noticias/2022-02/brasil-registra-274-linguas-indigenas-diferentes-faladas-por-305-etnias. Acesso em: 15/09/2023.

CENSO 2022, IBGE. Brasil tem 1,7 milhão de indígenas e mais da metade deles vive na Amazônia Legal. 24/08/2023. Disponível em: https://censo2022.ibge.gov.br/noticias-por-estado/37565-brasil-tem-1-7-milhao-de-indigenas-e-mais-da-metade-deles-vive-na-amazonia-legal. Acesso em: 15/09/2023.

CUPERTINO, E. Jovens pesquisadores criam teclado digital para 40 línguas indígenas. 19/09/2022. 203 – 2021 p. Disponível em: https://agenciabrasil.ebc.com.br/radioagencia-nacional/pesquisa-e-inovacao/audio/2022-09/pesquisadores-criam-teclado-digital-para-40-linguas-indigenas. Acesso em: 13/09/2023.

DAS, Bijay Kumar. A handbook of translation studies. Nova Delhi, Índia: Atlantic, 2008. Acesso em: 13/09/2023

DIETZSCH, R. A tipografia das línguas indígenas brasileiras. 2021. 454 p. Tese (Programa de Pós-Graduação em Comunicação) — Universidade de Brasília. Disponível em: https://repositorio.unb.br/bitstream/10482/42450/1/2021_RafaelDietzsch.pdf. Acesso em: 13/09/2023.

LÍNGUAS indígenas brasileiras podem ser utilizadas em smartphones Motorola. 25/03/2021. Disponível em: https://g1.globo.com/economia/tecnologia/noticia/2021/03/25/línguas-indígenas-brasileiras-kaingang-e-nheengatu-podem-ser-utilizadas-em-smartphones-motorola.html. Acesso em: 13/09/2023.

SEVERO, C. G.; QUERIQUELLI, L. H. O papel das traduções na historiografia das línguas. Confluência, n. 43, p. 203 – 2021, 07 2012. Disponível em: https://confluencia.emnuvens.com.br/rc/article/view/633/401. Acesso em: 14/09/2023.

NETO, Maria Gorete. A Década Internacional das Línguas Indígenas e as línguas em uso pelos povos indígenas brasileiros: contribuições da Linguística Aplicada. 19/12/2022. Disponível em: https://www.scielo.br/j/delta/a/zQLRgMwXjWRKjs7QHfJskpq/#. Acesso em: 10/11/2023.

BRASILEIRO, Roberto. Sprint Scrum: Ciclo de desenvolvimento. Disponível em:https://www.metodoagil.com/sprint-scrum/#:~:text=Sprint%20Scrum%20%C3%A9%20o%20nome,devem%20ter%20a%20mesma%20dura%C3%A7%C3%A3o. Acesso em: 20/11/2023.

LOSNAK, Giulia. Metodologia Ágil - O que é? 04/07/2023. Disponível em: https://www.alura.com.br/artigos/o-que-e-metodologia-agil#:~:text=A%20metodologia%20%C3%A1gil%20%C3%A9%20uma,longo%20do%20desenvolvimento%20de%20projetos. Acesso em: 20/11/2023.

O que é o Scrum? Disponível em: https://aws.amazon.com/pt/what-is/scrum/#:~:text=adotar%20o%20DevOps%3F-,O%20que%20%C3%A9%20o%20Scrum%3F,uma%20entrega%20eficiente%20de%20projetos. Acesso em: 20/11/2023.