

# UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Camila Pereira da Silva  
Cintia Izumi Shinoda  
Fernando Miguel Escribano Martinez  
Henrique Yassuyuki Tsuboi  
Pedro Henrique Faria Cruz  
Rogério Gonçalves da Silva  
Vitor Paes de Barros  
Willy Paulino de Oliveira Gomes

## **Tecnologia no Ensino de Kanji: Desenvolvimento de um Aplicativo Interativo para o Aprendizado e Prática de Kanji**

<b>Repositório do Projeto no GitHub</b> <a href="https://github.com/cintia-shinoda/projeto-integrador-3">https://github.com/cintia-shinoda/projeto-integrador-3</a>
<b>Vídeo de apresentação do Projeto Integrador</b> <a href="https://youtu.be/HCylGe78tdg?si=Crdk38W7GrX47R67">https://youtu.be/HCylGe78tdg?si=Crdk38W7GrX47R67</a>
<b>Site / App</b> <a href="https://kanji-web-deploy.vercel.app/">https://kanji-web-deploy.vercel.app/</a>

São Paulo - SP  
2025

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

**Tecnologia no Ensino de Kanji: Desenvolvimento de um Aplicativo  
Interativo para o Aprendizado e Prática de Kanji**

Relatório Técnico-Científico apresentado na  
disciplina de Projeto Integrador em  
Computação III para os cursos de Bacharelado  
em Ciência de Dados, Tecnologia da  
Informação e Engenharia de Computação da  
Universidade Virtual do Estado de São Paulo  
(UNIVESP).

São Paulo - SP  
2025

BARROS, Vitor Paes de; CRUZ, Pedro Henrique Faria; GOMES, Willy Paulino de Oliveira; MARTINEZ, Fernando Miguel Escribano; SHINODA, Cintia Izumi; SILVA, Rogério Gonçalves da; SILVA, Camila Pereira da; TSUBOI, Henrique Yassuyuki. **Tecnologia no Ensino de Kanji: Desenvolvimento de um Aplicativo Interativo para o Aprendizado e Prática de Kanji**. 00f. Relatório Técnico-Científico. Bacharelado em Ciência de Dados, Tecnologia da Informação e Engenharia de Computação – **Universidade Virtual do Estado de São Paulo**. Tutor: Dhyogo Nunes Costa. Polos: Parque Bristol, Jaguaré, Heliópolis, 2025.

## RESUMO

O presente trabalho partiu da crescente relevância das tecnologias educacionais no ensino de idiomas, especialmente do japonês. O objetivo principal foi desenvolver um aplicativo educacional interativo que auxiliasse no processo de aprendizagem e treino da escrita dos kanjis, promovendo um ambiente adaptativo e dinâmico. A metodologia adotada foi o *Design Thinking*, incluindo a criação de uma persona representativa do público-alvo, prototipagem das telas com foco na usabilidade e incorporação de elementos gamificados. A implementação técnica utilizou Flutter no *front-end*, Node.js no *back-end* e MongoDB Atlas no banco de dados. A gestão do projeto foi realizada com Kanban, permitindo organização e acompanhamento visual das etapas de desenvolvimento do projeto. Como resultado, foi desenvolvido um protótipo funcional com cadastro e *login* de usuários, exibição dos kanjis com suas respectivas leituras e traduções, treino da ordem correta dos traços com validação em tempo real e modo de alto contraste para acessibilidade. Conclui-se que o uso de *Design Thinking* aliado a práticas ágeis viabilizou a criação e desenvolvimento de uma solução educacional centrada no usuário, com potencial de expansão e aplicação em contextos reais de aprendizado do idioma japonês.

**PALAVRAS-CHAVE:** Aplicativo; Tecnologia; Aprendizado da língua japonesa; Kanjis; Banco de dados.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Persona .....	28
Figura 2 - Kanban.....	29
Figura 3 - Protótipo inicial das telas do aplicativo.....	30
Figura 4 - Protótipo das telas do aplicativo em estágio mais desenvolvido.....	30
Figura 5 - Diagrama UML de Casos de Uso.....	33
Figura 6 - Arquitetura do Projeto .....	34
Figura 7 - Estrutura do Projeto .....	35
Figura 8 - Pasta <i>back-end</i> .....	37
Figura 9 - Pasta <i>lib</i> .....	38
Figura 10 - Modelo Entidade-Relacionamento .....	39
Figura 11 - Banco de Dados MongoDB Atlas .....	40
Figura 12 - Repositório do KanjiVG.....	41
Figura 14 - JSON do kanji 「一」 .....	43
Figura 15 - Resultado do teste de Cadastro de Usuário no Postman.....	44
Figura 16 - Resultado do teste de listagem dos usuários cadastrados no Postman .....	45
Figura 17 - Resultado do teste de login no Postman .....	46
Figura 18 - Resultado do teste de cadastro de kanji no Postman .....	47
Figura 19 - Resultado do teste de listagem de kanjis cadastrados no banco de dados no Postman .....	48
Figura 20 - Resultado do teste unitário de validação de <i>login</i> .....	49
Figura 21 - Resultado do teste de <i>widget</i> implementado .....	50
Figura 22 - <i>Workflow</i> "Flutter CI" no GitHub Actions .....	51
Figura 23 - Captura de tela do repositório do projeto no GitHub .....	52
Figura 24 - Lista de <i>commits</i> realizados e listados pelo Git Graph.....	53
Figura 25 - <i>Dashboard</i> no Render com os logs do <i>deploy</i> .....	56
Figura 26 - Mensagem no navegador que indica que a API está ativa .....	57
Figura 27 - Resultado do teste realizado no Postman da listagem do kanji cadastrado.....	58
Figura 28 - Resultado do teste realizado no Postman da listagem dos usuários cadastrados ..	59
Figura 29 - <i>Dashboard</i> do <i>deploy</i> no Vercel.....	60

## LISTAS DE TABELAS

Tabela 1 - Principais <i>endpoints</i> da API.....	37
Tabela 2 - Tabela comparativa entre telas com o tema padrão e telas com o tema com a funcionalidade de alto contraste ativada. ....	53

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>9</b>
<b>2 DESENVOLVIMENTO .....</b>	<b>11</b>
2.1 Objetivos .....	11
2.1.1 Objetivo Geral .....	11
2.1.2 Objetivos Específicos .....	11
2.2 Justificativa e Delimitação do Problema .....	13
2.2.1 O Uso de Aplicativos como Recurso Pedagógico para Aprender Língua Japonesa .....	13
2.2.2 Aliança Cultural Brasil-Japão e os Benefícios que a elaboração de um Aplicativo pode significar à Comunidade .....	15
2.3 Fundamentação Teórica .....	18
2.3.1 Porque a Escolha do Framewok Flutter .....	18
2.3.2 Usos do Flutter .....	19
2.3.3 <i>Design Thinking</i> e MVP para Projetos de Software .....	20
2.3.4 Produto Mínimo Viável (MVP) .....	20
2.3.5 Integração de <i>Design Thinking</i> E MVP .....	21
2.3.6 Importância da Combinação <i>Design Thinking</i> + MVP .....	21
2.3.7 Meta Atingida .....	21
2.4 Aplicação das Disciplinas Estudadas no Projeto Integrador .....	23
2.4.1 Banco de Dados .....	23
2.4.2 Gestão da Inovação e Desenvolvimento de Produtos .....	23
2.4.3 Desenvolvimento Web .....	24
2.4.4 Interface Humano-Computador .....	25
2.4.5 Gerência e Qualidade de Software .....	25
2.5 Metodologia .....	26
2.5.1 Etapa de Empatia .....	26
2.5.2 Etapa de Definição .....	27

2.5.4 Etapa de Prototipação.....	29
2.5.5 Etapa de Testes.....	31
<b>3 RESULTADOS: SOLUÇÃO FINAL .....</b>	<b>31</b>
3.1 Definição da Solução .....	31
3.1.1 Ferramentas .....	32
3.1.2 Funcionalidades.....	32
3.2 Arquitetura e Estrutura do Projeto .....	34
3.2.1 <i>Back-End</i> .....	36
3.2.2 <i>Front-End</i> .....	38
3.3 Dados.....	39
3.3.1 Banco De Dados.....	39
3.3.2 KanjiVG .....	40
3.3.3 Estrutura de Dados .....	41
SVG.....	41
JSON .....	43
3.4 Testes.....	43
3.4.1 Testes de API .....	43
3.4.2 Testes Unitários.....	48
3.4.3 Testes de <i>Widgets</i> .....	49
3.5 Integração Contínua .....	50
3.6 Versionamento de Código.....	51
3.7 Acessibilidade .....	53
3.8 <i>Deploy</i> .....	55
3.8.1 <i>Back-End</i> .....	55
3.8.2 <i>Front-End</i> .....	59
<b>4 CONSIDERAÇÕES FINAIS .....</b>	<b>61</b>
<b>REFERÊNCIAS .....</b>	<b>62</b>

**ANEXO A ..... 65**

**ANEXO B..... 68**

**APÊNDICE A..... 73**

**APÊNDICE B ..... 74**



## 1 INTRODUÇÃO

A língua japonesa possui uma longa história e a sua estrutura foi moldada por diferentes influências culturais e linguísticas, um dos aspectos mais notáveis e, ao mesmo tempo, mais desafiadores dessa língua é o seu sistema de escrita tríplice formado pelos silabários hiragana, katakana e kanji. Esse sistema torna o aprendizado desafiador, principalmente para estrangeiros, por sua complexidade, exigindo do indivíduo a memorização de milhares de caracteres, o conhecimento da ordem de traços e o entendimento das leituras múltiplas atribuídas a cada elemento.

A origem do sistema de escrita japonês está profundamente enraizada na influência cultural chinesa, segundo Miller (1967) os caracteres kanji chegaram ao Japão por volta do século V, sendo gradativamente adaptados às características fonéticas e gramaticais exclusivas do idioma japonês. Em paralelo, os silabários **hiragana** e **katakana** foram desenvolvidos como soluções complementares ao uso dos kanji: enquanto o primeiro se destaca na representação de partículas gramaticais e palavras de leitura facilitada, o segundo se consolidou como ferramenta para escrever termos estrangeiros e nomes próprios (Shibatani, 1990).

DeFrancis (1984) argumenta que o desafio imposto pelo sistema não se restringe ao volume de caracteres, estimado em mais de dois mil somente para uso cotidiano, mas reside também em sua complexidade multifacetada, pois cada caractere kanji carrega múltiplas leituras e significados, divididos entre suas acepções de origem chinesa (*on'yomi*) e nativas japonesas (*kun'yomi*), além de dependências contextuais.

Outro ponto importante, mencionado por Kano (1995), diz respeito à memorização das formas e à ordem correta dos traços dos caracteres kanji, elemento que usualmente frustra alunos habituados a estruturas alfabéticas mais simples.

Com as inovações advindas das tecnologias digitais o ensino de línguas, especificamente da língua japonesa, passou a contar com ferramentas que ampliam as possibilidades pedagógicas. Aplicativos e dispositivos móveis oferecem funções interativas, como o reconhecimento de escrita e o fornecimento de *feedback* imediato, recursos que, segundo Chun (2011), podem

otimizar consideravelmente o processo de aprendizado. Ainda, Mori e Shimizu (2017) destacam que a adoção da gamificação e de ferramentas visuais, como exercícios interativos, têm demonstrado impacto positivo, promovendo maior engajamento e precisão no reconhecimento e reprodução de caracteres.

Nesse cenário, desenvolveu-se um aplicativo especializado voltado para o ensino dos kanji em parceria com a *Aliança Cultural Brasil-Japão*, com foco em três dimensões fundamentais: a ordem dos traços, a associação entre forma e significado (compreensão), e a correta pronúncia (leitura).

Futuras investigações nesse campo devem buscar não somente explorar a eficácia de aplicativos e metodologias específicas, mas também seus impactos entre diferentes perfis de aprendizes e contextos culturais. Ao combinarmos tradição e inovação, esperamos formular abordagens mais inclusivas e personalizadas, levando o ensino da língua japonesa e de seus encantos e desafios a um público ainda maior.

## 2 DESENVOLVIMENTO

### 2.1 OBJETIVOS

#### 2.1.1 OBJETIVO GERAL

O objetivo do presente projeto foi desenvolver um aplicativo interativo, por meio de exercícios, jogos e *quizzes* que facilitassem o aprendizado, memorização e aquisição da habilidade necessária para compreender e escrever kanjis.

#### 2.1.2 OBJETIVOS ESPECÍFICOS

Criar um aplicativo interativo voltado para o aprendizado, memorização e desenvolvimento de habilidades em leitura e escrita de kanjis exige a definição de metas específicas que guiem o desenvolvimento das funcionalidades necessárias. Esses objetivos são classificados em três categorias: investigativos, descritivos e analíticos, conforme a natureza das ações a serem realizadas.

Buscamos explorar e mapear os elementos essenciais para a concepção do aplicativo, considerando as necessidades dos usuários e as características do aprendizado de kanjis. Nesta fase, procura-se:

- Identificar os principais desafios enfrentados pelos estudantes de japonês na memorização e escrita de kanjis.
- Analisar os métodos de ensino mais eficazes para caracteres ideográficos, com uma ênfase em gamificação e exercícios de repetição espaçada.
- Compreender as expectativas dos usuários em relação a um aplicativo de aprendizado de kanjis, incluindo preferências por *quizzes*, jogos interativos ou desafios de escrita.

Já com relação aos objetivos descritivos, o foco nesta etapa foi buscar um detalhamento das funcionalidades do aplicativo e a organização do sistema de aprendizado. Os objetivos incluíram:

- Descrever as etapas de desenvolvimento do aplicativo, desde a escolha dos kanjis por nível de proficiência (N5 a N1) até a implementação de *feedbacks* instantâneos.

- Identificar as técnicas de gamificação a serem incorporadas, como sistemas de recompensas, rankings e desafios diários.
- Definir a estrutura dos exercícios de fixação, incluindo práticas de traços, associação de significados e formação de sentenças.
- Desmembrar o kanji em radicais e partículas, agrupá-los em ideogramas com radicais comuns e a inclusão de vocabulário composto por kanji como *jukugo* e especialmente *yojijukugo* (palavras formadas por quatro kanjis).

Por último, consideramos os objetivos de análise visando examinar e comprovar a eficácia do aplicativo, garantindo que ele atenda ao seu propósito educacional. Para isso, propõe-se:

- Analisar a retenção de kanjis pelos usuários após o uso do aplicativo, comparando diferentes métodos (*flashcards* vs. jogos interativos).
- Avaliar a usabilidade da ferramenta, verificando se a interface intuitiva e os exercícios progressivos facilitam o aprendizado.
- Explicar como a repetição espaçada e a contextualização em sentenças contribuem para a memorização a longo prazo.

#### Justificativa da Metodologia:

Dividir os objetivos em investigativos, descritivos e analíticos proporciona uma abordagem metodológica clara, assegurando que o aplicativo seja desenvolvido com base em evidências pedagógicas e feedback dos usuários. Enquanto os objetivos investigativos garantem que o produto atenda às necessidades reais dos aprendizes, os descritivos organizam sua estrutura, e os analíticos validam sua eficácia. Dessa forma, o aplicativo não somente ensina kanjis de forma divertida e interativa, mas também promove uma aquisição sólida e duradoura do conhecimento.

## 2.2 JUSTIFICATIVA E DELIMITAÇÃO DO PROBLEMA

### 2.2.1 O USO DE APLICATIVOS COMO RECURSO PEDAGÓGICO PARA APRENDER LÍNGUA JAPONESA

A utilização de aplicativos como recurso pedagógico no aprendizado da língua japonesa tem se mostrado uma ferramenta valiosa e eficaz. Com a crescente disponibilidade de smartphones e tablets, os aplicativos oferecem uma maneira conveniente e acessível de aprender japonês, adaptando-se a diferentes estilos de aprendizado e níveis de proficiência. A tecnologia favorece a dinâmica da aprendizagem e permite aos discentes o aproveitamento do tempo para ampliar ainda mais a interação com a língua japonesa. Portanto, o uso de aplicativos móveis no processo de aprendizagem funciona como mídias digitais e auxilia na prática.

Vantagens do uso de aplicativos no aprendizado de japonês:

- Flexibilidade e acessibilidade: os aplicativos permitem que os alunos aprendam no seu próprio ritmo e em qualquer lugar, aproveitando momentos livres para praticar. A possibilidade de estudar em qualquer lugar e a qualquer momento permite que os alunos integrem o aprendizado do japonês em suas rotinas diárias, promovendo uma aprendizagem contínua. Alguns aplicativos funcionam offline, facilitando o uso em qualquer lugar.
- Interatividade e engajamento: muitos aplicativos utilizam jogos, *quizzes* e outras atividades interativas para tornar o aprendizado mais divertido e envolvente. Aplicativos como o Duolingo utilizam elementos de gamificação, tornando o aprendizado mais dinâmico e motivador. Os aplicativos promovem aprendizado ativo mediante testes, jogos e atividades práticas.
- Variedade de recursos: os aplicativos oferecem uma ampla gama de recursos, incluindo vocabulário, gramática, kanji, áudio e exercícios de conversação. Ferramentas como o LingoDeer oferecem lições estruturadas que abrangem gramática, vocabulário e prática de pronúncia, atendendo a diferentes aspectos da língua japonesa.
- Personalização: Alguns aplicativos permitem que os alunos personalizem seu aprendizado, focando em áreas específicas que precisam de mais atenção. Muitos permitem que os usuários adaptem o conteúdo ao seu nível e ritmo de estudo.
- *Feedback* imediato: os aplicativos dão *feedback* instantâneo sobre o desempenho dos alunos, ajudando-os a identificar seus pontos fortes e fracos.
- Despertar o interesse dos alunos.

- Promover a colaboração entre os alunos.
- Melhorar a comunicação entre alunos e professores.
- Desenvolver habilidades práticas.
- Proporcionar acesso a novos métodos de ensino.
- Oferecer conteúdo e materiais de estudo disponíveis 24h por dia.
- Proporcionar mais flexibilidade e autonomia.
- Aplicativos como JA Sensei também exploram aspectos culturais do Japão, enriquecendo a experiência educativa.

Exemplos de aplicativos populares para aprender japonês:

- Duolingo: Um aplicativo popular que oferece um curso completo de japonês, com foco em vocabulário e gramática. É conhecido por sua abordagem lúdica e é ideal para iniciantes que desejam construir uma base sólida no idioma. No entanto, pode ter limitações na gramática avançada.
- Lingodeer: Um aplicativo que oferece lições detalhadas de gramática e exercícios de escrita e leitura. Focado em línguas asiáticas, oferece lições detalhadas e estruturadas, abrangendo desde o básico até níveis mais avançados. Utiliza gravações em HD feitas por nativos e atividades como *flashcards* e questionários para revisão.
- Mondly: Um aplicativo que se concentra em conversação, com lições diárias e exercícios interativos.
- Kanji Study: Um aplicativo dedicado ao aprendizado de kanji, com exercícios de escrita e reconhecimento.
- Bunpro: Um aplicativo que foca no aprendizado de gramática, com explicações detalhadas e exercícios práticos.
- Wanikani: Um aplicativo que utiliza um sistema de repetição espaçada para ajudar os alunos a memorizar kanji e vocabulário.
- JA Sensei: Abrange aspectos orais e escritos do japonês, incluindo gramática, vocabulário e cultura japonesa. Oferece testes de diferentes níveis e módulos interativos para reforçar a pronúncia e escrita.
- Hiragana Quest: Ensina Hiragana e Katakana com mascotes e mnemônicos. Inclui guias para desenhar caracteres e testes interativos.

- **Memrise:** Combina vídeos de falantes nativos com exercícios de memorização. Trabalha vocabulário prático usado no cotidiano, ideal para turistas.
- **Obenkyo:** Utiliza cartas para memorização dos *kanas*, repetindo os mais difíceis até que o usuário os assimile completamente. Auxilia na familiarização com os alfabetos japoneses, oferecendo ferramentas para o aprendizado de hiragana, katakana e kanji.

Considerações pedagógicas importantes:

- É importante escolher aplicativos que se adequem ao seu nível de proficiência e estilo de aprendizado.
- Os aplicativos devem ser utilizados como um complemento ao aprendizado tradicional, como aulas presenciais ou online. A integração de aplicativos móveis no ensino de línguas pode enriquecer o processo educacional, caso sejam utilizados de forma complementar e planejada.
- A prática regular é essencial para alcançar fluência em japonês.
- A necessidade de assinaturas ou restrições a certas plataformas pode limitar o acesso completo aos recursos disponíveis.

Em resumo, o uso de aplicativos no ensino da língua japonesa é uma ferramenta eficaz que combina tecnologia com pedagogia moderna. Eles são especialmente úteis para iniciantes que buscam aprender de forma prática e divertida, mas também podem complementar estudos avançados com recursos específicos como gramática ou caligrafia. Os aplicativos móveis representam ferramentas valiosas no ensino e aprendizagem da língua japonesa, proporcionando recursos interativos e acessíveis que podem complementar métodos tradicionais e enriquecer a experiência educacional.

## 2.2.2 ALIANÇA CULTURAL BRASIL-JAPÃO E OS BENEFÍCIOS QUE A ELABORAÇÃO DE UM APLICATIVO PODE SIGNIFICAR À COMUNIDADE

A **Aliança Cultural Brasil-Japão (ACBJ)** é uma entidade sem fins lucrativos fundada em 1956 pelo poeta Guilherme de Almeida com o objetivo de promover o intercâmbio cultural entre o Brasil e o Japão. A Aliança é reconhecida como a maior escola de língua japonesa da América Latina e desempenha um papel essencial na difusão da cultura japonesa e brasileira. A importância da Aliança Cultural Brasil-Japão é multifacetada:

- Fortalecimento das relações bilaterais: A ACBJ atua como um elo cultural entre Brasil e Japão, contribuindo para a amizade e cooperação entre os países. Ela promove o entendimento mútuo e o respeito entre as culturas brasileira e japonesa.
- Preservação e divulgação da cultura japonesa: A Aliança realiza diversas atividades para preservar e divulgar a cultura japonesa, oferecendo cursos de artes japonesas como origami, culinária japonesa e artes tradicionais. Organiza também eventos culturais diversos, como exposições, palestras, festivais e apresentações.
- Educação de qualidade: A instituição desenvolveu métodos exclusivos para o ensino do japonês, como o uso do material didático "Marugoto", que combina língua e cultura. Além do ensino do idioma, oferece cursos de português para estrangeiros. A Aliança é uma das cinco associações japonesas no Brasil reconhecidas pelo governo japonês, evidenciando sua relevância e credibilidade. Ao longo de sua história, já formou mais de 50 mil alunos.

O ensino da língua japonesa no Brasil tem uma história que se inicia com a imigração japonesa em 1908. A Aliança Cultural Brasil-Japão se destaca nesse cenário como referência no ensino do japonês, oferecendo cursos presenciais e online que atendem tanto descendentes quanto pessoas sem ascendência japonesa. Atualmente, mais de 40% dos alunos não possuem origem japonesa, refletindo o crescente interesse pela língua e cultura no Brasil. A entidade modernizou suas metodologias com tecnologias de ensino à distância (EAD), ampliando o acesso ao aprendizado.

A Aliança oferece cursos para todos os níveis, desde iniciantes até avançados. Utiliza metodologias inovadoras e materiais didáticos atualizados que facilitam o aprendizado da língua japonesa. Oferece cursos de japonês para diversos públicos, incluindo crianças, jovens, adultos e empresas.

O impacto do ensino vai além da língua. Segundo Noriteru Fukushima, côsul-geral do Japão em São Paulo, aprender japonês na Aliança aumenta o interesse pela cultura japonesa e fortalece as relações bilaterais. O Dr. Anselmo Nakatani ressalta que o aprendizado inclui valores culturais como disciplina e respeito, essenciais para a comunicação com japoneses. Além disso, a instituição contribui para a formação de profissionais qualificados em língua japonesa, que podem atuar em diversas áreas, como tradução, interpretação, turismo e negócios. A Aliança oferece um espaço de encontro e intercâmbio para pessoas interessadas na cultura japonesa, fortalecendo os laços da comunidade nipo-brasileira. O ensino da língua japonesa no Brasil tem crescido cada vez mais, e a Aliança tem sido uma grande força nesse crescimento. Em resumo, a Aliança Cultural Brasil-Japão desempenha um papel crucial na promoção e preservação da



língua e cultura japonesas no Brasil, fortalecendo os laços entre as duas nações e contribuindo significativamente para o multiculturalismo e a educação no país. A Aliança possui unidades em Pinheiros e Liberdade na cidade de São Paulo.

A criação de um aplicativo de aprendizado de Kanji pode trazer inúmeros benefícios para a comunidade vinculada à Aliança Cultural Brasil-Japão (ACBJ), reforçando ainda mais seu papel como ponte cultural entre Brasil e Japão. O aprendizado dos Kanjis, que são caracteres fundamentais da língua japonesa, é uma parte crucial da proficiência no idioma e, conseqüentemente, da imersão na cultura japonesa. Aqui estão algumas maneiras pelas quais esse aplicativo pode ajudar:

- **Acesso Ampliado ao Ensino de Kanji** - com um aplicativo, o aprendizado de Kanji se torna mais acessível, permitindo que indivíduos de diferentes origens e com diferentes horários possam estudar no seu próprio ritmo. As aulas podem ser programadas para atender a todos os níveis, desde iniciantes até avançados, complementando os cursos oferecidos pela ACBJ.
- **Métodos Interativos e Engajadores** - o uso de tecnologias e métodos interativos, como jogos, *quizzes* e *flashcards*, pode tornar o aprendizado mais dinâmico e motivador. O aplicativo pode incluir recursos que incentivam a prática diária, facilitando a retenção de informações e aumentando a eficiência do aprendizado.
- **Integração da Cultura** – o ensino de Kanji pode ser integrado à cultura japonesa por meio de recursos que expliquem o significado cultural por trás de cada caractere. Isso ajuda os alunos a compreenderem o contexto cultural, promovendo uma aprendizagem mais significativa e enriquecedora.
- **Suporte Comunitário e Colaboração** - o aplicativo pode incluir fóruns ou salas de bate-papo, onde estudantes podem interagir, trocar experiências e tirar dúvidas uns com os outros. Isso não somente fortalece o aprendizado, mas também cria um senso de comunidade e pertencimento, alinhando-se à missão da ACBJ de promover o intercâmbio cultural.
- **Feedback Personalizado** - a implementação de um sistema de *feedback* personalizado pode auxiliar os alunos a identificarem suas dificuldades e a melhorarem suas habilidades de forma direcionada. A capacidade de seguir o progresso individual pode incentivar os alunos a persistirem no aprendizado.

**Fomento às Relações Bilaterais** - ao facilitar o aprendizado do Kanji, o aplicativo contribuirá para aumentar o interesse pela língua e cultura japonesa entre os brasileiros, fortalecendo ainda mais as relações bilaterais promovidas pela ACBJ. Estudantes mais preparados podem se tornar

embaixadores culturais, promovendo interações mais ricas e respeitosas entre brasileiros e japoneses.

A introdução de um aplicativo de aprendizado de Kanji pode não apenas enriquecer a experiência educacional dos alunos da Aliança Cultural Brasil-Japão, mas também contribuir para a preservação e disseminação da cultura japonesa no Brasil. Com a modernização das metodologias de ensino, a ACBJ pode continuar a desempenhar um papel fundamental na formação de cidadãos plurais, que compreendem e respeitam a diversidade cultural.

## 2.3 FUNDAMENTAÇÃO TEÓRICA

### 2.3.1 PORQUE A ESCOLHA DO FRAMEWOK FLUTTER

Flutter é um framework de código aberto criado pelo Google, lançado em 2015 e baseado na linguagem Dart. Voltado para o desenvolvimento de aplicativos multiplataforma, abrangendo Android, iOS, web e desktop, o Flutter tem se destacado no mercado em virtude de sua versatilidade, desempenho superior e eficiência (Bai, 2020; Reddy, 2021).

A utilização do Flutter traz diversas vantagens significativas, que o tornam uma escolha atraente para desenvolvedores e empresas:

- **Desenvolvimento Multiplataforma:** O Flutter permite a criação de aplicativos para múltiplas plataformas a partir de um único código-base. Essa capacidade resulta em uma redução significativa nos custos e no tempo de desenvolvimento, já que não é necessário escrever códigos separados para cada plataforma (Chen et al., 2022).
- **Performance Elevada:** Com a linguagem Dart e a compilação antecipada (AOT) para código nativo, o Flutter garante que os aplicativos sejam rápidos e responsivos, apresentando desempenho semelhante ao de aplicativos desenvolvidos especificamente para cada plataforma. Diferentemente de outros frameworks, ele não depende de uma ponte JavaScript, o que minimiza a sobrecarga e melhora a velocidade de execução (Gonzalez et al., 2021).
- **Hot Reload:** Esta funcionalidade permite que os desenvolvedores visualizem alterações no código em tempo real, sem a necessidade de reiniciar o aplicativo, acelerando significativamente o processo de desenvolvimento e facilitando a experimentação e correção de erros (Kurtz et al., 2021).

- **Widgets Personalizáveis:** O Flutter oferece uma ampla gama de widgets personalizáveis, que permitem a criação de interfaces de usuário atraentes e adaptadas às diretrizes de *design* do Android (*Material Design*) e iOS (Cupertino). Esta arquitetura baseada na composição possibilita a construção de interfaces complexas a partir de elementos simples (Huang & Zhang, 2020).
- **Curva de Aprendizado Suave:** O Flutter é reconhecido por ser intuitivo, facilitando o aprendizado simultâneo da ferramenta e da linguagem Dart (Prasad, 2022).
- **Custo-Benefício:** A implementação de um único código-base para múltiplas plataformas possibilita economizar recursos, permitindo que equipes menores desenvolvam aplicativos de alta qualidade (Sinha, 2021).
- **Suporte do Google:** O suporte contínuo do Google e a comunidade ativa ao redor do Flutter oferecem recursos online, como documentação, tutoriais e plugins, que facilitam a resolução de problemas e a implementação de novas funcionalidades (Lee, 2021).
- **Integração com Firebase:** O Flutter se integra facilmente com o Firebase, permitindo adicionar funcionalidades como autenticação, armazenamento em nuvem e análise de dados de forma rápida e eficiente (Mishra et al., 2021).

### 2.3.2 USOS DO FLUTTER

O Flutter é considerado ideal para diversos tipos de aplicações, que incluem:

- Aplicativos móveis (iOS e Android) com interfaces ricas e interativas.
- Aplicativos empresariais, como ferramentas de produtividade e gerenciamento interno.
- Desenvolvimento rápido de protótipos.
- Criação de MVPs (*Minimum Viable Product*) (Nimavati, 2023).

Em suma, o Flutter se apresenta como uma solução eficiente para empresas que necessitam desenvolver aplicativos multiplataforma com rapidez, qualidade e baixo custo. Sua arquitetura robusta e o suporte contínuo do Google asseguram um futuro promissor no mercado de tecnologia.

### 2.3.3 DESIGN THINKING E MVP PARA PROJETOS DE SOFTWARE

*Design Thinking* e Produto Mínimo Viável (MVP) são metodologias que se complementam no desenvolvimento de software, unindo a criatividade centrada no usuário à validação prática e ágil de ideias. Essa combinação se mostra eficaz na criação de soluções inovadoras, redução de riscos e garantia de que o produto final atenda às reais necessidades do mercado (Brown, 2008; Ries, 2011).

O *Design Thinking* é uma abordagem que prioriza a solução de problemas com foco no ser humano, buscando compreender profundamente as necessidades dos usuários. Suas etapas principais incluem:

1. Empatia: Entender o usuário por meio de pesquisas, entrevistas e imersão em seu contexto (Kolko, 2015).
2. Definição: Sintetizar insights da fase de empatia para identificar claramente o problema a ser resolvido (Liedtka, 2015).
3. Ideação: Gerar um grande número de ideias inovadoras para solucionar o problema definido (Brown, 2009).
4. Prototipagem: Criar representações tangíveis das ideias para testá-las rapidamente (Razzano et al., 2019).
5. Teste: Avaliar os protótipos com usuários reais, aprendendo o que funciona e o que não funciona (Martin, 2009).

### 2.3.4 PRODUTO MÍNIMO VIÁVEL (MVP)

O MVP é uma versão inicial do produto que contém apenas as funcionalidades essenciais para resolver um problema específico do usuário. Essa estratégia permite testar hipóteses e coletar *feedback* real com o menor investimento possível, concentrando-se no que realmente importa (Ries, 2011).

### 2.3.5 INTEGRAÇÃO DE *DESIGN THINKING* E MVP

A combinação de *Design Thinking* e MVP cria um ciclo valioso, onde o MVP é desenvolvido após as etapas iniciais do *Design Thinking*, resultando em um produto que é tanto inovador quanto validado por usuários reais (Graham, 2020). Este processo pode ser resumido em:

- Empatia + Definição (*Design Thinking*): Entender o usuário e definir problemas reais.
- Ideação (*Design Thinking*): Gerar soluções potenciais através de brainstorming.
- Prototipagem + Teste (*Design Thinking*) → MVP: Construir uma versão mínima a partir das ideias validadas.
- Validação (*Lean*): Testar o MVP com usuários, medir resultados e iterar com base no feedback (Gothelf, 2013).

### 2.3.6 IMPORTÂNCIA DA COMBINAÇÃO *DESIGN THINKING* + MVP

A sinergia entre essas metodologias maximiza a probabilidade de sucesso ao concentrar esforços em soluções que atendam às necessidades reais dos usuários. Essa combinação é essencial por diversos motivos:

- Foco no Usuário: Garante que o MVP atenda às necessidades reais dos clientes (Kelley & Kelly, 2013).
- Redução de Riscos: Permite validar hipóteses rapidamente, evitando erros ao alinhar o produto às expectativas dos usuários (Blank, 2013).
- Agilidade e Economia: Reduz custos e acelera o desenvolvimento ao priorizar funcionalidades essenciais (Meyer, 2021).

### 2.3.7 META ATINGIDA

A meta do projeto era criar um produto funcional que, com base nas abordagens de *Design Thinking* e MVP, resolvesse um problema real enfrentado pelos estudantes do idioma japonês. A seguir, descreve-se como os principais objetivos dessas abordagens foram alcançados:

- **Compreender as necessidades dos usuários:**

Através de conversas informais com alunos e durante as reuniões realizadas com a Aliança, realizou-se a etapa de empatia. Criou-se a persona Michiko Tomonaga, que sintetiza características comuns entre estudantes do idioma japonês. Essa representação guiou todo o processo de ideação e priorização de funcionalidades.

- **Criar soluções viáveis e desejáveis:**

Com base nas necessidades identificadas, desenvolveu-se um aplicativo com interface simples e acessibilidade, buscando atender tanto aos aspectos técnicos exigidos pelo Projeto Integrador 3, quanto às limitações e características dos usuários finais alvo.

- **Validar hipóteses testando funcionalidades essenciais do MVP:**

As principais funcionalidades - cadastro e login de usuários, visualização de leituras e tradução e treino responsivo de kanji - foram implementadas no MVP.

O protótipo foi testado com usuários, incluindo membros da diretoria da Aliança Cultural Brasil-Japão, que forneceram *feedback* positivo e sugestões construtivas.

- **Aprender sobre o mercado e comportamento do usuário através de *feedback*:**

A interação com usuários, em especial durante as reuniões com a Aliança, proporcionou insights sobre funcionalidades, como inclusão de *tags*, possibilidade de criação de roteiros de estudo e alinhamento com material didático, indicando caminhos para evolução futura da aplicação.

- **Iterar e refinar continuamente o produto:**

Ao longo do desenvolvimento, a aplicação passou por ajustes com base nos contínuos testes realizados, incluindo ajustes visuais e de usabilidade.

O uso de metodologia ágil (Kanban) permitiu rápida adaptação às correções e refinamento do *app*.

Em suma, a meta foi atingida com êxito: um MVP funcional foi desenvolvido, alinhado a uma abordagem centrada no usuário e validado com estes.

O projeto demonstrou viabilidade técnica, despertou interesse institucional para continuidade e estabeleceu bases sólidas para expansão futura.

## 2.4 APLICAÇÃO DAS DISCIPLINAS ESTUDADAS NO PROJETO INTEGRADOR

### 2.4.1 BANCO DE DADOS

A disciplina de Banco de Dados exerceu um papel fundamental no desenvolvimento do aplicativo voltado ao aprendizado e prática de Kanji, contribuindo diretamente para a estruturação, organização e gerenciamento das informações essenciais ao seu funcionamento. Optou-se pela utilização do sistema de gerenciamento de banco de dados MongoDB, em razão de sua arquitetura orientada a documentos e da flexibilidade na modelagem de dados. Essa escolha mostrou-se especialmente adequada para lidar com os diversos tipos de informações utilizados no projeto, tais como ideogramas, significados, leituras (*onyomi* e *kunyomi*), exemplos de aplicação e dados relacionados ao desempenho individual dos usuários.

A estrutura relacional proporcionada pelo MongoDB permitiu a criação de coleções específicas para os diferentes módulos do aplicativo, como usuários, sessões de estudo, atividades realizadas e conteúdos didáticos. A utilização do formato JSON para armazenamento dos documentos possibilitou uma modelagem mais dinâmica e adaptável, atendendo às necessidades do projeto sem comprometer a integridade dos dados. Além disso, essa abordagem facilitou a integração com os serviços da aplicação, promovendo maior agilidade nas operações de leitura, gravação e atualização das informações.

Adicionalmente, destaca-se a escalabilidade como uma das principais vantagens do MongoDB, o que torna o sistema preparado para futuras expansões, tanto no número de usuários quanto na adição de novos conteúdos. Os conhecimentos adquiridos na disciplina de Banco de Dados foram imprescindíveis para a modelagem lógica e física do banco, bem como para a definição de estratégias de indexação, segurança e performance. Dessa forma, o banco de dados desenvolvido não apenas viabilizou o funcionamento do aplicativo, como também garantiu uma base sólida e eficiente para a evolução contínua da solução tecnológica proposta.

### 2.4.2 GESTÃO DA INOVAÇÃO E DESENVOLVIMENTO DE PRODUTOS

O desenvolvimento do aplicativo voltado ao aprendizado e prática de Kanji está diretamente relacionado aos conteúdos abordados na disciplina de Gestão da Inovação e Desenvolvimento de Produtos. A proposta do projeto surgiu da identificação de uma necessidade concreta: a dificuldade de estudantes em memorizar e praticar ideogramas japoneses de maneira eficiente. A disciplina contribuiu para estruturar o processo de desenvolvimento de forma estratégica, desde a geração e análise de ideias até a definição do escopo do produto e suas funcionalidades mais relevantes.

Durante a execução do projeto, foram aplicados conceitos fundamentais como análise de mercado, segmentação de público-alvo, desenvolvimento de proposta de valor e prototipagem. A gestão adequada dessas etapas foi essencial para garantir que o aplicativo atendesse às expectativas dos usuários, oferecendo uma experiência prática, interativa e com potencial de engajamento contínuo. O uso de métodos como mapeamento de necessidades, priorização de funcionalidades e avaliação de viabilidade técnica também demonstrou a importância de um planejamento bem fundamentado para o sucesso do produto.

Além disso, a disciplina destacou a relevância da inovação não apenas como introdução de novas tecnologias, mas como a criação de soluções que geram valor para o usuário final. O projeto do aplicativo de Kanji exemplifica essa abordagem, ao propor uma ferramenta educacional acessível, com foco no aprendizado personalizado e no estímulo ao progresso contínuo. Dessa forma, a disciplina contribuiu significativamente para a condução do projeto com visão de mercado, foco no usuário e orientação para resultados sustentáveis.

#### 2.4.3 DESENVOLVIMENTO WEB

A disciplina de Desenvolvimento Web foi essencial para a realização das tarefas, pois forneceu os conhecimentos fundamentais para a construção de aplicações modernas, seguras e escaláveis. Durante o curso, aprendemos sobre a estrutura de aplicações web, os principais componentes envolvidos (como *front-end*, *back-end*, servidores e bancos de dados), além das diferenças entre aplicações web e sistemas tradicionais. No projeto, esses conhecimentos foram aplicados na escolha do Flutter para o desenvolvimento do *front-end* multiplataforma, no uso do Node.js para o *back-end* e na integração com o banco de dados MongoDB Atlas.



A disciplina também abordou arquiteturas como MVC e o padrão REST, que foram diretamente utilizados na implementação da API do aplicativo, permitindo a comunicação eficiente entre as diferentes camadas do sistema e garantindo a interoperabilidade com outros serviços. Além disso, o entendimento dos protocolos de comunicação, segurança e boas práticas de desenvolvimento web foi fundamental para garantir que o aplicativo fosse confiável, responsivo e acessível aos usuários, inclusive com recursos de acessibilidade como o modo de alto contraste.

#### 2.4.4 INTERFACE HUMANO-COMPUTADOR

A disciplina de Interface Humano-Computador (IHC) foi determinante para garantir a qualidade da experiência do usuário no aplicativo desenvolvido. Os conteúdos abordados, como princípios de usabilidade, *design* centrado no usuário, criação de personas, prototipagem, *wireframes* e testes de usabilidade, foram aplicados em todas as etapas do projeto. O uso do *Design Thinking*, também estudado na disciplina, permitiu compreender profundamente as necessidades dos usuários, resultando na criação de uma persona representativa e no desenvolvimento de uma interface simples, intuitiva e acessível, adequada até mesmo para pessoas com pouca familiaridade tecnológica.

A disciplina também destacou a importância de considerar aspectos cognitivos, emocionais e contextuais na interação com sistemas computacionais. Isso se refletiu na implementação de funcionalidades como feedback imediato, gamificação, modo de alto contraste e navegação facilitada, que aumentaram o engajamento e a inclusão dos usuários. O processo de prototipação e os testes de interface garantiram que as telas fossem claras e que as funcionalidades estivessem alinhadas às expectativas e limitações do público-alvo.

Portanto, os conhecimentos adquiridos em IHC foram fundamentais para transformar o aplicativo em uma ferramenta realmente útil, agradável e inclusiva, contribuindo diretamente para a eficácia pedagógica e o impacto social do projeto.

#### 2.4.5 GERÊNCIA E QUALIDADE DE SOFTWARE

A disciplina de Gerência e Qualidade de Software foi crucial para o planejamento, execução e manutenção da qualidade do projeto. Os conceitos de versionamento de código (Git e GitHub), integração contínua (GitHub Actions), automação de testes (unitários, de API e de

*widgets*), métricas de código, refatoração, revisão de código e *clean code* foram amplamente utilizados durante o desenvolvimento do aplicativo. Essas práticas garantiram a organização do código, a rastreabilidade das alterações, a detecção precoce de erros e a manutenção da qualidade do software ao longo do tempo.

Além disso, a disciplina enfatizou a importância de processos ágeis de desenvolvimento, como o uso do Kanban para gerenciamento das tarefas, priorização de demandas e acompanhamento do progresso do projeto. O controle da dívida técnica, a geração automática de documentação e a preocupação com a escalabilidade e a arquitetura do sistema também foram aspectos essenciais para assegurar que o produto final fosse sustentável e preparado para futuras expansões.

O foco em qualidade de software permitiu que o grupo entregasse um produto funcional, confiável e seguro, validado por meio de testes automatizados e revisões constantes, elevando o padrão do trabalho desenvolvido e aumentando suas chances de sucesso e aceitação pela comunidade-alvo.

## 2.5 METODOLOGIA

Para o desenvolvimento do presente projeto, adotou-se a abordagem do *Design Thinking*, por esta ser centrada no usuário e adequada à criação de soluções educacionais inovadoras. Essa abordagem orientou todas as etapas: empatia, definição, ideação, prototipação e testes.

### 2.5.1 ETAPA DE EMPATIA

A fase de empatia foi conduzida por meio de observação informal, conversas e escuta ativa de alunos que cursaram o curso de Kanji da Aliança Cultural Brasil-Japão e potenciais usuários do aplicativo.

Embora não tenha sido realizada uma pesquisa formal com instrumentos estruturados, foram coletadas percepções sobre as principais dificuldades enfrentadas no aprendizado de kanji, especialmente entre iniciantes e pessoas com pouco familiaridade tecnológica.

Essas interações revelaram a demanda de ferramentas acessíveis e intuitivas, que facilitassem a aprendizagem e o treino individual e contínuo da leitura e escrita dos kanjis.

### 2.5.2 ETAPA DE DEFINIÇÃO

Com base nos dados levantados, foi delimitado o problema central: a falta de um aplicativo focado em kanji, gratuito, acessível e em português que auxilie na memorização e prática da leitura e escrita de kanjis.

Neste momento, foi criado um perfil fictício (Persona) Michiko Tomonaga (Figura 1) que sintetiza os principais perfis observados.

Michiko é uma mulher de 65 anos, residente em São Paulo, com conhecimento básico de inglês e nível conversacional de japonês. Ela deseja ter e/ou desenvolver a habilidade de leitura de material no idioma japonês e ser capaz de se orientar de forma mais independente em uma possível viagem ao Japão.

Figura 1 - Persona



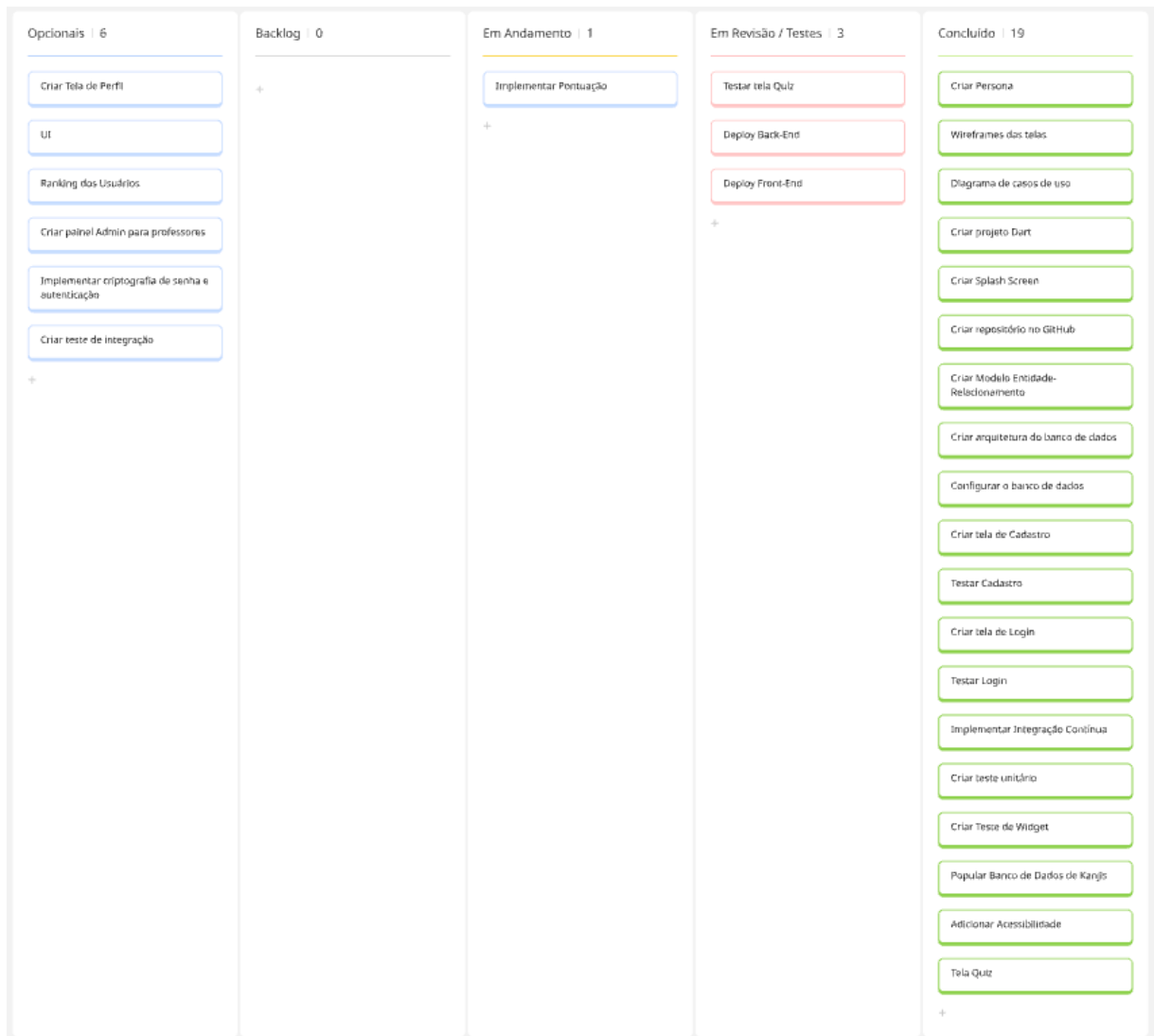
Fonte: SHINODA, Cintia Izumi. [Persona]. Disponível em:

[https://miro.com/app/board/uXjVIDg6SCs=?share\\_link\\_id=378253062752](https://miro.com/app/board/uXjVIDg6SCs=?share_link_id=378253062752). Acesso em: 22 mai. 2025.

Essas características influenciaram diretamente na criação de uma aplicação com interface visualmente simples, intuitiva e a inclusão de opção de acessibilidade (modo de alto contraste).

Como forma de otimizar o fluxo de trabalho, foi utilizado o Kanban (Figura 2), que indica, por meio de cartões, as tarefas a serem realizadas (*Backlog*), as que ainda estão em progresso (Em Andamento), as que precisam passar por revisão e/ou testes (Em Revisão/Testes) e, finalmente, as tarefas que já foram concluídas (Concluído).

Figura 2 - Kanban



Fonte: SHINODA, Cintia Izumi. [Kanban]. Disponível em:

[https://miro.com/app/board/uXjVIDCvAew=?share\\_link\\_id=279700456658](https://miro.com/app/board/uXjVIDCvAew=?share_link_id=279700456658). Acesso em: 16 mai. 2025.

Em um estágio mais avançado do desenvolvimento, criou-se a coluna "Opcionais", que inclui tarefas que, embora não sejam essenciais para a entrega do protótipo, agregam valor à experiência do usuário ou à escalabilidade futura do app.

## 2.5.4 ETAPA DE PROTOTIPAÇÃO

Com base nas funcionalidades priorizadas foram desenvolvidos protótipos das telas do aplicativo (Figuras 3 e 4).

Na prototipação objetivaram-se fluxo de navegação e clareza na interface.

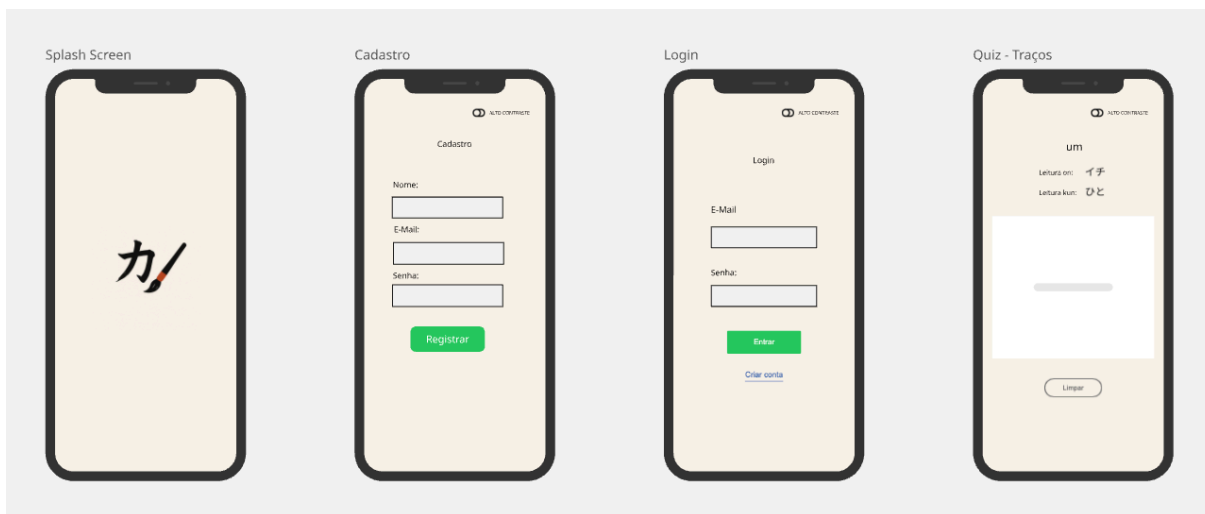
Figura 3 - Protótipo inicial das telas do aplicativo



Fonte: SHINODA, Cintia Izumi. [Protótipo inicial das telas do aplicativo]. Disponível em:

[https://miro.com/app/board/uXjVIDg8nKU=?share\\_link\\_id=238386759696](https://miro.com/app/board/uXjVIDg8nKU=?share_link_id=238386759696). Acesso em: 19 mar. 2025.

Figura 4 - Protótipo das telas do aplicativo em estágio mais desenvolvido



Fonte: SHINODA, Cintia Izumi. [Protótipo das telas]. Disponível em:

[https://miro.com/app/board/uXjVIDg8nKU=?share\\_link\\_id=238386759696](https://miro.com/app/board/uXjVIDg8nKU=?share_link_id=238386759696). Acesso em: 16 mai. 2025.

### 2.5.5 ETAPA DE TESTES

Exaustivos testes foram realizados internamente durante o desenvolvimento. Estes, simularam todas as funcionalidades desenvolvidas.

Além disso, o protótipo desenvolvido foi apresentado à parte da diretoria da Aliança Cultural Brasil-Japão em reunião ocorrida no dia 15 de maio de 2025.

Durante esse encontro, os membros da diretoria testaram a aplicação e forneceram um *feedback* positivo sobre a proposta e o potencial educativo do projeto.

Adicionalmente, foram levantadas sugestões pela Aliança. Dentre as quais, destacam-se:

- Inclusão de *tags* e categorização dos kanjis, facilitando a organização do conteúdo;
- Permitir que usuários criem roteiros de estudo personalizados;
- Explorar a integração com o material didático da Aliança, para ampliar a aplicabilidade do app em sala de aula.

Essas contribuições reforçam a relevância do projeto, demonstram engajamento real da Aliança e abrem caminho para futuras parcerias e melhorias na aplicação.

## 3 RESULTADOS: SOLUÇÃO FINAL

### 3.1 DEFINIÇÃO DA SOLUÇÃO

Como requisito adicional obrigatório do Projeto Integrador III, foi implementada uma API (*Application Programming Interface*), responsável por intermediar a comunicação entre o aplicativo *front-end*, desenvolvido em Flutter e o banco de dados hospedado em nuvem (MongoDB Atlas).

A API foi desenvolvida utilizando Node.js e Express, permitindo que as funcionalidades sejam registradas e acessadas de forma estruturada.

Essa implementação não apenas cumpre os critérios estabelecidos pela disciplina, como também viabiliza escalabilidade, modularidade e integração com outros sistemas no futuro, ampliando o potencial do app para além do ambiente acadêmico.

### 3.1.1 FERRAMENTAS

O projeto foi desenvolvido utilizando as ferramentas relacionadas a seguir:

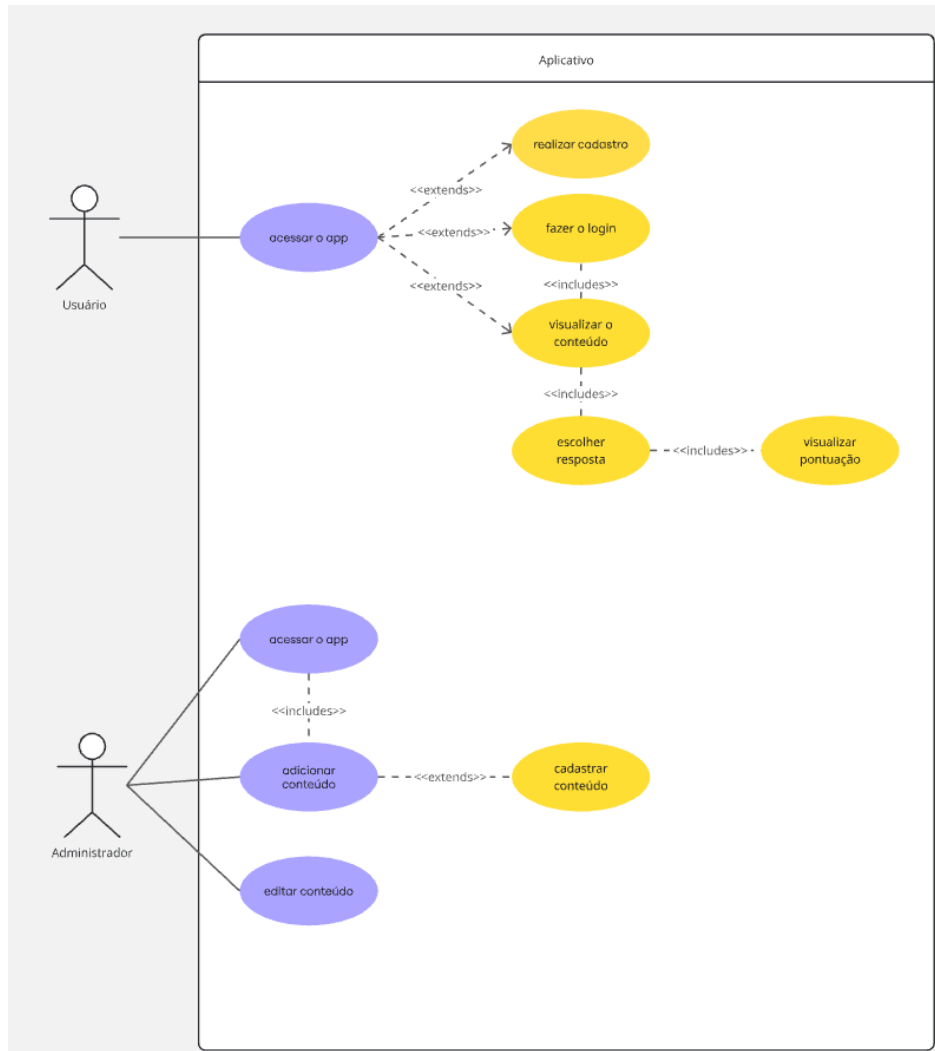
- Linguagens de programação:
  - *Front-End*: Dart (Framework: Flutter);
  - *Back-End*: JavaScript (Framework: Node.js (Express)).
- Diagramas e Modelos: Miro;
- IDE (*Integrated Development Environment*): Visual Studio Code (VS Code);
- Banco de Dados: MongoDB Atlas;
- Controle de Versões: Git e GitHub;
- Integração Contínua: GitHub Actions;
- Testes: Postman;
- Deploy:
  - *Back-End*: Render;
  - *Front-End*: Vercel.

### 3.1.2 FUNCIONALIDADES

As funcionalidades foram definidas com o apoio do diagrama UML de Casos de Uso (Figura 5), que demonstra as formas que o usuário pode interagir com o sistema.



Figura 5 - Diagrama UML de Casos de Uso



Fonte: SHINODA, Cintia Izumi. [Diagrama UML de Casos de Uso]. Disponível em: [https://miro.com/app/board/uXjVIDg6SAY=?share\\_link\\_id=280538043980](https://miro.com/app/board/uXjVIDg6SAY=?share_link_id=280538043980). Acesso em: 24 mar. 2025.

O aplicativo desenvolvido apresenta as seguintes funcionalidades:

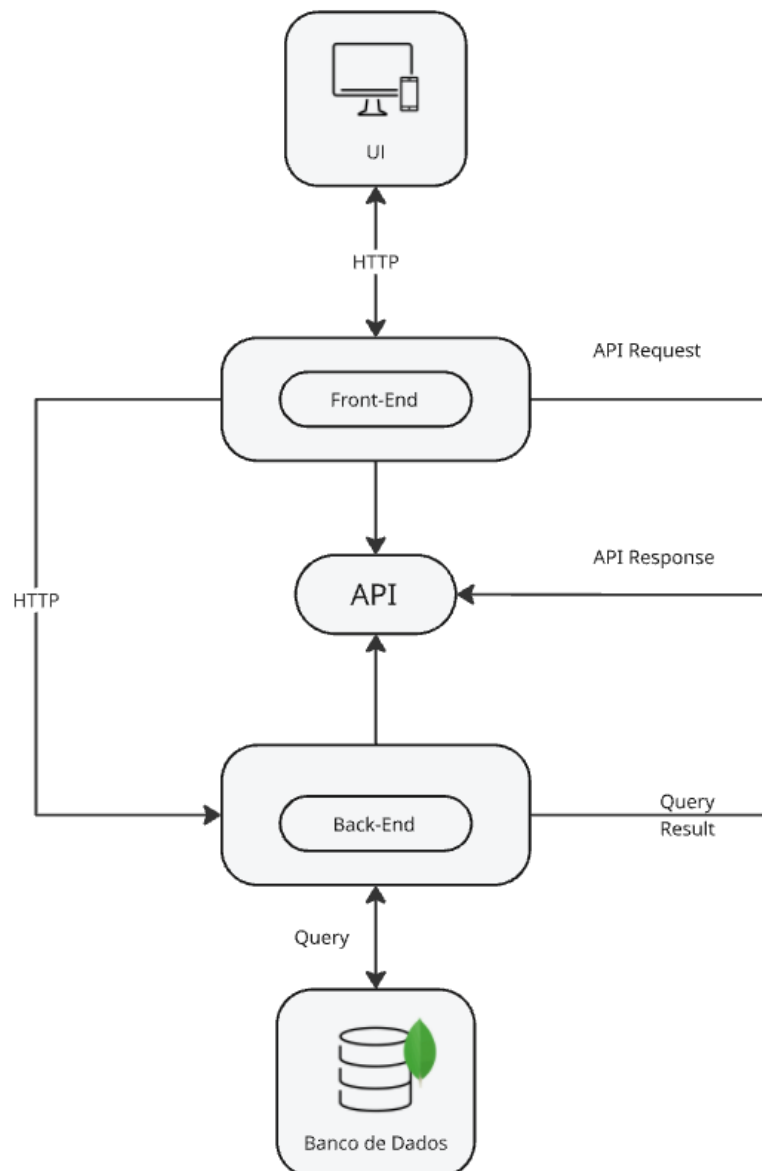
- Cadastro de usuário;
- Login de usuário
- Quiz;
- Interação através de traços na tela;
- Visualização do aplicativo em alto contraste.

### 3.2 ARQUITETURA E ESTRUTURA DO PROJETO

O *back-end* da aplicação expõe uma API REST/HTML, que é consumida pelo aplicativo Flutter (*Front-End*), disponibilizando, assim *endpoints*, como os de login e cadastro.

O *back-end* se comunica com o banco de dados usando uma API interna (por meio da biblioteca Mongoose).

Figura 6 - Arquitetura do Projeto



Fonte: SHINODA, Cintia Izumi. [Arquitetura do Projeto]. Disponível em: [https://miro.com/app/board/uXjVIAmW3ZQ=?share\\_link\\_id=645123918487](https://miro.com/app/board/uXjVIAmW3ZQ=?share_link_id=645123918487). Acesso em: 16 abr. 2025.

Figura 7 - Estrutura do Projeto

```
├── android/      # projeto Android
├── assets/       # arquivos estáticos (imagens, fontes, etc.)
│   └── logo.png
├── backend/      # API em Node.js
│   ├── controllers/
│   │   ├── kanjiController.js
│   │   ├── palavraController.js
│   │   ├── tentativaController.js
│   │   └── userController.js
│   ├── models/   # Modelos de dados
│   │   ├── Kanji.js
│   │   ├── Palavra.js
│   │   ├── Tentativa.js
│   │   └── User.js
│   ├── routes/   # Rotas da API
│   │   ├── kanjiRoutes.js
│   │   ├── palavraRoutes.js
│   │   ├── tentativaRoutes.js
│   │   └── userRoutes.js
│   ├── .env      # Arquivo de variáveis de ambiente
│   ├── package-lock.json
│   ├── package.json # Arquivo de dependências
│   └── server.js  # Arquivo principal da API
├── build/        # Arquivos de build do Flutter
├── ios/          # projeto iOS (Swift)
├── lib/          # Código fonte do aplicativo Flutter
│   ├── models/
│   │   ├── kanji.dart
│   │   ├── palavra.dart
│   │   ├── tentativa.dart
│   │   └── usuario.dart
│   ├── screens/  # Telas do aplicativo
│   │   ├── home_screen.dart
│   │   ├── login_screen.dart
│   │   ├── register_screen.dart
│   │   ├── splash_screen.dart
│   │   └── test_screen.dart
│   ├── services/ # Serviços de API
│   │   └── api_service.dart # Conexão com a API
│   ├── widgets/
│   │   ├── kanjiPainter.dart
│   │   └── kanjiTracingGame.dart
│   └── main.dart # Ponto de entrada do aplicativo
├── linux/        # projeto Linux
├── macos/        # projeto MacOS
├── test/         # Testes
├── web/          # projeto Web
├── windows/      # projeto Windows
├── .gitignore    # Pastas e arquivos a serem ignorados pelo Git
├── analysis_options.yaml # Configurações de análise do Dart
├── kanji_app.iml # Configuração do projeto
├── pubspec.lock  # Arquivo de bloqueio de dependências
├── pubspec.yaml  # Arquivo de dependências do Flutter
└── README.md     # Documentação do projeto
```

Fonte: SHINODA, Cintia Izumi. [Estrutura do projeto no GitHub]. Disponível em: <https://github.com/cintia-shinoda/projeto-integrador-3>. Acesso em: 7 abr. 2025.

### 3.2.1 BACK-END

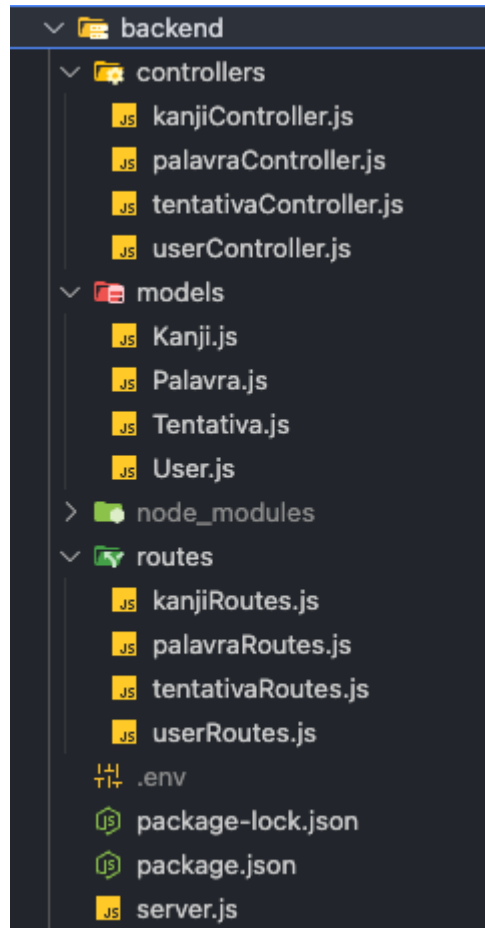
O *back-end*, que implementa a API que o *front-end* consome, está contido na pasta `backend/` (Figura 8). Esta parte foi desenvolvida em JavaScript (framework Node.js/Express).

Principais pastas no *back-end* são:

- `routes/`: Define os endpoints da API;
- `controllers/`: Contém a lógica de cada rota (exemplos: cadastro de usuário e sorteio de kanjis);
- `models/`: Define os *schemas* dos dados (Usuário, Kanji, Tentativa). Estes modelos são usados para interagir com a camada de persistência de dados (banco de dados MongoDB Atlas).

Ainda contido na pasta *back-end*, está o arquivo `server.js`, que é o ponto de partida da aplicação *back-end*, onde o servidor é configurado e iniciado.

Figura 8 - Pasta *back-end*



Fonte: SHINODA, Cintia Izumi. [Pasta *back-end*]. Captura em 17 mai. 2025

Os principais *endpoints* da API podem foram sumarizados na Tabela 1:

Tabela 1 - Principais *endpoints* da API

Método	Rota	Descrição
POST	/api/usuarios/cadastrar	Cadastro de novo usuário
POST	/api/usuarios/login	Autenticação e login
GET	/api/kanjis/aleatorio	Retorna um kanji aleatório
POST	/api/tentativas	Registra uma tentativa com traços
GET	/api/usuarios/:id/pontuacao	Retorna pontuação acumulada

Fonte: SHINODA, Cintia Izumi (2025)

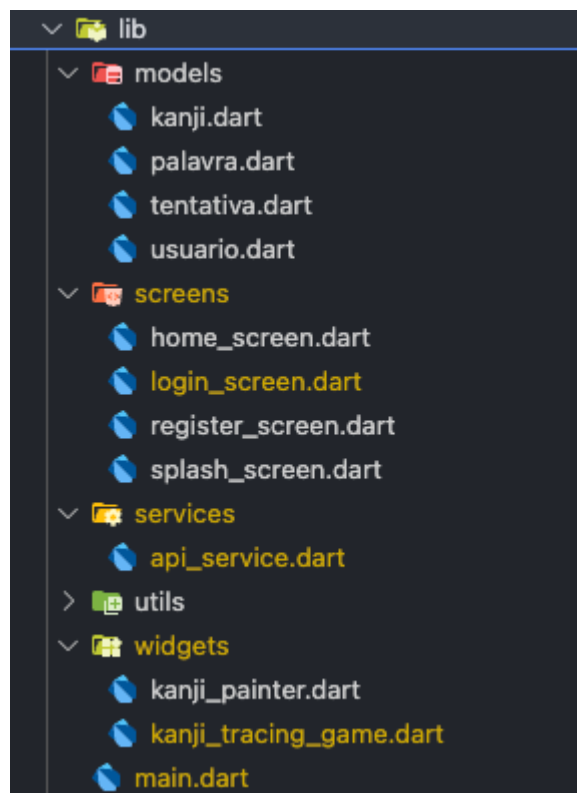
### 3.2.2 FRONT-END

O *front-end* da aplicação, desenvolvida em Dart (framework Flutter), se encontra na pasta `lib` (Figura 9).

Dentro de `lib/` as principais pastas são:

- `screens/`: armazena o código das telas que o usuário vê e interage (UI: User Interface);
- `widgets/`: contém componentes de UI reutilizáveis;
- `services/`: com o arquivo `api_service.dart`, é onde a lógica para se conectar com a API (consequentemente com o *back-end*), está;
- `models/`: define as estruturas de dados.

Figura 9 - Pasta `lib`



Fonte: SHINODA, Cintia Izumi. [Pasta `lib`]. Captura em 22 mai. 2025

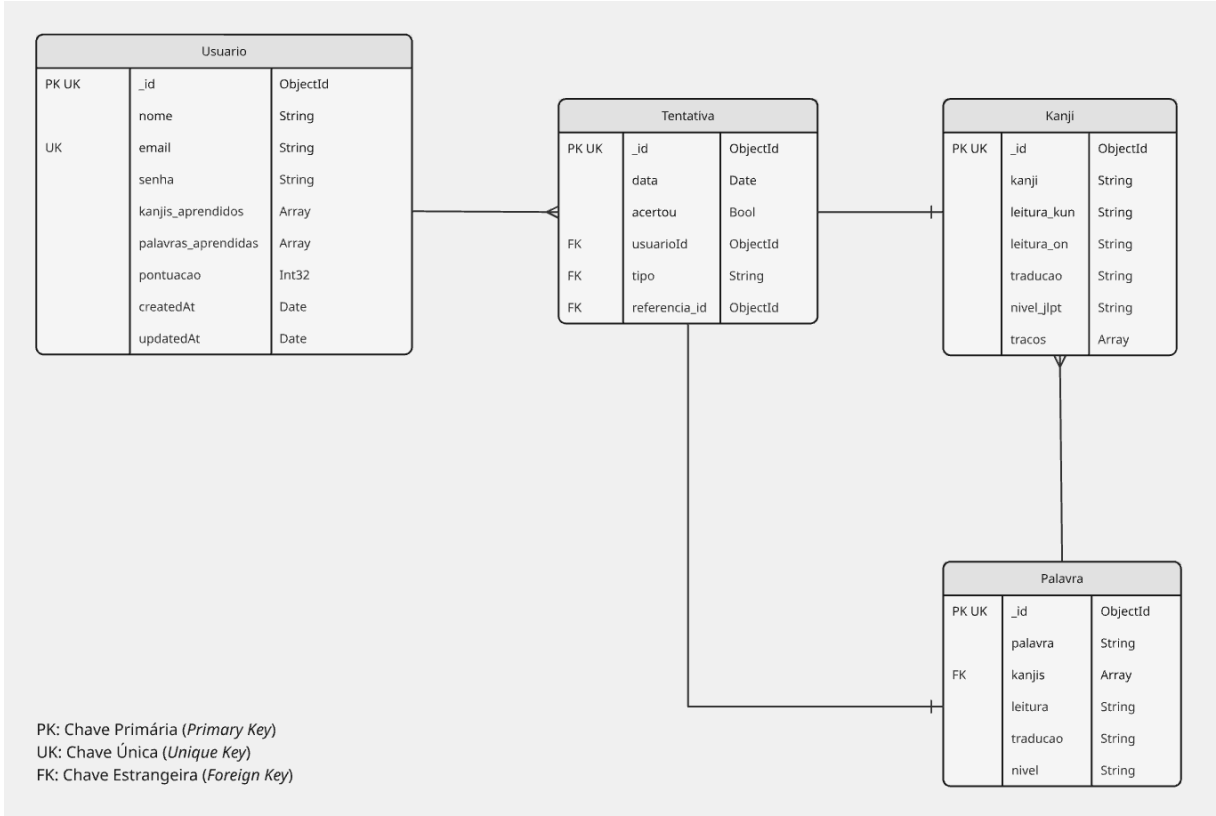
### 3.3 DADOS

#### 3.3.1 BANCO DE DADOS

MongoDB Atlas é o banco de dados NoSQL baseado em documentos, flexível (não tem esquemas rígidos), usado no projeto.

A Figura 10, apresenta o Modelo Entidade-Relacionamento (MER) do banco de dados construído.

Figura 10 - Modelo Entidade-Relacionamento



Fonte: SHINODA, Cintia Izumi. [Diagrama Modelo Entidade-Relacionamento (MER)]. Disponível em: [https://miro.com/app/board/uXjVID-Lill=?share\\_link\\_id=79473336483](https://miro.com/app/board/uXjVID-Lill=?share_link_id=79473336483). Acesso em: 10 mai. 2025.

Na Figura 11, está uma captura de tela do banco de dados com entradas de usuários de teste.

Figura 11 - Banco de Dados MongoDB Atlas

The screenshot displays the MongoDB Atlas web interface. At the top, the navigation bar shows the project 'kanji-app' and the cluster 'kanji-cluster'. The main header indicates the version is 8.0.9 and the region is AWS Sao Paulo (sa-east-1). Below the header, the left sidebar shows the database 'kanji\_app' with collections 'kanjis', 'palavras', 'tentativas', and 'usuarios'. The main content area is titled 'kanji\_app.usuarios' and shows storage and index statistics. A search bar is present with the query '{ field: 'value' }'. Below the search bar, two document snippets are displayed, each containing fields like '\_id', 'nome', 'email', 'senha', 'kanjis\_aprendidos', 'palavras\_aprendidas', 'pontuacao', 'createdAt', and 'updatedAt'.

Fonte: SHINODA, Cintia Izumi. [Banco de dados MongoDB Atlas]. Acesso restrito. Acesso em: 16 mai. 2025.

### 3.3.2 KANJIVG

Para que fosse possível obter a ordem correta dos traços dos kanjis, os dados foram extraídos diretamente do projeto KanjiVG, (Figura 12), um repositório colaborativo que fornece informações vetoriais completas dos kanjis em formato SVG/XML.

O repositório segue padrões de escrita japonesa e é amplamente utilizado em softwares educacionais.



Figura 12 - Repositório do KanjiVG

The screenshot shows the GitHub repository for KanjiVG. The repository is public and has 37 watchers, 192 forks, and 1.1k stars. The main content area displays a list of files and their commit history, including .github/workflows, kanji, .gitignore, COPYING, README.md, \_\_init\_\_.py, clean.py, kanjivg.py, kvg-index.json, kvg.py, kvg\_lookup.py, make-index.py, pyproject.toml, updatepublic.sh, utils.py, and xmlhandler.py. The right sidebar contains information about the repository, including 'About', 'Releases', 'Packages', 'Contributors', and 'Languages'.

File	Commit Message	Time Ago
.github/workflows	initial ci/cd	8 months ago
kanji	Fixups detected by read-write-test - see diffs	last week
.gitignore	fix imports	8 months ago
COPYING	Initial commit	16 years ago
README.md	Improve README by breaking into paragraphs	9 months ago
__init__.py	fix relative imports	8 months ago
clean.py	Stripped file related changes	9 months ago
kanjivg.py	Another instance of has_key remained	9 months ago
kvg-index.json	Change to use the Python for making the index	10 months ago
kvg.py	Fixups detected by read-write-test - see diffs	last week
kvg_lookup.py	fix relative imports	8 months ago
make-index.py	Add some comments in software and improve REA...	10 months ago
pyproject.toml	handle dated versioning	8 months ago
updatepublic.sh	Stripped file related changes	9 months ago
utils.py	Error found in 09af7.svg using kvg-lookup.py	10 months ago
xmlhandler.py	Add back python distro files; delete template.svg	3 years ago

**About**  
Kanji vector graphics  
[kanjivg.tagaini.net](https://kanjivg.tagaini.net)  
kanji vector-graphics  
Readme  
View license  
Activity  
Custom properties  
1.1k stars  
37 watching  
192 forks  
Report repository

**Releases** 12  
r20250422 (Latest)  
last month  
+ 11 releases

**Packages**  
No packages published

**Contributors** 20  
+ 6 contributors

**Languages**  
Python 98.5%  
Shell 1.5%

**README**  
KanjiVG: Kanji Vector Graphics  
**Licence**  
KanjiVG is copyright Ulrich Apel and released under the Creative Commons Attribution-Share Alike 3.0 licence:

Fonte: SHINODA, Cintia Izumi. [Repositório KanjiVG]. Disponível em: <https://github.com/KanjiVG/kanjivg>.

Acesso em: 17 mai. 2025.

### 3.3.3 ESTRUTURA DE DADOS

#### SVG

O formato SVG (*Scalable Vector Graphics*) é um formato de imagem vetorial que descreve gráficos utilizando XML. Assim, cada traço de um kanji é descrito por dados matemáticos.

Desta forma, os arquivos no formato SVG são primordiais para que seja possível que o usuário trace os traços na tela, e o *app* compare a entrada do usuário com o caminho definido no SVG.

No aplicativo, os SVGs foram convertidos e organizados em objetos JSON. Essas informações, então foram inseridas no banco de dados.

Alguns motivos pelos quais o uso de SVG se mostrou vantajoso:

- **Leveza:** são arquivos leves, assim carregam rapidamente;
- **Interatividade:** é possível manipular os SVGs para que os traços sejam exibidos um a um;
- **Escalabilidade:** as imagens vetoriais não perdem qualidade quando são redimensionadas, isso garante nitidez em quaisquer tamanhos de tela.

Figura 13 - Código SVG do kanji 「一」

```
04e00.svg
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3  Copyright (C) 2009/2010/2011 Ulrich Apel.
4  This work is distributed under the conditions of the Creative Commons
5  Attribution-Share Alike 3.0 Licence. This means you are free:
6  * to Share - to copy, distribute and transmit the work
7  * to Remix - to adapt the work
8
9  Under the following conditions:
10 * Attribution. You must attribute the work by stating your use of KanjiVG in
11   your own copyright header and linking to KanjiVG's website
12   (http://kanjivg.tagaini.net)
13 * Share Alike. If you alter, transform, or build upon this work, you may
14   distribute the resulting work only under the same or similar license to this
15   one.
16
17 See http://creativecommons.org/licenses/by-sa/3.0/ for more details.
18 --->
19 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd" [
20 <!--
21 xmlns:kvg CDATA #FIXED "http://kanjivg.tagaini.net"
22 kvg:element CDATA #IMPLIED
23 kvg:variant CDATA #IMPLIED
24 kvg:partial CDATA #IMPLIED
25 kvg:original CDATA #IMPLIED
26 kvg:part CDATA #IMPLIED
27 kvg:number CDATA #IMPLIED
28 kvg:tradForm CDATA #IMPLIED
29 kvg:radicalForm CDATA #IMPLIED
30 kvg:position CDATA #IMPLIED
31 kvg:radical CDATA #IMPLIED
32 kvg:phon CDATA #IMPLIED >
33 <!--
34 xmlns:kvg CDATA #FIXED "http://kanjivg.tagaini.net"
35 kvg:type CDATA #IMPLIED >
36 ]>
37 <svg xmlns="http://www.w3.org/2000/svg" width="109" height="109" viewBox="0 0 109 109">
38 <g id="kvg:StrokePaths_04e00" style="
39 fill:none;stroke:#000000;stroke-width:3;stroke-linecap:round;stroke-linejoin:round;">
40 <g id="kvg:04e00" kvg:element="一" kvg:radical="general">
41 <path id="kvg:04e00-s1" kvg:type="一" d="
42 M11,54.25c3.19,0.62,6.25,0.75,9.73,0.5c20.64-1.5,50.39-5.12,68.58-5.24c3.6-0.02,5.77,0.24,7.57,0.49"/>
43 </g>
44 </g>
45 <g id="kvg:StrokeNumbers_04e00" style="font-size:8;fill:#808080">
46 <text transform="matrix(1 0 0 1 4.25 54.13)">1</text>
47 </g>
48 </svg>
```

Fonte: SHINODA, Cintia Izumi. [Código SVG do kanji 「一」]. Captura em: 22 mai. 2025.

## JSON

As informações de cada kanji foram convertidas do SVG para o formato JSON.

Cada objeto JSON contém:

- Informações gerais do kanji: tradução e leituras *kun* e *on*;
- *Array* contendo:
  - número do traço;
  - caminho vetorial (*path*);
  - pontos de início e fim.

O JSON, então é armazenado no banco de dados MongoDB Atlas (Figura 14).

Figura 13 - JSON do kanji 「一」

```
_id: ObjectId('68200d8eecfc48f714aaa62a')
kanji : "一"
leitura_kun : "ひと"
leitura_on : "イチ"
traducao : "um"
nivel_jlpt : "N5"
▼ tracos : Array (1)
  ▼ 0: Object
    ordem : 1
    svg : "M20,50 L280,50"
    ▼ ponto_inicio : Object
      x : 20
      y : 50
    ▼ ponto_fim : Object
      x : 280
      y : 50
```

Fonte: SHINODA, Cintia Izumi. [JSON do kanji 「一」 ]. Acesso restrito. Captura em: 22 mai. 2025.

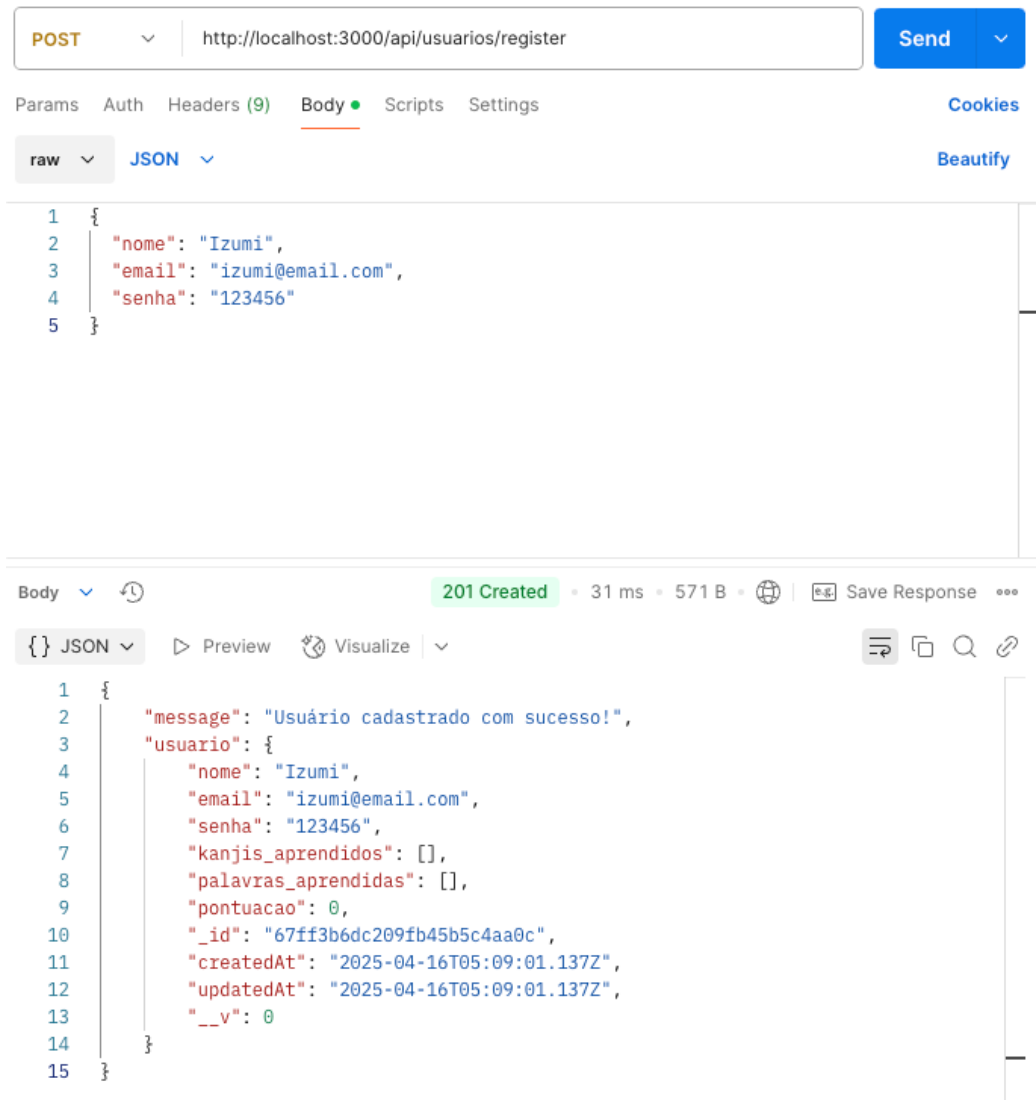
## 3.4 TESTES

### 3.4.1 TESTES DE API

Utilizou-se o Postman para testar as rotas RESTful do *back-end* e verificou-se os comportamentos esperados nas operações de cadastro de usuário (Figura 15), listagem dos

usuários cadastrados (Figura 16), login de usuário (Figura 17), cadastro de kanji (Figura 18), listagem dos kanjis cadastrados (Figura 19).

Figura 14 - Resultado do teste de Cadastro de Usuário no Postman



Fonte: SHINODA, Cintia Izumi. [Resultado do teste de Cadastro de Usuário no Postman]. Acesso Restrito. Captura em: 16 abr. 2025.

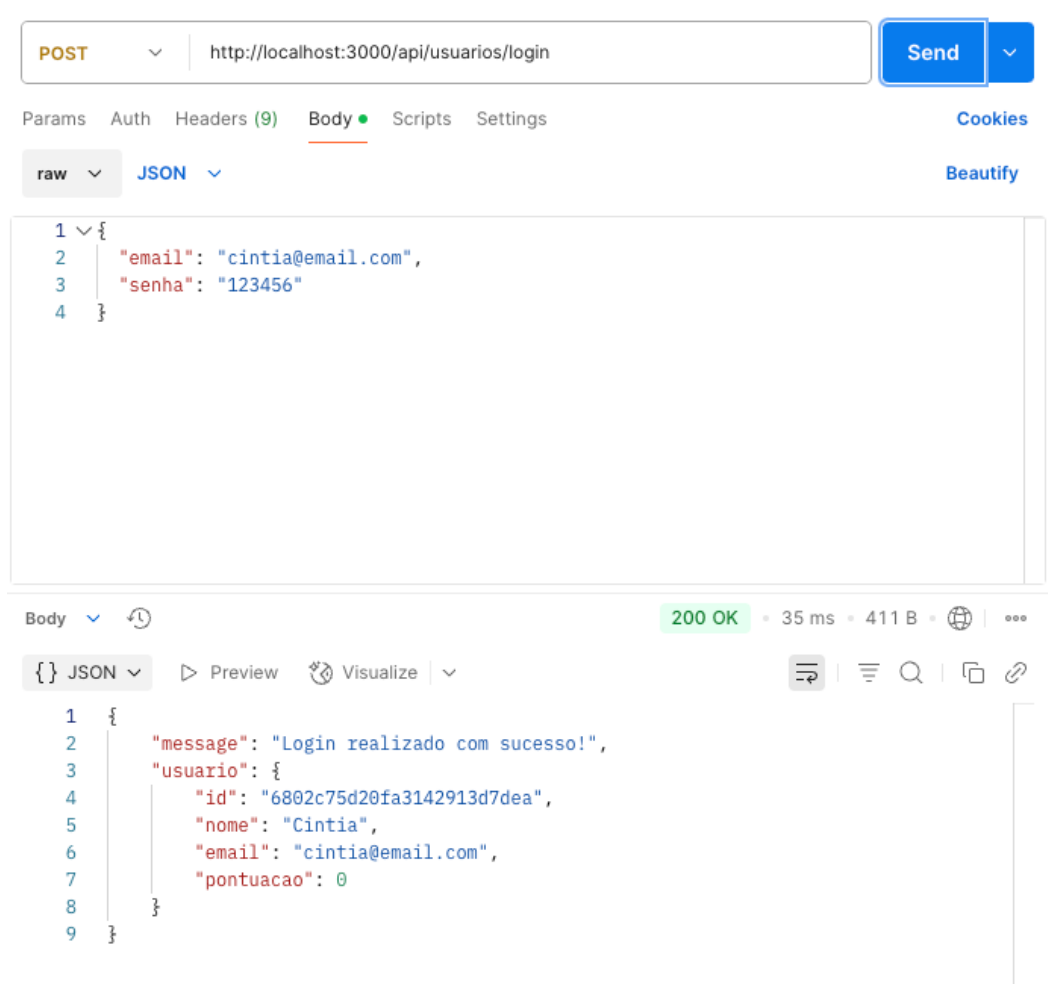
Figura 15 - Resultado do teste de listagem dos usuários cadastrados no Postman

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/api/usuarios`. The response is a 200 OK status with a response time of 19 ms and a body size of 1.21 KB. The response body is a JSON array containing three user objects. The first user is 'Cintia', the second is 'Aluno-1', and the third is 'Aluno-2'. Each user object includes fields for \_id, nome, email, senha, kanjis\_aprendidos, palavras\_aprendidas, pontuacao, createdAt, updatedAt, and \_\_v.

```
1  [
2    {
3      "_id": "67fdb59af0b3f301329c57f8",
4      "nome": "Cintia",
5      "email": "cintia@email.com",
6      "senha": "123456",
7      "kanjis_aprendidos": [],
8      "palavras_aprendidas": [],
9      "pontuacao": 25,
10     "createdAt": "2025-04-15T01:25:46.364Z",
11     "updatedAt": "2025-04-15T01:59:42.702Z",
12     "__v": 0
13   },
14   {
15     "_id": "67fec2c903c94c74595843a8",
16     "nome": "Aluno-1",
17     "email": "aluno@email.com",
18     "senha": "123456",
19     "kanjis_aprendidos": [],
20     "palavras_aprendidas": [],
21     "pontuacao": 0,
22     "createdAt": "2025-04-15T20:34:17.011Z",
23     "updatedAt": "2025-04-15T20:34:17.011Z",
24     "__v": 0
25   },
26   {
27     "_id": "67fec33f03c94c74595843ae",
28     "nome": "Aluno-2",
29     "email": "aluno2@email.com",
30     "senha": "123456",
31     "kanjis_aprendidos": [],
32     "palavras_aprendidas": [],
33     "pontuacao": 0,
34     "createdAt": "2025-04-15T20:36:15.566Z",
35     "updatedAt": "2025-04-15T20:36:15.566Z",
36     "__v": 0
37   },
38   ]
```

Fonte: SHINODA, Cintia Izumi. [Resultado do teste de listagem dos usuários cadastrados no Postman]. Captura em: 16 abr. 2025.

Figura 16 - Resultado do teste de login no Postman



Fonte: SHINODA, Cintia Izumi. [Resultado do teste de login no Postman]. Captura em: 18 abr. 2025.

Figura 17 - Resultado do teste de cadastro de kanji no Postman

POST

http://localhost:3000/api/kanjis

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   "leitura": "さん",
3   "traducao": "montanha",
4   "leitura_kun": "やま",
5   "leitura_on": "サン",
6   "tracos": [
7     {
8       "ordem": 1,
9       "svg": "M50 10 V90",
10      "ponto_inicio": { "x": 50, "y": 10 },
11      "ponto_fim": { "x": 50, "y": 90 }
12    },
13    {
14      "ordem": 2,
15      "svg": "M30 30 V70",
16      "ponto_inicio": { "x": 30, "y": 30 },
17      "ponto_fim": { "x": 30, "y": 70 }
18    },
19    {
20      "ordem": 3,
21      "svg": "M70 30 V70",
22      "ponto_inicio": { "x": 70, "y": 30 },
23      "ponto_fim": { "x": 70, "y": 70 }
24    }
25  ]
26 }
```

Body

201 Created

31 ms

683 B

Save Response

JSON

Preview

Visualization

leitura	さん
traducao	montanha
leitura_kun	やま
leitura_on	サン
> tracos [3]	
_id	67ff3ad7c209fb45b5c4aa08
_v	0

Fonte: SHINODA, Cintia Izumi. [Resultado do teste de cadastro do kanji no Postman]. Captura em: 16 abr. 2025.

GET
http://localhost:3000/api/kanjis
Send

Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies

Key	Value	Description	Bulk Edit	Presets
<input checked="" type="checkbox"/> Content-Type	application/json			
Key	Value	Description		

Body
Cookies
Headers (8)
Test Results
200 OK
30 ms
2.92 KB
Save Response

JSON
Preview
Visualize

_id	leitura	traducao	leitura_kun	leitura_on	tracos	_v			
0	にち	dia	ひ	ニチ	ponto_inicio	ponto_fim	ordem	svg	0
					x 10	x 90	1	M10 10 H90	
					y 10	y 10			
					x 90	x 90	2	M90 10 V90	
					y 10	y 90			
					x 90	x 10	3	M90 90 H10	
					y 90	y 90			
					x 10	x 10	4	M10 90 V10	
					y 90	y 10			
					1	げつ	lua	つき	
x 10	x 10	1	M10 10 V90						
y 10	y 90								
x 10	x 70	2	M10 10 H70						
y 10	y 10								
x 10	x 60	3	M10 50 H60						
y 50	y 50								
x 10	x 70	4	M10 90 H70						
y 90	y 90								
67fdacb08ad10eeb7ed9e44d	さん	montanha	やま	サン					ponto_inicio

### 3.4.2 TESTES UNITÁRIOS

A Figura 20, mostra o resultado da execução de um teste unitário de validação de *login*. Valida se o e-mail e a senha não estão vazios antes de chamar a API de *login*.



Figura 19 - Resultado do teste unitário de validação de login

```
• (base) cintiashinoda@192 pi-3-repo % flutter test test/utils/validacao_test.dart
Resolving dependencies...
Downloading packages...
  async 2.12.0 (2.13.0 available)
  fake_async 1.3.2 (1.3.3 available)
  leak_tracker 10.0.8 (10.0.9 available)
  lints 5.1.1 (6.0.0 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  vm_service 14.3.1 (15.0.0 available)
Got dependencies!
6 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.
00:04 +4: All tests passed!
```

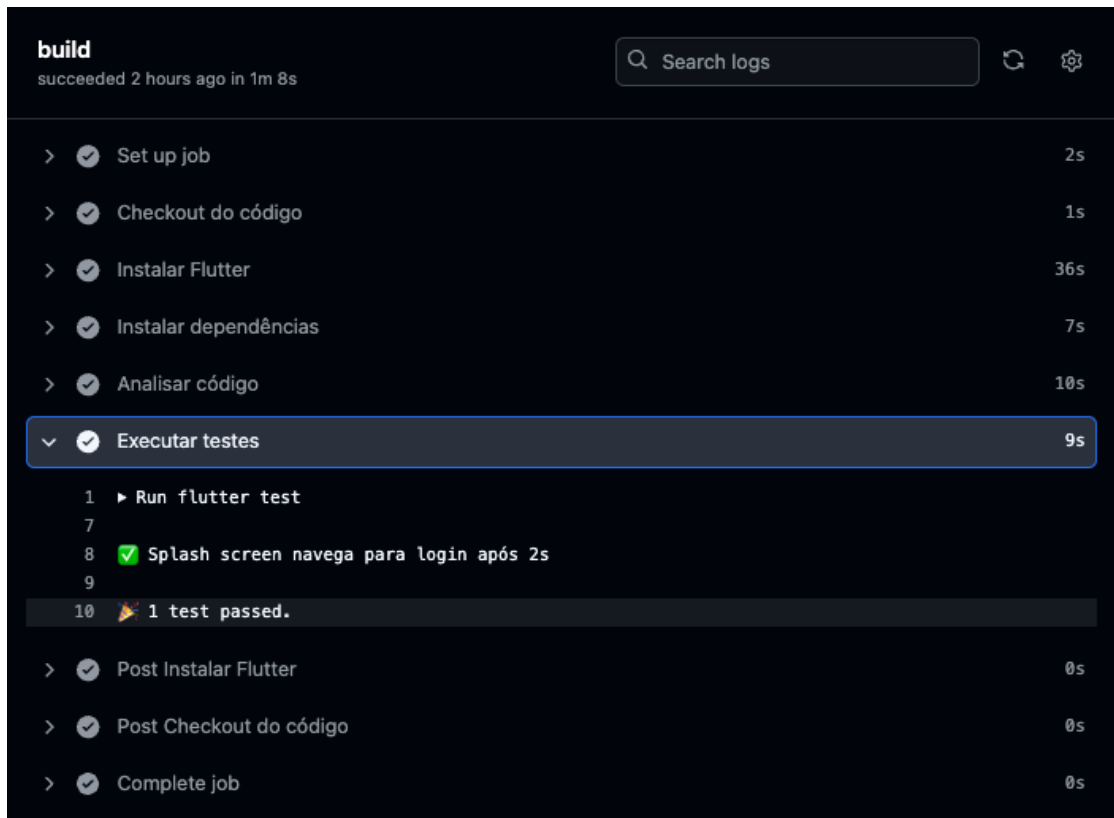
Fonte: SHINODA, Cintia Izumi. [Resultado do teste unitário de validação de login]. Captura em: 15 abr. 2025.

### 3.4.3 TESTES DE *WIDGETS*

Testes de widgets são exclusivos do Flutter e testam interações visuais, navegação e comportamentos.

O teste de *widget* implementado (Figura 21), assegura que após dois segundos do momento em que a *Splash Screen* é exibida, a tela de login será exibida.

Figura 20 - Resultado do teste de widget implementado



Fonte: SHINODA, Cintia Izumi. [Resultado do teste de widget implementado]. Captura em: 15 abr. 2025.

### 3.5 INTEGRAÇÃO CONTÍNUA

A integração contínua foi implementada com GitHub Actions para Flutter.

O *workflow* de integração contínua: "Flutter CI" (Figura 22) pode ser consultada em: [https://github.com/cintia-shinoda/projeto-integrador-3/actions/workflows/flutter\\_ci.yml](https://github.com/cintia-shinoda/projeto-integrador-3/actions/workflows/flutter_ci.yml)

O *workflow* "Flutter CI", tem como objetivo, fazer com que, sempre que se faça um *push* no GitHub, os testes do aplicativo sejam executados automaticamente.

Figura 21 - *Workflow* "Flutter CI" no GitHub Actions

The screenshot displays the GitHub Actions interface for the repository 'projeto-integrador-3' owned by 'cintia-shinoda'. The 'Actions' tab is selected, showing a list of workflow runs for the 'Flutter CI' workflow (flutter\_ci.yml). The left sidebar contains navigation links for 'All workflows', 'Flutter CI', and various management tools like 'Caches', 'Attestations', 'Runners', 'Usage metrics', and 'Performance metrics'. The main area shows a table of 6 workflow runs, each with a status icon, a title, a description, a commit hash, the branch name 'main', and the time since the run completed.

Event	Status	Branch	Actor
update: widget_test.dart timer	Success	main	cintia-shinoda
add: widget_test.dart	Failure	main	cintia-shinoda
remove: test/widget_test.dart	Failure	main	cintia-shinoda
update: flexibilidade em Analisar código	Failure	main	cintia-shinoda
update: versão do Flutter no workflow CI	Failure	main	cintia-shinoda
Integração Contínua com GitHub	Failure	main	cintia-shinoda

Fonte: SHINODA, Cintia Izumi. [Workflow "Flutter CI" no GitHub Actions]. Disponível em: [https://github.com/cintia-shinoda/projeto-integrador-3/actions/workflows/flutter\\_ci.yml](https://github.com/cintia-shinoda/projeto-integrador-3/actions/workflows/flutter_ci.yml). Acesso em: 7 mai. 2025.

### 3.6 VERSIONAMENTO DE CÓDIGO

A seguir, na Figura 23, está uma captura de tela do repositório utilizado para versionamento e compartilhamento do código-fonte do projeto hospedado no GitHub.

Este, encontra-se disponível em <https://github.com/cintia-shinoda/projeto-integrador-3>.

Figura 22 - Captura de tela do repositório do projeto no GitHub

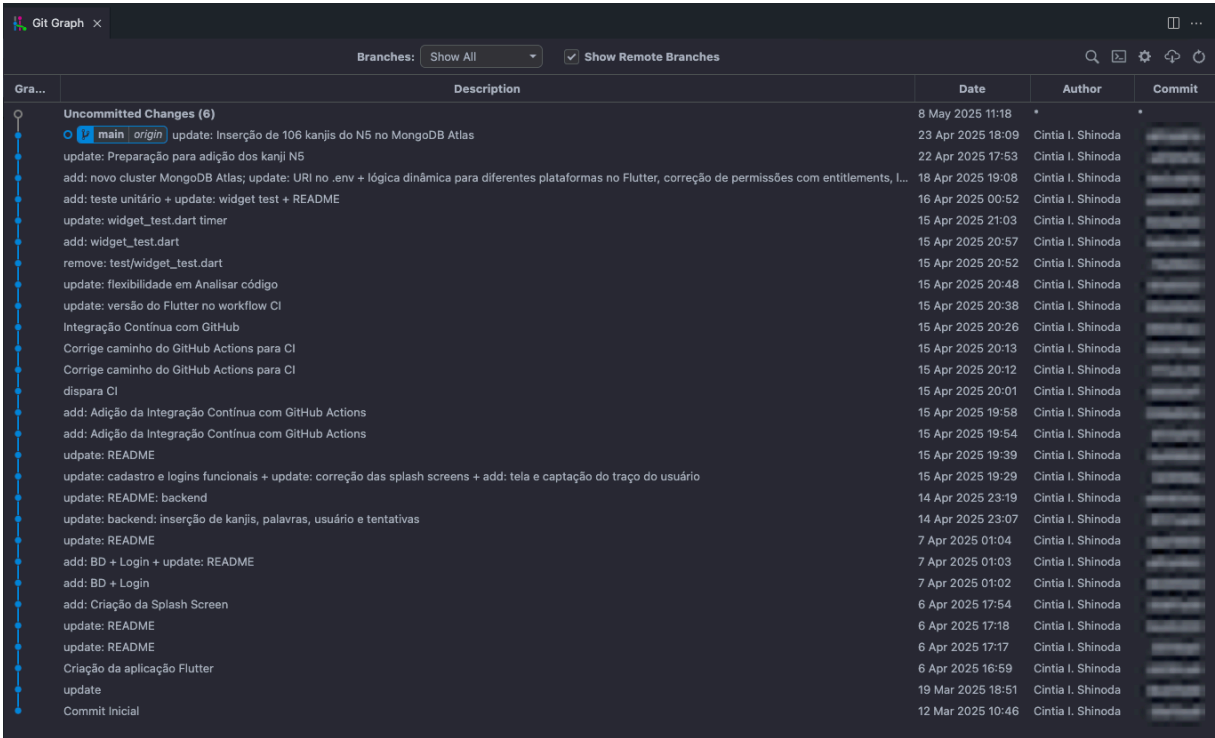
The screenshot displays the GitHub interface for the repository 'projeto-integrador-3' owned by 'cintia-shinoda'. The repository is public and has 24 commits. The main content area shows a list of files and folders, including .github/workflows, android, assets, backend, ios, lib, linux, macos, test, web, windows, .DS\_Store, .gitignore, .metadata, README.md, analysis\_options.yaml, pubspec.lock, and pubspec.yaml. The commit history shows updates to widget\_test.dart timer and integration with GitHub. The sidebar on the right provides information about the repository, including its description 'Projeto Integrador em Computação III - 1/2025', tags for dart, mobile-app, kanji, node-js, flutter, express-js, and mongodb-atlas, and statistics for stars, forks, and releases. The 'Languages' section shows a bar chart of the code's language distribution.

Language	Percentage
Dart	30.2%
C++	26.7%
CMake	21.8%
JavaScript	11.1%
HTML	3.4%
Ruby	3.1%
Other	3.7%

Fonte: SHINODA, Cintia Izumi. [Repositório GitHub do projeto]. Disponível em: <https://github.com/cintia-shinoda/projeto-integrador-3>. Acesso em: 7 abr. 2025.

Na Figura 24, uma captura de tela com a listagem dos *commits* realizados, usando a extensão Git Graph da IDE VS Code.

Figura 23 - Lista de *commits* realizados e listados pelo Git Graph

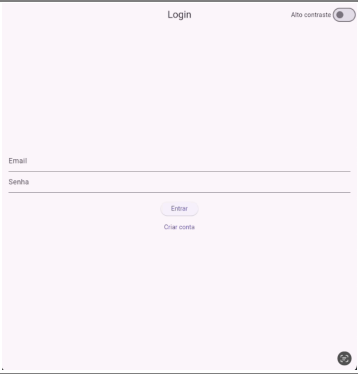
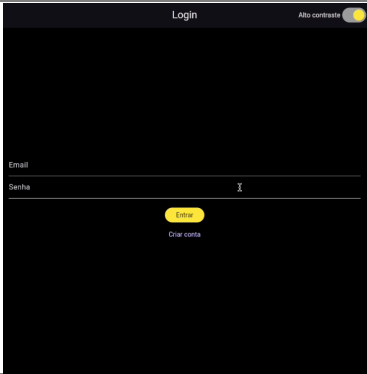


Fonte: SHINODA, Cintia Izumi. [Lista de commits realizados e listados pelo Git Graph]. Captura em: 8 mai. 2025.

3.7 ACESSIBILIDADE

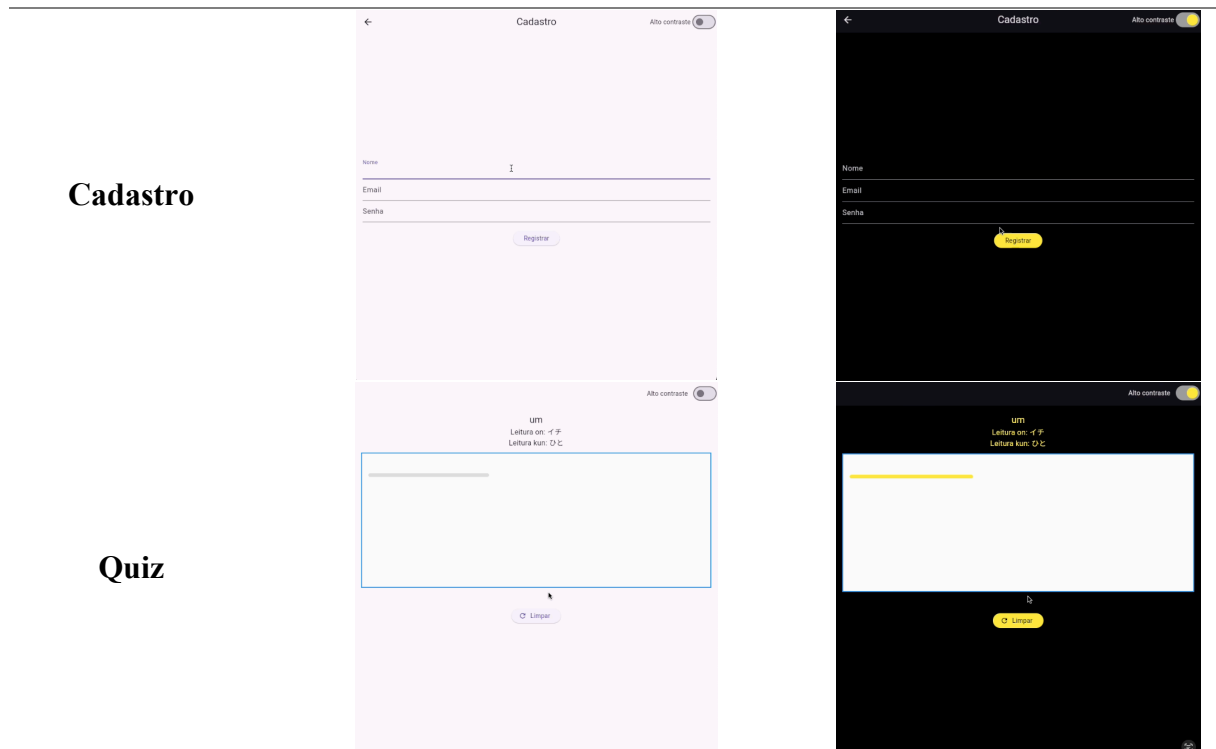
Foi implementado um modo de alto contraste na aplicação, promovendo desta forma, a inclusão de pessoas com deficiências visuais, especialmente com daltonismo ou baixa visão. A funcionalidade permite alternar entre o tema padrão e o tema com alto contraste de cores, facilitando a distinção visual dos elementos na tela.

Tabela 2 - Tabela comparativa entre telas com o tema padrão e telas com o tema com a funcionalidade de alto contraste ativada.

Tela	Modo Normal	Modo Alto Contraste
Login		

## Cadastro

## Quiz



Fonte: SHINODA, Cintia Izumi. [Tabela comparativa de telas em Modo Normal e Modo de Alto Contraste].  
Captura em 10 mai. 2025.

Com isso, as funcionalidades inclusas foram:

- Alteração de cores da interface para tons mais contrastantes;
- Manutenção da estética geral e responsividade do aplicativo;
- Aplicação consistente do tema em todas nas telas;
- Alternância acessível por *switch* com texto explicativo na parte superior da tela.

As modificações para a implementação foram:

- Gerenciamento de estado global com `StatefulWidget`
- a variável `modoAltoContraste` foi declarada no `MyApp` (widget raiz da aplicação)
- um método `toggleContraste(bool novoValor, atualiza o estado visual do app ao ser acionado.`
- Tema dinâmico com `ThemeData`: foram definidos dois temas:
- `temaNormal`
- `temaAltoContraste`: cores com contraste elevado (fundo escuro com texto branco e realces em amarelo vibrante para elementos gráficos)
- Uso de `onGenerateRoute` para propagação do modo entre telas

- Adaptação visual dos widgets customizados
- Componente de alternância (*Switch*).

### 3.8 DEPLOY

#### 3.8.1 BACK-END

Para disponibilizar a API REST de forma pública, optou-se pelo uso da plataforma Render, uma solução gratuita para *deploys* automatizados de serviços web.

Foi criada uma instância de Web Service no Render, com integração direta ao repositório "projeto-integrador-3" do GitHub (que contém o código *back-end* da aplicação).

Após o *build* automatizado realizado pela plataforma, o serviço foi disponibilizado com sucesso no endereço: <https://kanji-backend-kbhr.onrender.com>.

#### **Observação sobre o tempo de resposta inicial do servidor:**

Ao acessar o aplicativo pela primeira vez, pode haver um atraso de alguns segundos no carregamento. Isso ocorre porque o servidor, por estar hospedado em um serviço gratuito (Render), é automaticamente colocado em modo de suspensão após 15 minutos de inatividade. Quando um novo acesso é realizado, o serviço precisa ser reinicializado, causando um pequeno tempo de espera antes da resposta inicial. Após esse processo, o desempenho do aplicativo volta ao normal.

Essa é uma característica comum de hospedagens gratuitas com modo de economia de recursos, e não representa um problema de funcionamento do sistema.

Figura 24 - Dashboard no Render com os *logs* do *deploy*

My Workspace

kanji-backend

WEB SERVICE

kanji-backend

Node Free Upgrade your instance →

cintia-shinoda / projeto-integrador-3 ↗ main

<https://kanji-backend-kbhr.onrender.com>

ⓘ Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

[Upgrade now](#)

All logs Search Live tail GMT-3 ↕

```
May 11 01:42:32 AM x5m7c at wrapModuleLoad (node:internal/modules/cjs/loader:220:24)
May 11 01:42:32 AM x5m7c at Function.executeUserEntryPoint [as runMain] (node:internal/module
s/run_main:170:5)
May 11 01:42:32 AM x5m7c at node:internal/main/run_main_module:36:49 {
May 11 01:42:32 AM x5m7c code: 'MODULE_NOT_FOUND',
May 11 01:42:32 AM x5m7c requireStack: []
May 11 01:42:32 AM x5m7c }
May 11 01:42:32 AM x5m7c Node.js v22.14.0
May 11 01:44:08 AM x5m7c ==> Deploying...
May 11 01:44:22 AM hr2jk ==> Running 'node server.js'
May 11 01:44:25 AM hr2jk (node:97) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated opti
on: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next ma
jor version
May 11 01:44:25 AM hr2jk (Use 'node --trace-warnings ...' to show where the warning was created)
May 11 01:44:25 AM hr2jk (node:97) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated o
ption: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the n
ext major version
May 11 01:44:25 AM hr2jk 🚀 Servidor acessível em todas as interfaces na porta 10000
May 11 01:44:27 AM hr2jk 🟢 Conectado ao MongoDB
May 11 01:44:29 AM hr2jk ==> Your service is live 🎉
```

Need better ways to work with logs? Try the [Render CLI](#) or set up a [log stream integration](#) →

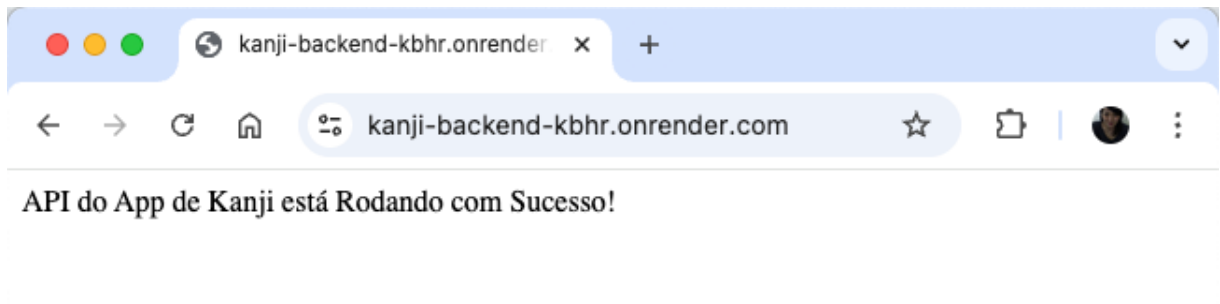
Fonte: SHINODA, Cintia Izumi. [Dashboard do Render com logs do *deploy* do *back-end*] Acesso restrito.

Captura em: 11 mai. 2025.

Para facilitar a visualização e garantir que o serviço está ativo, foi criada uma rota de verificação (GET /) que retorna uma mensagem simples de *status* da API (Figura 26).



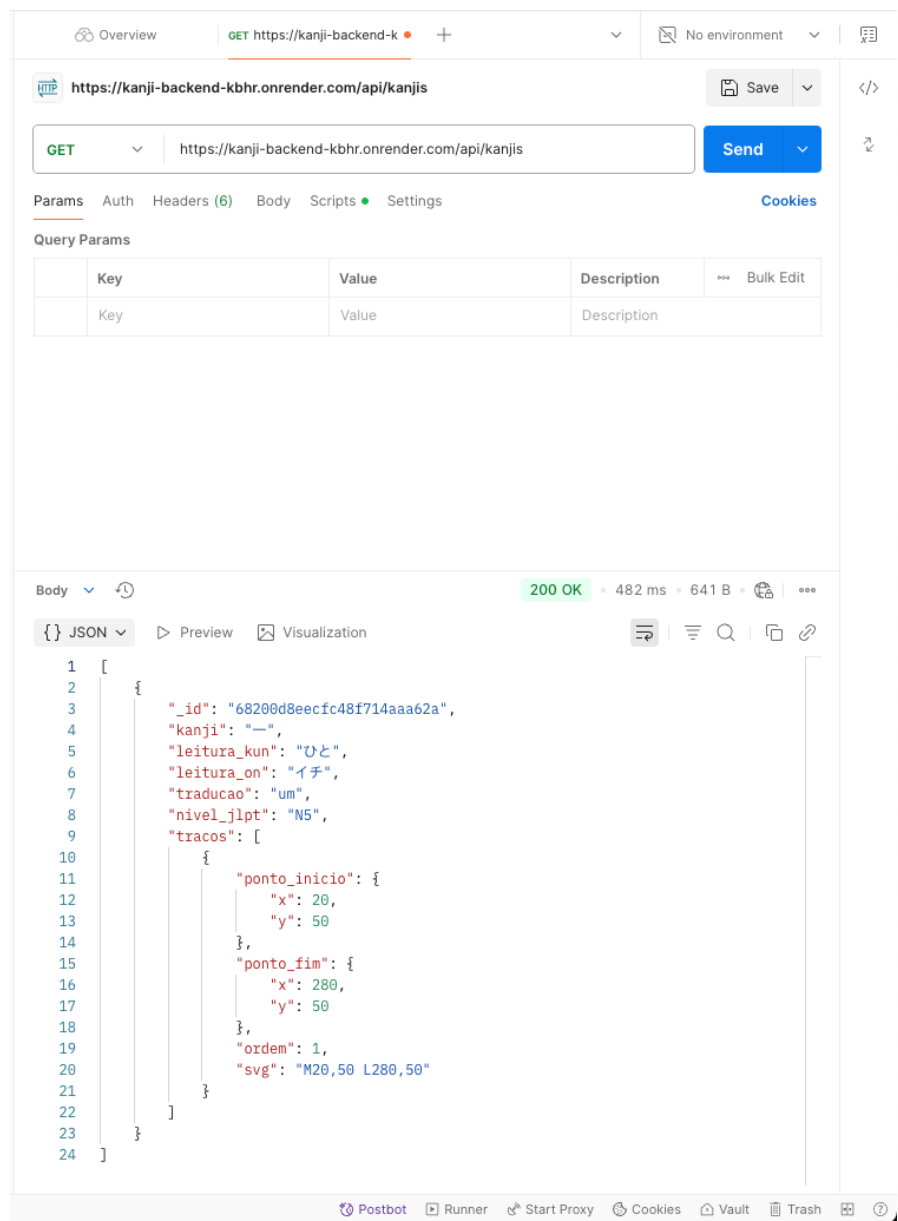
Figura 25 - Mensagem no navegador que indica que a API está ativa



Fonte: SHINODA, Cintia Izumi. [Navegador ao acessar a URL]. Disponível em: <https://kanji-backend-kbhr.onrender.com>. Acesso em: 11 mai. 2025.

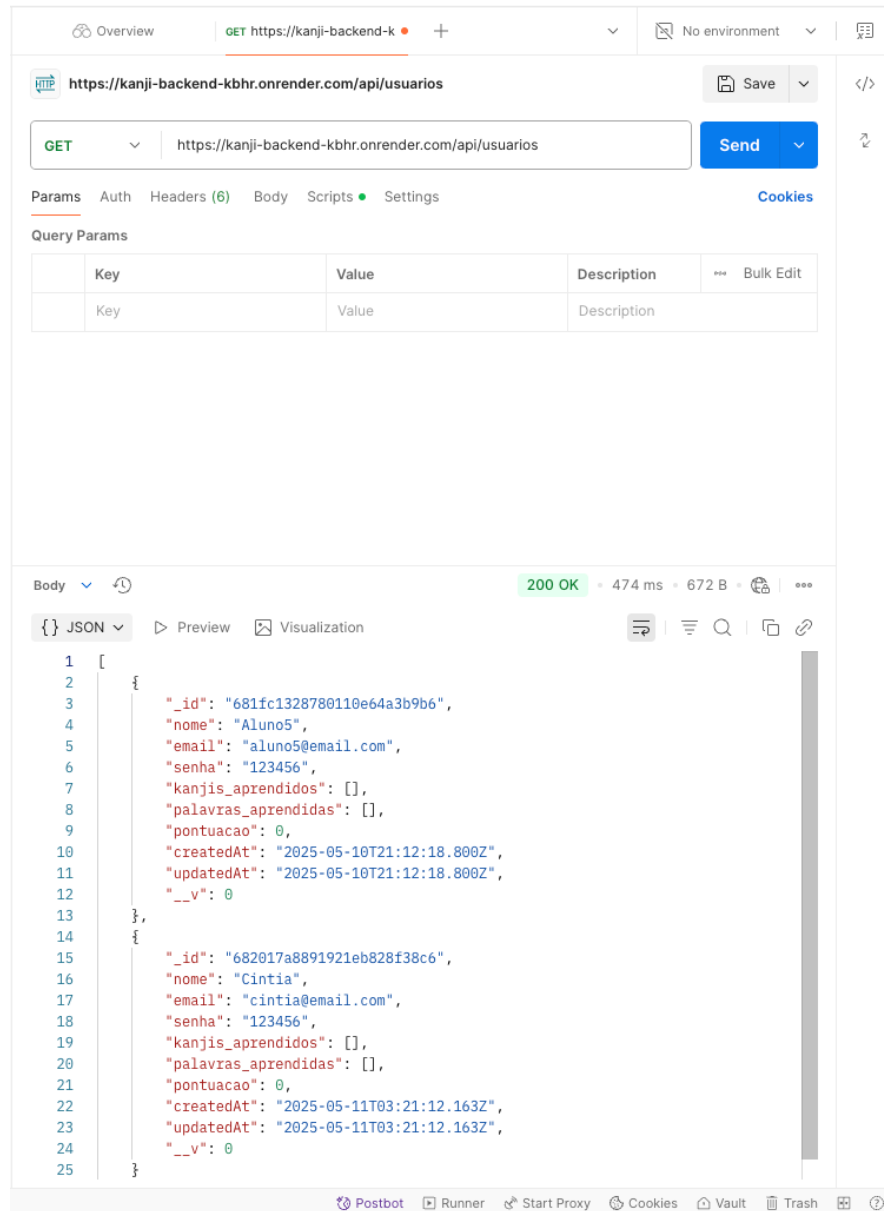
Para testarmos os *endpoints*, realizou-se testes de API no Postman, com a URL criada no Render (Figuras 27 e 28).

Figura 26 - Resultado do teste realizado no Postman da listagem do kanji cadastrado



Fonte: SHINODA, Cintia Izumi. [Resultado do teste realizado no Postman da listagem de kanji cadastrado].  
Captura em: 11 mai. 2025.

Figura 27 - Resultado do teste realizado no Postman da listagem dos usuários cadastrados



Fonte: SHINODA, Cintia Izumi. [Resultado do teste realizado no Postman da listagem dos usuários cadastrados]. Captura em: 11 mai. 2025.

### 3.8.2 FRONT-END

Para tornar a aplicação acessível publicamente, foi utilizada a plataforma Vercel, que permite a hospedagem gratuita de aplicações web estáticas.

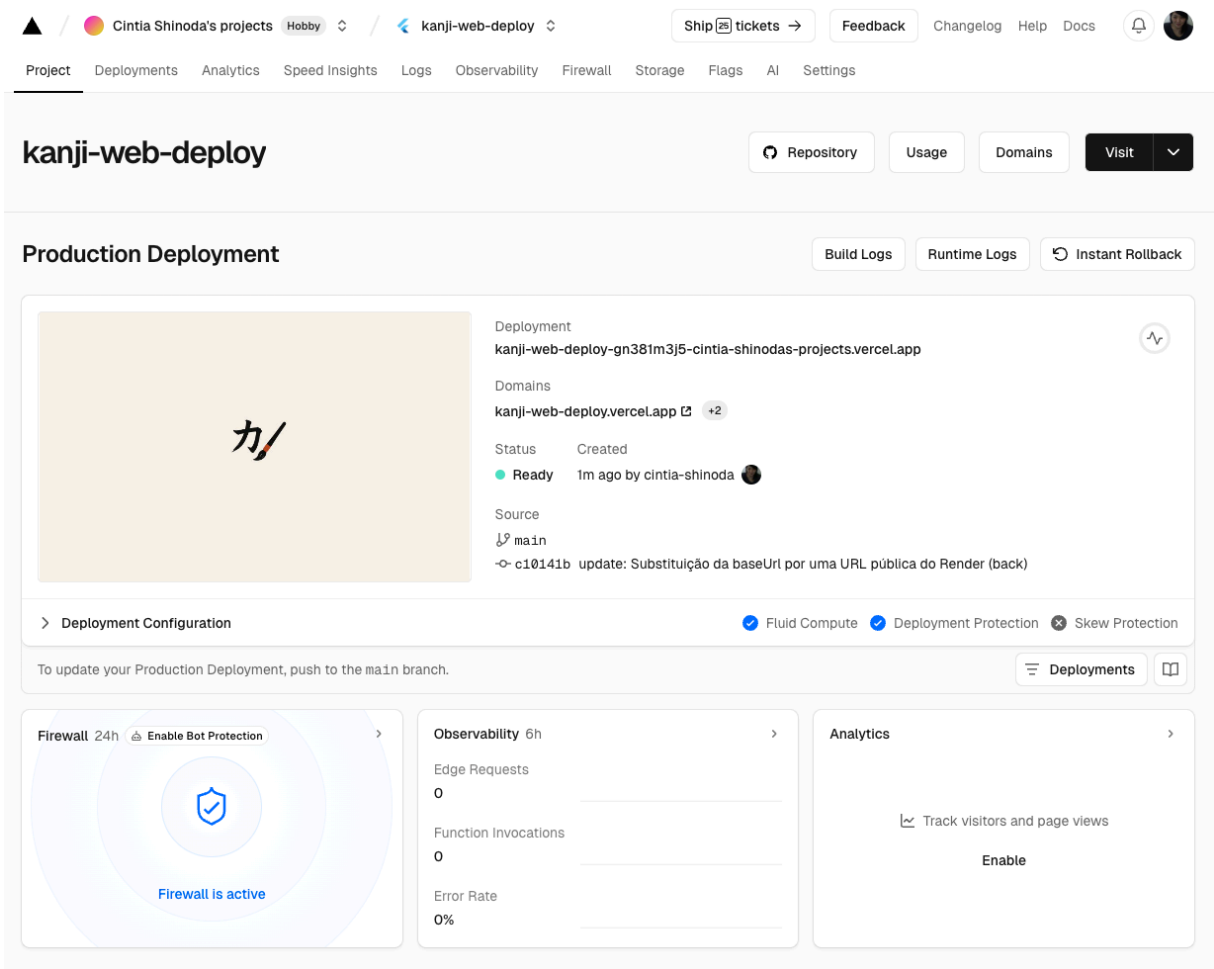
Um novo repositório GitHub foi criado exclusivamente para a versão compilada (*build*) do aplicativo Flutter Web. Ele pode ser acessado através do seguinte link:

<https://github.com/cintia-shinoda/kanji-web-deploy>.

No *dashboard* da Vercel (Figura 29), foi criado um projeto a partir do repositório mencionado anteriormente.

Após a conclusão do *deploy*, a aplicação, finalmente, se tornou disponível publicamente no endereço: <https://kanji-web-deploy.vercel.app>.

Figura 28 - Dashboard do *deploy* no Vercel



Fonte: SHINODA, Cintia Izumi. [Dashboard do deploy no Vercel]. Acesso restrito. Captura em: 11 mai. 2025.

## 4 CONSIDERAÇÕES FINAIS

O presente projeto teve como objetivo principal a criação de um aplicativo móvel voltado ao aprendizado e prática de Kanji, ideogramas essenciais para o domínio da língua japonesa.

A proposta surgiu da identificação de uma demanda crescente por métodos mais acessíveis, interativos e eficientes de ensino do idioma, especialmente no que se refere ao Kanji, que representa uma das maiores dificuldades para estudantes.

Por meio da aplicação dos conhecimentos adquiridos ao longo das disciplinas estudadas, nosso grupo buscou desenvolver uma solução tecnológica que não apenas facilitasse o processo de aprendizado, mas também promovesse o engajamento contínuo dos usuários por meio de práticas gamificadas e personalizadas.

A intenção do grupo foi proporcionar uma ferramenta educacional moderna que impactasse positivamente a jornada de estudo dos usuários, contribuindo para a fixação dos ideogramas, o acompanhamento da evolução individual e a criação de uma rotina de estudos mais eficaz. Acreditamos que, ao otimizar esses aspectos, conseguimos não apenas tornar o aprendizado mais eficiente e motivador, mas também democratizar o acesso a um conteúdo tradicionalmente restrito a materiais pagos ou cursos presenciais, ampliando o alcance educacional e cultural do idioma japonês.

Entretanto, é importante destacar que o projeto enfrentou limitações significativas, sobretudo devido ao tempo reduzido para sua execução e à complexidade de implementação de funcionalidades mais avançadas, como reconhecimento de escrita e análise de desempenho em tempo real. Além disso, a necessidade de testes com usuários reais foi parcialmente comprometida pela dificuldade de reunir uma base diversificada de estudantes para validação. Apesar desses desafios, o grupo permanece comprometido em dar continuidade ao projeto nos próximos meses, com o intuito de realizar os testes necessários, refinar a aplicação e disponibilizá-la futuramente em plataformas móveis.

Acreditamos que, ao oferecer uma ferramenta tecnológica acessível e eficiente, estamos contribuindo para a valorização do ensino de idiomas e para a formação de aprendizes mais autônomos e motivados.

O impacto educacional e cultural gerado por essa iniciativa reflete o potencial transformador da tecnologia quando aplicada com propósito e responsabilidade.

## REFERÊNCIAS

APEL, Ulrich. **The Kanji Vector Graphics (KanjiVG) project – KanjiVG**. KanjiVG, 2025. Disponível em: <https://kanjivg.tagaini.net/projects.html>. Acesso em: 17 abr. de 2025.

BAI, Y. (2020). *Flutter from Zero: A complete guide to web and mobile app development using Flutter and Dart*. Independently published.

BLANK, S. (2013). *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. K&S Ranch

BROWN, T. (2008). *Design Thinking*. Harvard Business Review, 86(6), 84-92.

BROWN, T. (2009). *Change by Design: How Design Thinking Creates New Alternatives for Business and Society*. Harper Business.

CHEN, L., Xu, H., & WANG, T. (2022). *Cross-Platform Mobile Development: A State-of-the-Art Survey*. IEEE Access, 10, 12345-12359.

CHUN, D. *Using handheld technology for language learning: Benefits and challenges*. *Language Learning & Technology*, v. 15, n. 3, p. 3-10, 2011.

DEFRANCIS, J. *The Chinese language: Fact and fantasy*. Honolulu: University of Hawaii Press, 1984.

GONZALEZ, A., Rojas, A., & Salinas, M. (2021). *Analyzing the Performance of Flutter Applications: A Benchmark Study*. Proceedings of the International Conference on Software Engineering.

GOTHELF, J. (2013). *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly Media.

GRAHAM, R. (2020). *Product Management in Practice: A Real-World Guide to Delivering Products that Customers Love*. O'Reilly Media.

HUANG, J., & ZHANG, Y. (2020). *Flutter Widgets: A Comprehensive Guide to Building Beautiful UIs with Flutter*. Packt Publishing.

KANJIVG. **KanjiVG** – **GitHub**. GitHub, 2025. Disponível em: <https://github.com/KanjiVG/kanjivg>. Acesso em: 17 abr. de 2025.

KANO, C. *The learning process of kanji characters by second language learners*. Tokyo: Japan Language Institute, 1995.

KELLEY, T., & KELLY, D. (2013). *Creative Confidence: Unleashing the Creative potential within us all*. Crown Business.

KINDAICHI, H. (1988). *The Japanese Language*. Tuttle Publishing.

KURTZ, A., Soma, B. & VESELOV, D. (2021). *Hot Reload and Its Effective Usage in Flutter Development*. Journal of Software Engineering and Development, 10(4), 283-295.

LEE, J. (2021). *The Complete Guide to Flutter Development: Cross-Platform Apps*. CreateSpace Independent Publishing Platform.

LIEDTKA, J. (2015). *Why Design Thinking Works*. Harvard Business Review, 93(9), 72–79.

MARTIN, R. L. (2009). *The Design of Business: Why Design Thinking is the Next Competitive Advantage*. Harvard Business Press.

MEYER, S. (2021). *Leap: How to Thrive in a World Where Everything is Coming at You*. Penguin.

MILLER, R. A. *The Japanese language*. Chicago: University of Chicago Press, 1967.

MISHRA, A., Kumar, S., & Prasad, S. (2021). *Building Applications with Flutter and Firebase: Firebase Integration with Flutter Applications*. Packt Publishing.

MORI, Y.; Shimizu, H. *Digital tools for kanji learning*. Journal of Japanese Language Education, v. 12, p. 45-62, 2017.

SHIBATANI, M. (1990). *The languages of Japan*. Cambridge University Press.

NIMAVATI, H. (2023). *Flutter for Beginners: An Introduction to Cross-Platform App Development*. Springer.

PRASAD, V. (2022). *Understanding Flutter: A Beginner's Guide to App Development*. Independently published.

RAZZANO, A., Brown, T., & Cross, N. (2019). *Rapid Prototyping in Designing Apps*. Journal of Visual Languages & Computing, 613, 159-172.

REDDY, D. (2021). *The Complete Flutter Development Bootcamp: Build iOS and Android Apps with Flutter*. John Wiley & Sons.

RIES, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business.

SINHA, S. (2021). *Cross-Platform Development Using Flutter*. International Journal of Computer Applications, 174(33), 10-16.



## **ANEXO A**

### **Ata da 1ª Reunião sobre Desenvolvimento de Aplicativo para Aprendizado de Kanji**

**Data:** 4 de abril de 2025

**Local:** Reunião virtual

#### **Participantes:**

- Cintia Izumi Shinoda - UNIVESP
- Fernando Miguel Escribano Martinez - UNIVESP
- Carlos Hideaki Fujinaga - Aliança Cultural Brasil-Japão

#### **Assuntos Discutidos:**

- **Introdução e Alinhamento Inicial:** A reunião teve como objetivo principal alinhar expectativas entre o grupo de estudantes da Univesp (Fernando e Cintia) e a Aliança sobre o desenvolvimento de um aplicativo para aprendizado de Kanji. Carlos representou a Aliança nesta reunião inicial.
- **Objetivo do Projeto:** Foi estabelecido que o objetivo principal é o desenvolvimento de um protótipo de aplicativo para um trabalho de universidade. A Aliança poderá utilizar o aplicativo futuramente, mas o foco inicial é atender aos requisitos acadêmicos, incluindo testes do protótipo.
- **Escopo do Aplicativo:**
  - O aplicativo será focado no aprendizado da leitura e da escrita de Kanji, sendo este um subconjunto do vasto universo da língua japonesa.
  - Foi reconhecida a amplitude do estudo de Kanji, mencionando a existência de exames de proficiência complexos.
  - Foram levantadas possíveis funcionalidades a serem consideradas, como o desmembramento de Kanji em radicais e partículas, agrupamento de ideogramas com

radicais comuns e a inclusão de vocabulário composto por Kanji (*jukugo*, especialmente *yojijukugo*).

- Será adotada uma abordagem de MVP (Produto Mínimo Viável), lançando uma versão básica para obter *feedback* dos usuários e adicionar funcionalidades posteriormente.
- **Cronograma:** Foi enfatizada a urgência do prazo, com a entrega final prevista para o final de maio, restando pouco mais de um mês para desenvolvimento e testes.
- **Divulgação da Aliança:** Discutiu-se o potencial do aplicativo como ferramenta de divulgação dos cursos da Aliança, podendo ser apresentado em eventos relacionados à cultura japonesa, como os de anime e mangá.
- **Infraestrutura:** A infraestrutura inicial para a aplicação será um ambiente de teste (homologação). A infraestrutura para um possível ambiente de produção pela Aliança será discutida em um momento futuro.
- **Protótipo das Telas:** Cintia informou ter enviado para Carlos um protótipo das telas do aplicativo.
- **Parceria e Autorização:**
  - Carlos mencionou a importância de apresentar o direcionamento do aplicativo aos demais membros da diretoria da Aliança.
  - A universidade requer uma autorização formal da Aliança para o uso do nome e de imagens em apresentações e relatórios do projeto.
  - Cintia se responsabilizou por enviar o modelo de termo de autorização por e-mail (através do e-mail institucional da universidade).
- **Evento na Aliança:** Carlos convidou Fernando e Cintia para uma palestra de antropologia sobre a imigração japonesa e okinawana no Brasil, que ocorreria no dia seguinte.
- **Novas Unidades da Aliança:** Fernando questionou sobre as unidades da Aliança, notando uma divergência entre o site (Vergueiro) e o Instagram (Pinheiros e Liberdade). Carlos

confirmou que as unidades atuais são Pinheiros e Liberdade (Rua da Glória) e que o site precisa ser atualizado.

- **Próximos Passos:** Ficou acordado que, após a obtenção da autorização, será agendada uma nova reunião (presencial ou por Zoom) para discutir em mais detalhes as funcionalidades do aplicativo e apresentar o projeto à diretoria da Aliança.

**Considerações Finais:** Ambas as partes demonstraram entusiasmo com a potencial colaboração e esperam que esta seja a primeira de muitas parcerias.

## **ANEXO B**

### **Ata da 2ª Reunião - Apresentação de projeto de aplicativo educacional (UNIVESP) e discussão de potencial parceria (Aliança)**

**Data:** 15 de maio de 2025

**Horário:** 15h00 às 15h45. A reunião tinha um limite de tempo mencionado de aproximadamente 40 minutos.

**Local:** Reunião virtual

#### **Participantes:**

- Cláudio Takeda, presidente Aliança Cultural Brasil-Japão
- Victor Shimabukuro, diretor administrativo adjunto / jurídico Aliança Cultural Brasil-Japão
- Carlos Fujinaga, diretor cultural Aliança Cultural Brasil-Japão
- Luzia Takayasu, gerente administrativa Aliança Cultural Brasil-Japão
- William Soares, coordenador de ensino Aliança Cultural Brasil-Japão
- Cintia Izumi Shinoda, UNIVESP
- Fernando Miguel Escribano Martinez, UNIVESP
- Pedro Henrique Faria Cruz, UNIVESP

#### **1. Abertura da Reunião**

A reunião foi iniciada com as boas-vindas e apresentações aos participantes.

Carlos Fujinaga apresentou os membros da Aliança presentes. Em seguida, Fernando, representante dos estudantes da UNIVESP, agradeceu a oportunidade e iniciou a apresentação do projeto. Foi formalizado o agradecimento ao interesse e participação de todos, tanto do grupo da UNIVESP quanto dos representantes da Aliança.

A entrada e participação de Cláudio Takeda, Presidente da Aliança, foi destacada e agradecida.

Carlos Fujinaga se apresentou como Diretor Voluntário da Aliança.

Dr. Victor Miranda se apresentou como Diretor Jurídico e Adjunto Administrativo, indicando que seu papel seria mais de assessorar questões de parceria e aspectos técnicos do aplicativo.

## **2. Apresentação do Projeto de Aplicativo de Kanji (UNIVESP)**

Fernando, aluno da UNIVESP, apresentou a universidade: pública, totalmente virtual, mantida pelo governo, criada em 2012 com foco na democratização e descentralização do ensino, com aulas online, material disponível na internet e avaliação presencial obrigatória em polos. Explicou o contexto da disciplina "Projeto Integrador", na qual os alunos identificam problemas em comunidades para buscar soluções, buscando desenvolver habilidades em cenários reais e dar retorno à sociedade como universidade pública. A ideia do projeto surgiu de um insight da aluna Cintia, que estuda japonês.

Fernando e Cintia apresentaram um protótipo de aplicativo educacional voltado ao estudo de kanji, ideogramas japoneses.

### **Objetivos do Projeto:**

- Criar uma ferramenta de apoio lúdica e interativa para o aprendizado de kanji.
- Desenvolver um aplicativo gratuito, em português, acessível a estudantes da língua japonesa.
- Utilizar o projeto como trabalho prático na disciplina "Projeto Integrador" da UNIVESP.
- Enriquecer a experiência dos alunos da Aliança e servir como ferramenta de atração e retenção.

Um benchmark identificou a falta de ferramentas no mercado com foco em Kanji, em português e gratuitas.

O conceito do aplicativo é ser uma ferramenta interativa para treinar a escrita e identificação de kanjis, funcionando como um passatempo ou joguinho. Espera-se que o aplicativo ajude no aprendizado, sirva como reforço das aulas e tenha interface em português.

Pode ter disseminação independente da matrícula na Aliança.

A solução inicial é simples, permitindo cadastro/login e escrita dos traços na tela, com pontuação (em implementação).

### **Características da Proposta:**

- Aplicativo multiplataforma (mobile e desktop).
- Funcionalidade de escrita de kanji com pontuação por traço correto.
- Protótipo desenvolvido com base no conceito de Produto Mínimo Viável (MVP). O protótipo ainda não é um produto acabado ou profissional, e a ideia é que este conceito básico seja validado pela Aliança, solicitando retorno sobre o que seria interessante adicionar ou modificar.
- Código-fonte aberto e disponível no GitHub.
- Possibilidade de uso independente da matrícula na Aliança.
- Foram mostradas telas de login e escrita, e apresentado um pequeno vídeo do funcionamento do protótipo, disponível para celular e multiplataforma. Um QR Code foi disponibilizado para que os participantes pudessem baixar e testar o protótipo, que está em uma plataforma provisória apenas para validação do conceito.
- Os estudantes da UNIVESP reiteraram que buscam principalmente feedback da Aliança e dos usuários sobre a utilidade do aplicativo e caminhos de melhoria.

### **3. Discussão e Feedback Inicial**

Houve agradecimento pela apresentação. Carlos mencionou ter dado algumas opiniões em uma reunião prévia.

- Dr. Victor Miranda:
  - Parabenizou a iniciativa.
  - Reforçou que a análise técnica será realizada pelas áreas pedagógica e administrativa da Aliança.
  - Questionou se a parceria seria gratuita. Os alunos confirmaram que sim, sendo o projeto aberto e sem fins lucrativos. Dr. Victor confirmou que a parceria seria gratuita para a Aliança, uma vez que é um projeto de escola e o código fonte é aberto.
  - Delegou a análise técnica e pedagógica para Carlos e Professor William.
- Prof. William Soares:

- Ofereceu feedback detalhado.
  - Sugeriu a implementação de *tags* para classificar os kanji e criar coleções.
  - Exemplos de *tags* sugeridas: Por nível do JLPT (prova de proficiência) ou por módulos específicos dos materiais da Aliança. Isso permitiria aos alunos estudar grupos específicos de Kanji.
  - Mencionou também a possibilidade de *tags* personalizadas pelo usuário.
  - Ressaltou a importância crucial da escolha da fonte tipográfica usada no aplicativo, para que represente corretamente a forma manuscrita dos ideogramas.
  - Destacou a diferença entre a escrita manual e digital e como a fonte pode influenciar a forma como os alunos aprendem a escrever.
  - Observou que fontes inadequadas, comuns em outros aplicativos, podem comprometer a aprendizagem dos alunos.
  - Ofereceu-se para sugerir fontes, como as do Google Fonts, se possível.
- Carlos Fujinaga:
    - Concordou com as sugestões sobre as *tags*.
    - Reforçou que a adaptação ao conteúdo da Aliança pode ser um diferencial do aplicativo.
    - Reforçou que o projeto ainda está em fase inicial e será aperfeiçoado conforme os feedbacks.
- Metodologia e Continuidade do Projeto (Discutido neste ponto):
    - Fernando reforçou a abordagem de MVP, sugerindo que em um primeiro momento seja implementada a parte básica do aplicativo para fomentar discussão e captar necessidades, e só depois ir acrescentando funcionalidades com o tempo. Explicou que fazer um aplicativo completo de início é custoso.
    - Foi levantada a questão sobre o prosseguimento do projeto após este período inicial (se poderia evoluir para um projeto profissional etc.). Fernando indicou que isso precisaria ser discutido em uma segunda reunião para alinhar expectativas, especialmente com Cintia. No momento, o foco é validar o conceito do protótipo.

#### **4. Encaminhamentos e Próximos Passos**

- A equipe da UNIVESP aguarda retorno da Aliança sobre a utilidade e adequação do aplicativo.
- A Aliança avaliará internamente a proposta junto à equipe pedagógica.
- Aliança (representantes como William e Carlos) fornecerá feedback detalhado sobre o protótipo de aplicativo, incluindo sugestões de *tags* e considerações sobre a fonte.
- Pendência de Documentação (Termos/Acordo):
  - Fernando lembrou Carlos da urgência em receber o retorno sobre os termos/acordo que Cintia havia enviado. Explicou que a não entrega destes termos prejudica a avaliação dos estudantes com o orientador.
  - Carlos mencionou não ter recebido e solicitou que fossem encaminhados novamente.
  - Dr. Victor Miranda solicitou que os termos fossem encaminhados para seu e-mail institucional para análise e devido encaminhamento, comprometendo-se a fazer isso até a sexta-feira (16 de maio de 2025) se recebê-los. Solicitou a gentileza do envio via e-mail para registro.
  - Ação Imediata: Grupo UNIVESP (Cintia/Fernando) reenviar os termos/acordo para o e-mail institucional do Dr. Victor Miranda na Aliança.
  - Ação da Aliança: Dr. Victor Miranda analisar os termos recebidos e fornecer considerações/encaminhamentos até a sexta-feira (16 de maio de 2025).
- Ficou aberta a possibilidade de agendar nova reunião para aprofundar as contribuições e alinhar expectativas futuras.
  - Ação: Agendar uma segunda reunião para discutir o futuro e o prosseguimento do projeto.
- Foi acordado definir prazos gerais para que o projeto não caia no esquecimento.
  - Ação: Definir prazos gerais para as próximas etapas do projeto.

#### **5. Encerramento**

A reunião foi encerrada com agradecimentos mútuos e elogios ao esforço e dedicação dos estudantes na elaboração de um projeto com potencial de aplicação prática no ambiente educacional.



## APÊNDICE A

### Script verificar\_kanji.js

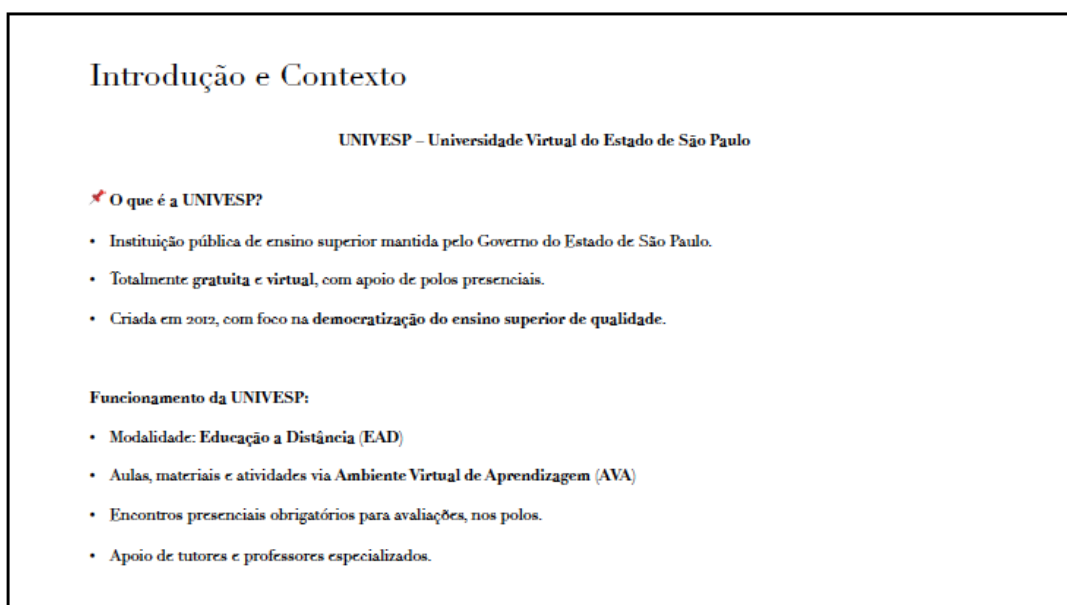
Desenvolvido para verificar se os kanjis a serem inseridos no banco de dados estão adequados.

```
Users > cintiashinoda > Desktop > Pl-3 > scripts-kanji > verificar_kanji.js > ...
1  const mongoose = require('mongoose');
2
3  mongoose.connect('mongodb+srv://cintias@cluster0-kanji.mongodb.net/kanji_app?retryWrites=true&w=majority&appName=kanji-cluster', {
4    useNewUrlParser: true,
5    useUnifiedTopology: true,
6  });
7
8  const Kanji = mongoose.model('Kanji', new mongoose.Schema({}, { strict: false }));
9
10 (async () => {
11   const kanjis = await Kanji.find();
12
13   let total = 0;
14   let comErros = 0;
15
16   for (const k of kanjis) {
17     total++;
18     const problemas = [];
19
20     if (!Array.isArray(k.tracos) || k.tracos.length === 0) {
21       problemas.push('✗ tracos ausentes ou vazios');
22     } else {
23       for (let i = 0; i < k.tracos.length; i++) {
24         const t = k.tracos[i];
25         if (
26           typeof t.ordem !== 'number' ||
27           typeof t.svg !== 'string' ||
28           !t.ponto_inicio || typeof t.ponto_inicio.x !== 'number' || typeof t.ponto_inicio.y !== 'number' ||
29           !t.ponto_fim || typeof t.ponto_fim.x !== 'number' || typeof t.ponto_fim.y !== 'number'
30         ) {
31           problemas.push(`⚠ Traço inválido na posição ${i}`);
32         }
33       }
34     }
35
36     if (problemas.length > 0) {
37       comErros++;
38       console.log(`\nKANJI: ${k.kanji}`);
39       problemas.forEach(p => console.log(`  ${p}`));
40     }
41   }
42
43   console.log(`\n✅ Kanjis verificados: ${total}`);
44   console.log(`✗ Kanjis com problemas: ${comErros}`);
45
46   mongoose.disconnect();
47 })();
```

## APÊNDICE B

### Apresentação de slides utilizada na 2ª reunião com a Aliança Cultural Brasil-Japão

A seguir, a versão miniaturizada da apresentação exibida durante a reunião realizada com parte da diretoria da Aliança Cultural Brasil-Japão realizada no dia 15 de maio de 2025. Contém os principais pontos, funcionalidades e demonstrações do projeto.



## Introdução e Contexto

### Projeto Integrador

- **O que é:** Disciplina fundamental/obrigatória para aplicar conhecimentos de computação na solução de problemas REAIS, integrando teoria e prática, em grupos.
- **Propósito / Valor Estratégico:**
  - Formar profissionais com competências para resolver desafios tecnológicos reais da sociedade e empresas.
  - Desenvolver habilidades essenciais para o mercado: resolução de problemas, trabalho em equipe, inovação, autonomia e pensamento crítico.
  - Preparar o estudante para o mercado de trabalho através da experiência prática.
- **Como Funciona (Síntese do Processo Semestral):**
  - Identificação do Problema Real.
  - Desenvolvimento de Soluções Viáveis e Inovadoras.
  - Entrega e Apresentação de Resultados.

## Introdução e Contexto

### Equipe do Projeto

- **Nomes:**
  - Camila Pereira da Silva
  - Cintia Izumi Shinoda
  - Fernando Miguel Escribano Martinez
  - Henrique Yassuyuki Tsuboi
  - Pedro Henrique Faria Cruz
  - Rogério Gonçalves da Silva
  - Vitor Paes de Barros
  - Willy Paulino de Oliveira Gomes
- **Curso:** Bacharelado em Ciência de Dados / Tecnologia da Informação / Engenharia de Computação
- **Universidade:** Universidade Virtual do Estado de São Paulo (UNIVESP)
- **Tutor:** Dhyogo Nunes Costa

## Oportunidade para a Aliança Cultural Brasil-Japão

- Alinhamento com a Missão da Aliança
- Pontos:
  - Aliança: Papel crucial na promoção do intercâmbio cultural e ensino da língua japonesa no Brasil.
  - Reconhecida pela qualidade e métodos (ex: Marugoto).
  - Crescente interesse pela cultura e língua japonesa (inclusive por não descendentes).
  - Tecnologia como aliada: Modernizar, ampliar alcance, engajar alunos e fortalecer a posição da Aliança como referência.
  - Este projeto visa criar uma ferramenta que pode *potencialmente* integrar e enriquecer a experiência dos alunos da Aliança.

## Cenário Atual: Ferramentas de Aprendizado

### Análise do Mercado: Aplicativos para Aprender Japonês/Kanji

- **Introdução:** Realizamos um benchmark para entender as soluções existentes, seus pontos fortes e fracos, focando em:
  - Funcionalidades
  - Disponibilidade em Português
  - Custo (Gratuidade)
  - Foco específico no aprendizado de Kanji

## Tabela Comparativa

Aplicativo	Foco em Kanji	Português	Gratuito
Duolingo	✗	✓	✓
LingoDeer	✓	✓	●
Mondly	✗	✓	●
Kanji Study	✓ ✓ ✓	✗	●
Bunpro	⚠	✗	✗
WaniKani	✓ ✓ ✓	✗	●
JA Sensei	✓	✗	●
Hiragana Quest	✗	✗	●
Memrise	⚠	✓	✓
Obenkyo	✓	✗	✓

✓✓✓ Foco total em kanji

✓ Introduce kanjis

● Parcialmente gratuito

⚠ Não voltado diretamente

Obs.: Avaliamos desde apps gerais de idiomas até ferramentas mais especializadas.

## Objetivo Geral do Projeto

- **Proposta:** Um Aplicativo Dedicado ao treino de Kanji em Português para a Comunidade.
- **Objetivo Específico:**
  - Desenvolver um aplicativo interativo (*exercícios e quizzes*) para facilitar o aprendizado, memorização e a prática (leitura e escrita) de Kanjis.
- **Público:** Estudantes da Aliança, autodidatas, entusiastas da língua japonesa no Brasil.

## Parceria com a Aliança Cultural Brasil-Japão

### Benefícios para Alunos e Aliança

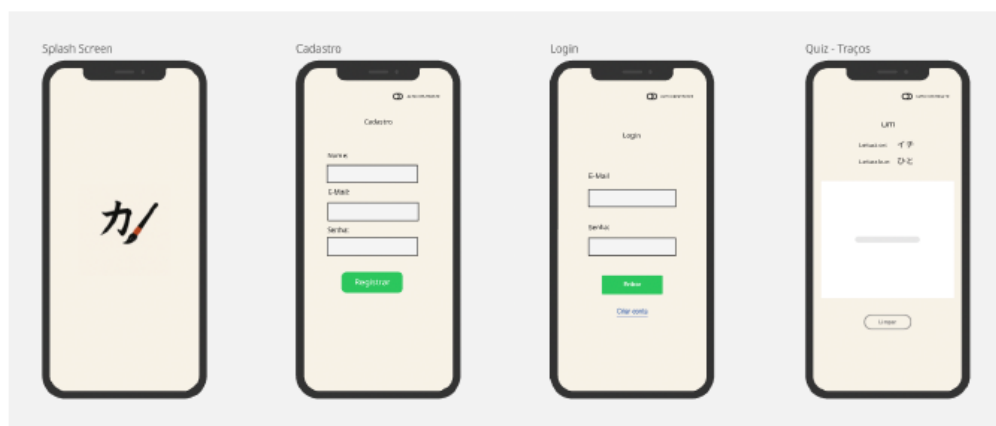
- Para os Alunos:
  - Ferramenta focada na maior dificuldade (Kanji).
  - Aprendizado interativo, engajador e gamificado (potencial).
  - Flexibilidade para estudar a qualquer hora/lugar.
  - Reforço do conteúdo aprendido em aula.
  - Interface em português.
- Para a Aliança:
  - Recurso digital inovador para complementar os cursos.
  - Ferramenta de apoio para professores e alunos.
  - Potencial para atrair e reter alunos.
  - Reforço da imagem da Aliança como instituição moderna e atenta às necessidades dos estudantes.
  - Possibilidade de customização futura alinhada aos materiais da Aliança.

## Solução Inicial

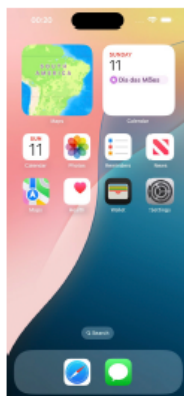
### Recursos Essenciais do Aplicativo:

- Cadastro e Login de Usuário
- Módulo de Prática de Escrita de Kanji (guiada por traços, reconhecimento)
- Visualização de Pontuação (a ser implementado)

## Protótipo das telas do App



## Vídeo com os Resultados Preliminares



## Considerações Finais

- **Próximos Passos:**

- Continuar o desenvolvimento (adicionar mais Kanjis e funcionalidades);
- Realizar testes com usuários reais (estudantes);
- Refinar a interface e a experiência do usuário;
- Disponibilizar em plataformas móveis (Android/iOS).

- **Convite à Aliança:**

- Gostaríamos do *feedback* da diretoria sobre a proposta;
- Explorar possibilidades de colaboração (teste piloto com alunos, alinhamento de conteúdo);
- Acreditamos que este projeto pode ser um recurso valioso para a comunidade da Aliança.

Vamos testar?





## Agradecimentos

Agradecemos a oportunidade de apresentar este projeto.

Estamos entusiasmados com o potencial de usar a tecnologia para facilitar o aprendizado do Kanji e apoiar a missão da Aliança Cultural Brasil-Japão.

Caso queira acompanhar o desenvolvimento, o repositório do projeto pode ser acessado em:  
<https://github.com/cintia-shinoda/projeto-integrador-3>