SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

CARLA CAROTA MOZENA

LETICIA APARECIDA HOFMAN DE SOUZA

**LEKA**

**APRENDA JOGANDO**

JAGUARIUNA-SP

2024

CARLA CAROTA MOZENA

LETICIA APARECIDA HOFMAN DE SOUZA

**LEKA**

**APRENDA JOGANDO**

Trabalho parcial de conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da escola SENAI Jaguariúna.

Orientadores: Wellington Fábio

JAGUARIUNA-SP

2024

CARLA CAROTA MOZENA

LETICIA APARECIDA HOFMAN DE SOUZA

**LEKA – APRENDA JOGANDO**

Trabalho parcial de conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da escola SENAI Jaguariúna, como requisito parcial para obtenção do grau de Técnico em Desenvolvimento de Sistemas.

Aprovado em 04 de dezembro de 2024.

**BANCA EXAMINADORA**



**O código que você escreve pode ser um produto, mas a forma como você o escreve pode ser uma arte.**

Anonymous

RESUMO

O aplicativo LEKA visa promover o aprendizado de habilidades de programação de forma gamificada e personalizada. Ele utiliza questionários iniciais para entender os objetivos do usuário, como razões para o aprendizado (profissional, diversão etc.) e nível de experiência. Com base nas respostas, o aplicativo sugere trajetórias de aprendizado adaptadas, estabelecendo metas diárias e progresso visual por meio de gráficos. Além disso, LEKA incentiva a competição saudável, permitindo que os usuários convidem amigos para participar. O aplicativo também oferece funcionalidades para criar resumos personalizados das aulas e acompanhar o progresso, certificados e recordes, tudo dentro de um ambiente interativo e envolvente.

**Palavras-Chaves:** Controle; Saída; Entrada; Desenvolvimento; Sistema.

ABSTRACT

The LEKA app is a gamified learning platform designed to teach programming skills in an interactive and personalized way. It begins with questionnaires to understand each user’s goals, such as reasons for learning (professional, recreational etc.) and experience level. Based on these responses, LEKA suggests tailored learning paths, sets daily goals, and displays progress visually through graphs, fostering consistent engagement. Beyond learning, the app encourages healthy competition by allowing users to invite friends and track each other’s progress. Features like personalized lesson summaries, tracking of progress, certificates, and achievements create an immersive and motivating environment, supporting users' autonomy and enthusiasm in their learning journey.

**LISTA DE abreviaturas e SIGLAS**

Ágil. Metodologia Ágil (Metodologia de desenvolvimento de sistemas).

API. Application Programming Interface (Interface de programação de aplicação).

BackEnd. Prática de programação para Servidor de Aplicação e ou Banco de Dados.

Backlog. Requisitos funcionais do projeto a serem desenvolvidos.

CPM. (Critical Path Method) é uma técnica utilizada para identificar o caminho crítico de um projeto, ou seja, a sequência de atividades que determina a duração total do projeto.

CRUD. Create, Read, Update, Delete.

CSS. Linguagem de marcação e estilização de páginas.

DCU. Diagrama de Casos de Uso.

DER. Diagrama de Entidade e Relacionamento.

EAP. Estrutura Analítica do Projeto

Figma. Ferramenta de design de interface do usuário baseada em navegador, que oferece a possibilidade de colaboração em tempo real.

Framework. Estrutura, conjunto de códigos genéricos capaz de unir trechos de um projeto.

FrontEnd. Prática de programação para cliente de programas para internet.

FullStack. Prática de programação em todas as camadas Cliente e Servidor.

Gantt. Gráfico para gestão do tempo e recursos do projeto.

GitHub. Repositório para códigos fonte e rede social de programadores.

HTML. Hyper Text Management Language, linguagem de marcação.

JavaScript. Linguagem de programação para a internet.

JSON. Objeto JavaScript, padrão de dados de uma API.

MER. Modelo Entidade e Relacionamento.

Mobile. Prática de programação para dispositivos móveis também clientes de programas para a internet.

MVC. Modelo Visão e Controle.

Node.js. Framework que utiliza a linguagem JavaScript no backend.

PERT. (Program Evaluation and Review Technique) é uma técnica utilizada para estimar o tempo necessário para a conclusão de um projeto.

Protótipo. Demonstra ao cliente como será o produto.

SCRUM. Subcategoria da Metodologia Ágil.

Sprint. Tempo de desenvolvimento de duas a quatro semanas.

SWOT. Serve para analisar a viabilidade de um projeto.

TAP. Termo de Abertura do Projeto.

UML. Unified Modeling Langue (Linguagem de Modelagem Unificada).

**LISTA DE FIGURAS**

[Figura 7-1 - Tela Inicial 22](#_Toc182599322)

[Figura 7-2 - Tela de Login 23](#_Toc182599323)

[Figura 7-3 - Tela inicial de Cadastro 24](#_Toc182599324)

[Figura 7-4 - Tela de Cadastro, após Login 25](#_Toc182599325)

[Figura 7-5 - Tela de Opções 26](#_Toc182599326)

[Figura 7-6 - Tela Nível de Experiência 27](#_Toc182599327)

[Figura 7-7 - Tela de Aspecto Preferido 28](#_Toc182599328)

[Figura 7-8 - Tela Área de interesse pessoal 29](#_Toc182599329)

[Figura 7-9 - Tela Escolha Pessoal Realizada 30](#_Toc182599330)

[Figura 7-10 - Tela Tempo de Estudo 31](#_Toc182599331)

[Figura 7-11 - Tela Conclusão tempo do Curso 32](#_Toc182599332)

[Figura 7-12 - Tela lembrete diário 33](#_Toc182599333)

[Figura 7-13 - Tela progresso do Aluno 34](#_Toc182599334)

[Figura 7-14 - Tela Início do Curso 35](#_Toc182599335)

[Figura 7-15 - Tela Amigos Online 36](#_Toc182599336)

[Figura 7-16 - Tela Criar Resumos 37](#_Toc182599337)

[Figura 7-17 - Tela Perfil do Aluno 38](#_Toc182599338)

[Figura 8-1 - Ilustrativo DCU Token 47](#_Toc182599339)

[Figura 8-2 - Ilustrativo DCU Registrar 48](#_Toc182599340)

[Figura 8-3 - Ilustrativo DCU Questionário Inicial 49](#_Toc182599341)

[Figura 8-4 - Ilustrativo DCU Definição de Metas 50](#_Toc182599342)

[Figura 8-5 - Ilustrativo DCU Progresso e Etapas 51](#_Toc182599343)

[Figura 8-6 - Ilustrativo DCU Atividades 52](#_Toc182599344)

[Figura 8-7 - Ilustrativo DCU Interação com Amigos 53](#_Toc182599345)

[Figura 8-8 - Ilustrativo DCU Conquistas e Recompensas 54](#_Toc182599346)

[Figura 8-9 - Ilustrativo DCU Resumos Personalizados 55](#_Toc182599347)

[Figura 8-10 - Ilustrativo DCU Gerenciamento de Perfil 56](#_Toc182599348)

[Figura 8-11 - Ilustrativo DCU Competição em Salas 57](#_Toc182599349)

[Figura 9-1 - Caminho Crítico 65](#_Toc182599350)

[Figura 10-1 - Diagrama de Classes 68](#_Toc182599351)

[Figura 10-2 - Diagrama de Atividades 69](#_Toc182599352)

[Figura 10-3 - Diagrama de Entidade Relacionamento 70](#_Toc182599353)

[Figura 11-1 - Tela Inicial 76](#_Toc182599354)

[Figura 11-2 - Tela de Login 77](#_Toc182599355)

[Figura 11-3 - Tela de Cadastro 78](#_Toc182599356)

[Figura 11-4 - Tela Questionário 01 79](#_Toc182599357)

[Figura 11-5 - Tela Questionário 02 80](#_Toc182599358)

[Figura 11-6 - Tela Questionário 03 81](#_Toc182599359)

[Figura 11-7 - Tela Questionário 04 82](#_Toc182599360)

[Figura 11-8 - Tela Resultado 83](#_Toc182599361)

[Figura 11-9 - Tela Tempo 84](#_Toc182599362)

[Figura 11-10 - Tela Gráfico Profissional 85](#_Toc182599363)

[Figura 11-11 - Tela Jogos 86](#_Toc182599364)

[Figura 11-12 - Tela Resumo 87](#_Toc182599365)

[Figura 11-13 - Tela Amigos (Chat) 88](#_Toc182599366)

[Figura 11-14 - Tela Comunidade 89](#_Toc182599367)

**LISTA DE TABELAS**

[Tabela 1 - Tabela Matriz SWOT 40](#_Toc181814723)

[Tabela 2 - Cronograma de Gantt 63](#_Toc181814724)

[Tabela 3 - Tabela PERT 64](#_Toc181814725)

[Tabela 4 - Tabela Orçamento 66](#_Toc181814726)

[Tabela 5 - Tecnologias utilizadas 87](#_Toc181814727)

**SUMÁRIO**

[1 INTRODUÇÃO 10](#_Toc182599410)

[1.1.1 Objetivo Geral 11](#_Toc182599411)

[1.1.2 Objetivos Específicos 11](#_Toc182599412)

[2 JUSTIFICATIVA 12](#_Toc182599413)

[3 TAP (TERMO DE ABERTURA DO PROJETO) 13](#_Toc182599414)

[4 METODOLOGIA 15](#_Toc182599415)

[4.1 Fases do Projeto: 15](#_Toc182599416)

[5 Metodologias Ágeis: 17](#_Toc182599417)

[5.1 Ferramentas e Tecnologias 17](#_Toc182599418)

[5.2 Tecnologias Utilizadas: 17](#_Toc182599419)

[5.3 Documentação e Comunicação: 18](#_Toc182599420)

[6 Resultados Esperados 18](#_Toc182599421)

[6.1 Funcionalidade Completa do Sistema: 18](#_Toc182599422)

[6.2 Integração Eficiente: 19](#_Toc182599423)

[6.3 Experiência do Usuário: 19](#_Toc182599424)

[6.4 Desempenho e Confiabilidade: 19](#_Toc182599425)

[6.5 Segurança dos Dados: 19](#_Toc182599426)

[6.6 Feedback Positivo: 19](#_Toc182599427)

[6.7 Documentação Adequada: 19](#_Toc182599428)

[6.8 Acessibilidade: 20](#_Toc182599429)

[6.9 Relatórios e Análises: 20](#_Toc182599430)

[6.10 Escalabilidade: 20](#_Toc182599431)

[6.11 Preparação para a Implantação: 20](#_Toc182599432)

[7 PROTÓTIPO 21](#_Toc182599433)

[7.1 Protótipo do Dispositivo Móvel 21](#_Toc182599434)

[7 EAP 39](#_Toc182599435)

[7.1 Análise de Riscos – SWOT 39](#_Toc182599436)

[7.2 Escopo do Projeto 40](#_Toc182599437)

[8 Documento de Regras de Negócio (RN) 42](#_Toc182599438)

[8.1 Regras de Negócio 42](#_Toc182599439)

[8.1.1 Geração e Validação de Token 42](#_Toc182599440)

[8.1.2 Registro e Login 43](#_Toc182599441)

[8.1.3 Questionário Inicial 43](#_Toc182599442)

[8.1.4 Definição de Metas 43](#_Toc182599443)

[8.1.5 Progresso e Etapas 44](#_Toc182599444)

[8.1.6 Atividades 44](#_Toc182599445)

[8.1.7 Interação com Amigos 44](#_Toc182599446)

[8.1.8 Conquistas e Recompensas 45](#_Toc182599447)

[8.1.9 Resumos Personalizados 45](#_Toc182599448)

[8.1.10 Gerenciamento de Perfil 45](#_Toc182599449)

[8.1.11 Competição em Salas 46](#_Toc182599450)

[8.2 Requisitos Funcionais (RF) 47](#_Toc182599451)

[8.2.1 RF001 – Gerar Token 47](#_Toc182599452)

[8.2.2 RF002 - Registro e Login 48](#_Toc182599453)

[8.2.3 RF003 - Questionário Inicial 49](#_Toc182599454)

[8.2.4 RF004 - Definição de Metas 50](#_Toc182599455)

[8.2.5 RF005 - Progresso e Etapas 51](#_Toc182599456)

[8.2.6 RF006 - Atividades 52](#_Toc182599457)

[8.2.7 RF007 - Interação com Amigos 53](#_Toc182599458)

[8.2.8 RF008 - Conquistas e Recompensas 54](#_Toc182599459)

[8.2.9 RF009 - Resumos Personalizados 55](#_Toc182599460)

[8.2.10 RF010 - Gerenciamento de Perfil 56](#_Toc182599461)

[8.2.11 RF011 - Competição em Salas 57](#_Toc182599462)

[8.3 Requisitos Não Funcionais (RNF) 58](#_Toc182599463)

[8.3.1 Segurança 58](#_Toc182599464)

[8.3.2 Usabilidade 59](#_Toc182599465)

[8.3.3 Confiabilidade 60](#_Toc182599466)

[8.3.4 Escalabilidade 61](#_Toc182599467)

[9 PLANEJAMENTO (CRONOGRAMA GRÁFICO DE GANTT) 63](#_Toc182599468)

[9.1 CPM – Caminho crítico 64](#_Toc182599469)

[9.1.1 PERT 64](#_Toc182599470)

[9.1.2 Caminho Crítico (CPM) 65](#_Toc182599471)

[9.2 Orçamento 66](#_Toc182599472)

[10 Documento de Arquitetura 67](#_Toc182599473)

[10.1 Diagrama de Classes 68](#_Toc182599474)

[10.2 Diagrama de Atividades 69](#_Toc182599475)

[10.3 MER – Der (Diagrama de Entidade Relacionamento) 70](#_Toc182599476)

[10.4 MER-DER (Dicionário de Dados) 71](#_Toc182599477)

[11 CONCLUSÃO/ENTREGA 72](#_Toc182599478)

[11.1 Evidências 73](#_Toc182599479)

[11.2 Front-End (Print das Telas) 75](#_Toc182599480)

[11.3 Mobile (Print das Telas) 76](#_Toc182599481)

[11.4 Testes 90](#_Toc182599482)

[12 DOCUMENTO DE IMPLANTAÇÃO 92](#_Toc182599483)

[12.1 Tecnologias Utilizadas 92](#_Toc182599484)

[12.2 Configuração do Ambiente de Desenvolvimento Web 93](#_Toc182599485)

[12.3 Instalação Mobile 93](#_Toc182599486)

[13 MANUAL DO USUÁRIO 95](#_Toc182599487)

[13.1 Acesso ao Sistema 95](#_Toc182599488)

[13.2 Tela “Inicial” 95](#_Toc182599489)

[13.3 Criando uma conta (Login) 96](#_Toc182599490)

[13.4 Tela “Cadastro” 97](#_Toc182599491)

[13.5 Tela “Motivo” 99](#_Toc182599492)

[13.6 Tela “Melhor Opção” 99](#_Toc182599493)

[13.7 Tela “Experiência” 100](#_Toc182599494)

[13.8 Tela “Interesse” 101](#_Toc182599495)

[13.9 Tela “Resultado” 102](#_Toc182599496)

[13.10 Tela “Tempo” 103](#_Toc182599497)

[13.11 Tela “Trilha” 104](#_Toc182599498)

[13.12 Tela “Game” 106](#_Toc182599499)

[13.13 Tela “Comunidade” 108](#_Toc182599500)

[13.14 Tela “Resumo” 112](#_Toc182599501)

[14 REFERÊNCIAS BIBLIOGRÁFICAS 116](#_Toc182599502)

# INTRODUÇÃO

O avanço das tecnologias de informação e comunicação transformou profundamente a forma como as pessoas acessam conhecimento e desenvolvem novas habilidades. Em particular, a área de programação tem se beneficiado dessas mudanças, oferecendo recursos digitais que facilitam o aprendizado, especialmente com o uso da gamificação — técnica que utiliza elementos de jogos em contextos de aprendizado para engajar e motivar os usuários (Deterding et al., 2011). Estudos apontam que a gamificação aumenta a motivação intrínseca e a retenção do conteúdo, especialmente em ambientes educacionais (Hamari et al., 2014). Com base nesse contexto, surge o Leka, um aplicativo de aprendizado gamificado que visa promover o desenvolvimento de habilidades de programação de forma interativa e personalizada.

O Leka diferencia-se por combinar a gamificação com personalização, um recurso essencial para atender às necessidades individuais de cada usuário. A personalização tem se mostrado uma prática eficaz em plataformas de aprendizado, permitindo que os conteúdos e atividades sejam adaptados de acordo com o perfil e os objetivos de cada estudante (Brusilovsky & Millán, 2007). No Leka, a jornada de aprendizado é ajustada com base nas respostas a questionários iniciais, que buscam compreender as motivações do usuário — sejam elas profissionais, pessoais ou de lazer — e seu nível de experiência. A partir dessas informações, o aplicativo sugere trajetórias de aprendizado adaptadas e estabelece metas diárias que facilitam o progresso contínuo e o engajamento do usuário.

1. **OBJETIVOS**

### Objetivo Geral

O objeto geral deste trabalho é desenvolver um aplicativo de aprendizado de programação gamificado e personalizado, que promova o engajamento e a autonomia dos usuários, proporcionando uma jornada de aprendizado adaptada aos diferentes níveis de conhecimento e interesses.

### Objetivos Específicos

* 1. **Criar um questionário inicial** para identificar as motivações e o nível de experiência do usuário, oferecendo uma trilha de aprendizado personalizada.
  2. **Implementar funcionalidades** que permitam ao usuário definir metas diárias e acompanhar seu progresso por meio de gráficos e indicadores visuais.
  3. **Incorporar elementos de gamificação**, como conquistas e competições saudáveis entre amigos, para incentivar o engajamento no processo de aprendizado.
  4. **Oferecer funcionalidades** para criação de resumos personalizados e acompanhamento de certificados e recordes, promovendo uma experiência completa de aprendizado.
  5. **Desenvolver uma interface amigável e responsiva**, que permita fácil navegação em dispositivos móveis e desktops.

# JUSTIFICATIVA

A escolha do tema para este trabalho é justificada com o objetivo de atender à crescente demanda por ferramentas educacionais inovadoras e eficazes no aprendizado de programação. Nos últimos anos, a programação tornou-se uma habilidade essencial, não só para profissionais da tecnologia, mas para diversas áreas do conhecimento, devido à importância da automação e da análise de dados no mundo moderno. O ensino de programação, entretanto, pode apresentar desafios, especialmente para iniciantes que frequentemente encontram barreiras conceituais e técnicas.

Diante disso, o **LEKA** busca introduzir uma abordagem gamificada para tornar o aprendizado de programação mais atraente e acessível. Ao incorporar elementos de jogo, como metas, conquistas e interações com amigos, o aplicativo visa aumentar a motivação e o engajamento dos usuários, facilitando a construção de conhecimento de maneira lúdica e interativa. Esse tipo de abordagem é especialmente eficaz para públicos jovens e adultos que valorizam a experiência de aprendizado personalizada e que se beneficiam de uma interface intuitiva e estimulante.

Além disso, o **LEKA** se diferencia por oferecer um sistema de personalização baseado em questionários iniciais e um sistema de progresso que possibilita ao usuário visualizar sua evolução de forma clara. Essa estrutura incentiva a continuidade no aprendizado e promove um senso de conquista, ajudando o usuário a superar dificuldades comuns no aprendizado de programação. Dessa forma, o **LEKA** não só contribui para o desenvolvimento de habilidades técnicas, mas também estimula a disciplina e a autogestão dos estudos, habilidades cada vez mais valorizadas em contextos acadêmicos e profissionais.

# TAP (TERMO DE ABERTURA DO PROJETO)

**Leka** :

O aplicativo LEKA visa promover o aprendizado de habilidades de programação de forma gamificada e personalizada. Ele utiliza questionários iniciais para entender os objetivos do usuário, como razões para o aprendizado (profissional, diversão etc.) e nível de experiência. Com base nas respostas, o aplicativo sugere trajetórias de aprendizado adaptadas, estabelecendo metas diárias e acompanhando o progresso através de etapas interativas. Além disso, LEKA incentiva a competição saudável, permitindo que os usuários convidem amigos para participar. O aplicativo também oferece funcionalidades para criar resumos personalizados das aulas e acompanhar o progresso, certificados e recordes, tudo dentro de um ambiente interativo e envolvente.

**Patrocinadores:** SENAI Jaguariúna

Nossos professores, **Robson Souza** e **Wellington Fabio de Oliveira Martins** que também nos ajudarão no desenvolvimento do projeto.

|  |  |
| --- | --- |
| Desenvolvedores  RH  (Recursos Humanos) | Carla Carota Mozena  Letícia Hofman |
| Gerente do Projeto | Carla Carota Mozena |
| Patrocinador | Wellington, Robson |
| Cliente | Escola Senai Jaguariúna |
| Prazo ou data de entrega do Projeto | **04/12/2024** |

|  |
| --- |
| Local\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data: \_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_/\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Patrocinador | Ass: |
| Cliente | Ass: |
| Gerente do Projeto | Ass: |

# METODOLOGIA

A metodologia adotada para o desenvolvimento do aplicativo LEKA integra princípios do Design Thinking e práticas ágeis, visando criar uma experiência de aprendizado gamificada e adaptável. O objetivo é garantir que o sistema seja desenvolvido de forma flexível, possibilitando ajustes contínuos e respondendo de maneira eficaz às necessidades e expectativas dos usuários.

A aplicação do Design Thinking permite que a equipe compreenda profundamente os desafios e objetivos dos usuários, promovendo a criação de soluções centradas no ser humano e ajustadas ao contexto do aprendizado de programação. Aliado a isso, o uso de práticas ágeis facilita o desenvolvimento iterativo e incremental, permitindo revisões constantes e adaptação rápida a feedbacks. Assim, busca-se garantir que o aplicativo ofereça uma experiência envolvente e alinhada aos requisitos de aprendizado dos usuários, promovendo a eficácia do produto.

## Fases do Projeto:

**Planejamento:**

**Pesquisa e Definição de Requisitos:** A primeira fase envolveu uma pesquisa aprofundada para identificar as necessidades e os desafios enfrentados por aprendizes de programação, especialmente iniciantes. A partir dessas informações, foram definidos os requisitos funcionais e não funcionais, levando em conta a usabilidade, personalização e gamificação.

**Planejamento e Estruturação**: Com os requisitos definidos, a equipe estruturou o sistema em módulos que permitem a implementação gradual das funcionalidades, como geração de token, personalização do aprendizado, definição de metas e progressão por etapas. O uso de métodos ágeis como **Scrum** facilitou o planejamento das sprints, permitindo uma abordagem iterativa e incremental.

**Prototipação e Design**: A criação de protótipos de baixa e alta fidelidade permitiu testar a **interface amigável e responsiva** do aplicativo antes do desenvolvimento completo. Durante essa etapa, o foco esteve na navegabilidade e na experiência do usuário, simulando como as funcionalidades gamificadas, como competições e conquistas, poderiam ser integradas de forma eficaz.

**Desenvolvimento**: Utilizando **práticas de desenvolvimento ágil**, cada módulo foi implementado e testado de forma independente, garantindo que os componentes, como o sistema de login com token e a criação de resumos personalizados, fossem integrados de maneira coesa. O desenvolvimento seguiu um ciclo contínuo de teste e revisão, adaptando-se às necessidades dos usuários identificadas durante os testes.

**Testes e Validação**: A etapa de testes incluiu a verificação de funcionalidades em múltiplos dispositivos (desktop e mobile) para assegurar a responsividade e o funcionamento do sistema. Testes de usabilidade foram realizados para identificar áreas de melhoria, além de testes de segurança para validar o sistema de autenticação por token

**Lançamento e Feedback:** O lançamento inicial do aplicativo incluiu uma versão de teste para um grupo específico de usuários, com a coleta de feedbacks para ajustes finais. A metodologia ágil permitiu que o feedback dos usuários fosse incorporado rapidamente ao projeto, aprimorando funcionalidades e ajustando o design.

**Manutenção e Suporte**: Após o lançamento, a equipe continua a monitorar o desempenho do aplicativo e fornecer suporte aos usuários. A manutenção envolve correções de bugs, melhorias de desempenho e adição de novas funcionalidades com base no feedback contínuo dos usuários, garantindo que o LEKA permaneça atualizado e alinhado com as necessidades de seu público.

# Metodologias Ágeis:

A equipe do aplicativo Leka adotou práticas ágeis, como sprints e revisões de sprints, para promover colaboração contínua e possibilitar ajustes rápidos com base no feedback dos usuários. Cada sprint focou no desenvolvimento de funcionalidades específicas e em revisões detalhadas para garantir que os requisitos e expectativas fossem atendidos. As revisões periódicas permitiram identificar melhorias e implementar mudanças no decorrer do processo, assegurando que o aplicativo se mantivesse alinhado com o objetivo de oferecer uma experiência personalizada e envolvente de aprendizado gamificado.

* 1. Ferramentas e Tecnologias**:**

**Ferramentas de Desenvolvimento:**

**Visual Studio Code**: IDE utilizada para o desenvolvimento do código, com suporte a várias linguagens e plugins que ajudam na produtividade

**Git e GitHub**: Utilizados para controle de versão e colaboração, garantindo que todas as alterações fossem documentadas e facilitando o trabalho em equipe.

**Figma**: Ferramenta de design colaborativo para prototipagem das interfaces e validação dos conceitos visuais antes da implementação.

## Tecnologias Utilizadas:

**React Native**: Framework para o desenvolvimento de aplicativos móveis multiplataforma (iOS e Android), possibilitando um desenvolvimento ágil e uniforme para ambos os sistemas operacionais.

**Node.js e Express**: Utilizados no backend para gerenciar a lógica do servidor e as APIs RESTful, permitindo a comunicação entre o frontend e o backend.

**MySQL**: Sistema de gerenciamento de banco de dados relacional utilizado para armazenar e organizar dados dos usuários, como progresso no aprendizado e preferências, de forma estruturada e segura.

**Metodologias Ágeis (Scrum)**: A equipe utilizou práticas ágeis, como sprints e reuniões de revisão, para promover a colaboração contínua e permitir ajustes rápidos com base no feedback dos usuários.

## Documentação e Comunicação:

**Documentação Contínua:** Manutenção de documentação técnica e de usuário ao longo de todo o ciclo de vida do projeto.

**Comunicação Eficaz:** Estabelecimento de canais de comunicação claros entre os membros da equipe e para garantir alinhamento e transparência.

# Resultados Esperados

Os principais resultados esperados para o projeto LEKA incluem:

## ****Funcionalidade Completa do Sistema:****

O sistema LEKA deve estar totalmente funcional, com todas as principais características e funcionalidades implementadas e operacionais, conforme definido nos requisitos do projeto.

## ****Integração Eficiente:****

As diferentes partes do sistema (módulos) devem trabalhar juntas de forma fluida, garantindo que a comunicação e a troca de dados entre elas ocorram sem problemas.

## ****Experiência do Usuário:****

Os usuários devem encontrar uma interface intuitiva e de fácil navegação, com um fluxo de trabalho que atenda às suas necessidades de forma eficaz e facilite a interação com as funcionalidades do sistema.

## ****Desempenho e Confiabilidade:****

O sistema deve apresentar um desempenho rápido e confiável, com tempos de resposta adequados e sem falhas significativas durante a operação, assegurando que os usuários possam contar com a estabilidade do sistema.

## ****Segurança dos Dados:****

Os dados dos usuários devem ser protegidos de acordo com os padrões de segurança estabelecidos, garantindo a confidencialidade, integridade e disponibilidade das informações sensíveis.

## ****Feedback Positivo:****

A apresentação do sistema para os stakeholders deve resultar em feedback construtivo e positivo, permitindo identificar áreas para melhorias e validações adicionais antes da implantação completa.

## ****Documentação Adequada:****

Toda a documentação técnica, incluindo manuais de usuário e guias de instalação, deve estar completa e clara, facilitando a compreensão e o uso do sistema pelos usuários finais.

## ****Acessibilidade:****

O sistema deve ser projetado para ser acessível a todos os usuários, incluindo aqueles com necessidades especiais, garantindo que todos possam utilizar suas funcionalidades sem barreiras.

## ****Relatórios e Análises:****

O sistema deve ser capaz de gerar relatórios detalhados e análises que ajudem os usuários a tomarem decisões informadas, proporcionando insights valiosos sobre o uso e a eficácia do sistema.

## ****Escalabilidade:****

O design do sistema deve permitir facilmente expansão e integração de novas funcionalidades, garantindo que o sistema possa crescer e se adaptar às necessidades futuras dos usuários.

## Preparação para a Implantação:

O sistema deve estar pronto para uma futura implantação completa, com todas as melhorias e ajustes realizados com base no feedback recebido e nas necessidades identificadas durante o desenvolvimento.

# PROTÓTIPO

As imagens a seguir foram geradas utilizando a técnica de prototipagem para apresentar um esboço das telas do sistema LEKA para dispositivos móveis.

## Protótipo do Dispositivo Móvel

As imagens mostram a sequência de telas que um usuário do aplicativo móvel deve acessar para usufruir de todas as funcionalidades da aplicação.

Figura 7-1 - Tela Inicial

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-2 - Tela de Login

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-3 - Tela inicial de Cadastro

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-4 - Tela de Cadastro, após Login

Tela de um aparelho celular

Descrição gerada automaticamente com confiança média

Fonte: Feito pelo Autor, 2024.

Figura 7-5 - Tela de Opções

Tela de um aparelho celular

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-6 - Tela Nível de Experiência

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-7 - Tela de Aspecto Preferido

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-8 - Tela Área de interesse pessoal

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-9 - Tela Escolha Pessoal Realizada

Tela de celular

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-10 - Tela Tempo de Estudo

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-11 - Tela Conclusão tempo do Curso

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-12 - Tela lembrete diário

Tela de um aparelho celular

Descrição gerada automaticamente com confiança média

Fonte: Feito pelo Autor, 2024.

Figura 7-13 - Tela progresso do Aluno

Tela de celular

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-14 - Tela Início do Curso

Tela de celular

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

Figura 7-15 - Tela Amigos Online

Desenho de pessoa com relógio digital

Descrição gerada automaticamente com confiança baixa

Fonte: Feito pelo Autor, 2024.

Figura 7-16 - Tela Criar Resumos

Texto

Descrição gerada automaticamente com confiança média

Fonte: Feito pelo Autor, 2024.

Figura 7-17 - Tela Perfil do Aluno

Tela de celular

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

# EAP

A Estrutura Analítica do Projeto (EAP) é uma ferramentaria de gerenciamento de projetos que divide o trabalho em partes menores e mais gerenciáveis. A EAP é uma representação gráfica da decomposição do trabalho a ser executado em um projeto que permite a visualização de todas as entregas que o projeto deve realizar. A EAP é uma ferramenta fundamental para o planejamento e controle de projetos, pois permite a visualização de todas as entregas que o projeto deve realizar.

Este documento ficará estruturado da seguinte forma

• Análise de Riscos

• Escopo do projeto - Requisitos

• Cronograma - Gráfico GANTT

• CPM - Caminho Crítico

• Custos - Orçamento do Projeto

## Análise de Riscos – SWOT

Análise de riscos utilizando a Matriz SWOT (Strengths, Weaknesses, Opportunities, Threats), uma sigla em inglês que significa Forças, Fraquezas, Oportunidades e Ameaças. FOFA em português.

Tabela 1 - Tabela Matriz SWOT



Fonte: Feito pelo Autor, 2024.

A análise conclui que o projeto é viável pois as forças e oportunidades são maiores do que as ameaças e fraquezas.

## Escopo do Projeto

O escopo deste projeto abrange o desenvolvimento de uma aplicação Full Stack, destinada tanto para a web quanto para dispositivos móveis.

O projeto está estruturado em torno de requisitos funcionais, que definem as funcionalidades que a aplicação deve oferecer, e requisitos não funcionais, que especificam os critérios de qualidade que a aplicação deve atender, como desempenho, segurança e usabilidade.

Cada um desses aspectos será detalhado para garantir que a solução final atenda às necessidades dos usuários e cumpra com os padrões técnicos estabelecidos.

# Documento de Regras de Negócio (RN)

**Introdução**

Este documento descreve as regras de negócio que governam as funcionalidades do projeto **Leka**, uma plataforma de aprendizado gamificado focada no ensino de programação. As regras de negócio aqui apresentadas são diretrizes fundamentais que o sistema deve seguir para garantir que suas operações estejam em conformidade com os requisitos funcionais e não funcionais previamente estabelecidos. Elas determinam o comportamento do sistema, garantindo consistência, segurança, personalização da experiência do usuário e a correta execução dos processos de aprendizado, interação com amigos, progresso e recompensas. O objetivo deste documento é proporcionar uma visão clara e detalhada de como o sistema deve operar, assegurando o desenvolvimento e manutenção de funcionalidades que atendam às expectativas e necessidades dos usuários.

## Regras de Negócio

### Geração e Validação de Token

**Descrição**: O sistema deve gerar um token para o usuário após o login bem-sucedido.

1. **RN001**: O sistema deve realizar a leitura do token.
2. **RN002**: O sistema deve autenticar o usuário com base no token.
3. **RN003**: O sistema deve validar o token a cada acesso a uma funcionalidade que exija autenticação.

### Registro e Login

**Descrição**: O sistema deve permitir que o usuário faça o cadastro e o login no sistema.

1. **RN004**: O sistema deve realizar o cadastro de novos usuários.
2. **RN005**: O sistema deve validar as credenciais do usuário (e-mail e senha) no momento do login.
3. **RN006**: O sistema deve validar o token do usuário ao fazer login.

### Questionário Inicial

**Descrição**: O sistema deve apresentar um questionário inicial para personalização da experiência de aprendizado.

1. **RN007**: O sistema deve exibir o questionário ao usuário.
2. **RN008**: As respostas do questionário devem ser armazenadas no banco de dados.
3. **RN009**: Com base nas respostas, o sistema deve sugerir uma trilha de aprendizado personalizada para o usuário.

### Definição de Metas

**Descrição:** O sistema deve permitir que o usuário defina metas diárias de aprendizado.

1. **RN010**: O sistema deve exibir uma interface para que o usuário defina suas metas.
2. **RN011**: O sistema deve armazenar as metas diárias no banco de dados.
3. **RN012**: O sistema deve permitir que o usuário ative um lembrete diário com base nas metas estabelecidas.

### Progresso e Etapas

**Descrição**: O sistema deve acompanhar o progresso do usuário e apresentar as etapas interativas de aprendizado.

1. **RN013**: O sistema deve exibir as etapas de aprendizado.
2. **RN014**: O sistema deve armazenar o progresso das etapas de aprendizado.
3. **RN015**: O sistema deve atualizar o progresso conforme o usuário avança nas etapas.

### Atividades

**Descrição**: O sistema deve permitir que o usuário realize atividades relacionadas ao aprendizado.

1. **RN016**: O sistema deve exibir atividades personalizadas para o usuário.
2. **RN017**: O sistema deve registrar a conclusão das atividades.
3. **RN018**: O sistema deve atualizar o progresso do usuário após a conclusão das atividades.

### Interação com Amigos

**Descrição**: O sistema deve permitir que o usuário interaja com amigos, promovendo competição saudável.

1. **RN019**: O sistema deve permitir que o usuário convide amigos.
2. **RN020**: O sistema deve armazenar as conexões de amigos.
3. **RN021**: O sistema deve exibir o progresso dos amigos para o usuário.

### Conquistas e Recompensas

**Descrição**: O sistema deve permitir que o usuário ganhe conquistas à medida que avança no aprendizado.

1. **RN022**: O sistema deve notificar o usuário quando ele receber conquistas.
2. **RN023**: O sistema deve armazenar as conquistas do usuário.
3. **RN024**: O sistema deve exibir as conquistas na interface do usuário.

### Resumos Personalizados

**Descrição**: O sistema deve permitir que o usuário crie resumos personalizados do conteúdo estudado.

1. **RN025**: O sistema deve permitir que o usuário crie resumos.
2. **RN026**: O sistema deve armazenar os resumos criados pelo usuário.
3. **RN027**: O sistema deve permitir que o usuário visualize e edite seus resumos a qualquer momento.

### Gerenciamento de Perfil

**Descrição**: O sistema deve permitir que o usuário gerencie seu perfil pessoal.

1. **RN028**: O sistema deve permitir que o usuário visualize e edite suas informações pessoais.
2. **RN029**: O sistema deve permitir que o usuário visualize seu progresso geral.
3. **RN030**: O sistema deve exibir os certificados alcançados pelo usuário.

### Competição em Salas

**Descrição**: O sistema deve permitir que o usuário participe de competições em salas criadas por professores.

1. **RN031**: O sistema deve permitir que o professor crie salas para competições.
2. **RN032**: O sistema deve registrar e exibir o progresso individual dos alunos em cada sala.
3. **RN033**: O sistema deve permitir que o professor visualize o percentual de acertos e erros dos alunos nas atividades.

## Requisitos Funcionais (RF)

### RF001 – Gerar Token

Diagrama

Descrição gerada automaticamente

Figura 8-1 - Ilustrativo DCU Token

**Prioridade:** [X]Essencial, [ ]Importante, [ ]Desejável

**Referência:** [RN001], [RN002], [RN003]

### RF002 - Registro e Login

Diagrama

Descrição gerada automaticamente

Figura 8-2 - Ilustrativo DCU Registrar

**Prioridade:** [X]Essencial, [ ]Importante, [ ]Desejável

**Referência:** [RN004], [RN005], [RN006]

### RF003 - Questionário Inicial

Diagrama

Descrição gerada automaticamente

Figura 8-3 - Ilustrativo DCU Questionário Inicial

**Prioridade:** [X]Essencial, [ ]Importante, [ ]Desejável

**Referência:** [RN007], [RN008],[RN009]

### RF004 - Definição de Metas

Diagrama

Descrição gerada automaticamente

Figura 8-4 - Ilustrativo DCU Definição de Metas

**Prioridade:** [X]Essencial, [ ]Importante, [ ]Desejável

**Referência:** [RN010], [RN011], [RN012]

### RF005 - Progresso e Etapas

Diagrama

Descrição gerada automaticamente

Figura 8-5 - Ilustrativo DCU Progresso e Etapas

**Prioridade:** [ ]Essencial, [X]Importante, [ ]Desejável

**Referência:** [RN013], [RN014], [RN015]

### RF006 - Atividades

Diagrama

Descrição gerada automaticamente

Figura 8-6 - Ilustrativo DCU Atividades

**Prioridade:** [X]Essencial, [ ]Importante, [ ]Desejável

**Referência:** [RN16], [RN17], [RN018]

### RF007 - Interação com Amigos

Diagrama

Descrição gerada automaticamente

Figura 8-7 - Ilustrativo DCU Interação com Amigos

**Prioridade:** [ ]Essencial, [ ]Importante, [X]Desejável

**Referência:** [RN19], [RN20], [RN021]

### RF008 - Conquistas e Recompensas

Diagrama

Descrição gerada automaticamente

Figura 8-8 - Ilustrativo DCU Conquistas e Recompensas

**Prioridade:** [ ]Essencial, [ ]Importante, [X]Desejável

**Referência:** [RN22], [RN23], [RN024]

### RF009 - Resumos Personalizados

Diagrama

Descrição gerada automaticamente

Figura 8-9 - Ilustrativo DCU Resumos Personalizados

**Prioridade:** [ ]Essencial, [ ]Importante, [X]Desejável

**Referência:** [RN25], [RN26], [RN027]

### RF010 - Gerenciamento de Perfil

Diagrama

Descrição gerada automaticamente

Figura 8-10 - Ilustrativo DCU Gerenciamento de Perfil

**Prioridade:** [X]Essencial, [ ]Importante, [ ]Desejável

**Referência:** [RN28], [RN29], [RN030]

### RF011 - Competição em Salas

Diagrama

Descrição gerada automaticamente

Figura 8-11 - Ilustrativo DCU Competição em Salas

**Prioridade:** [ ]Essencial, [ ]Importante, [X]Desejável

**Referência:** [RN31], [RN32], [RN033]

## Requisitos Não Funcionais (RNF)

### Segurança

**RNF001:** O sistema deve garantir que todos os dados armazenados dos usuários, como informações pessoais e progresso de aprendizado, sejam protegidos por criptografia.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN001]

**RNF002:** Deve ser utilizada autenticação multifator (MFA) para maior segurança no processo de login.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN002]

**RNF003:** A comunicação entre cliente e servidor deve ser feita exclusivamente por meio de HTTPS (SSL/TLS) para garantir a integridade dos dados transmitidos.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN003]

### Usabilidade

**RNF004:** O sistema deve apresentar uma interface simples e intuitiva para facilitar o uso por iniciantes, sem a necessidade de instruções complexas.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN004]

**RNF005:** Deve possuir uma navegação amigável, com botões bem definidos e acessíveis, com destaque para as principais funcionalidades.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN005]

**RNF006:** O layout deve ser responsivo, funcionando corretamente em dispositivos móveis, como smartphones e tablets, bem como em desktops.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN006]

**Nota**: Mais requisitos nesta categoria serão adicionadas conforme novas necessidades forem identificadas.

### Confiabilidade

**RNF007:** O sistema deve garantir que as informações inseridas pelos usuários não sejam perdidas ou corrompidas, mesmo em casos de falha do servidor ou de hardware.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN007]

**RNF008:** Deve haver um sistema de backup automatizado para evitar a perda de dados, com frequência diária.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN008]

**RNF009:** Em caso de falha, o sistema deve ser capaz de restaurar o estado anterior em até 10 minutos.

* **Prioridade**: [X] Essencial [ ] Importante [ ] Desejável
* **Referência**: [RN009]

**Nota**: Requisitos de confiabilidade ainda não foram definidos. Esta seção será atualizada conforme o desenvolvimento do projeto.

### Escalabilidade

**RN0010:** O sistema deve ser capaz de suportar muitos usuários simultâneos sem impactar o desempenho, escalando horizontalmente ou verticalmente conforme necessário.

* **Prioridade:** [X] Essencial [ ] Importante [ ] Desejável
* **Referência:** [RN010]

**RN0011:** A arquitetura do sistema deve permitir a adição de novos módulos e funcionalidades sem necessidade de grandes alterações na estrutura existente.

* **Prioridade:** [X] Essencial [ ] Importante [ ] Desejável
* **Referência:** [RN011]

**RN0012:** O banco de dados deve ser otimizado para lidar com um grande volume de transações, suportando consultas rápidas e eficazes.

* **Prioridade:** [X] Essencial [ ] Importante [ ] Desejável
* **Referência:** [RN012]

**Nota**: Requisitos de escalabilidade ainda não foram definidos. Esta seção será atualizada conforme o desenvolvimento do projeto.

**Considerações Finais**

Este documento de requisitos não funcionais será atualizado conforme o projeto **Leka** evolua e novas necessidades sejam identificadas.

# PLANEJAMENTO (CRONOGRAMA GRÁFICO DE GANTT)

Planejamento e gestão do tempo e recursos humanos.

O backlog do projeto está dividido em atividades e segue apresentado utilizando um gráfico Gantt.

Este documento apresenta apenas uma versão intermediária do planejamento, durante a execução e controle sob a metodologia Scrum, a cada Sprint o cronograma foi atualizado.

Tabela 2 - Cronograma de Gantt

**Projeto LEKA**

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

## CPM – Caminho crítico

Para a elaboração do cronograma inicial acima foram aplicadas as ferramentas PERT e CPM.

O PERT (Program Evaluation and Review Technique) e o CPM (Critical Path Method) são técnicas de gerenciamento de projetos que ajudam a planejar, agendar e coordenar tarefas dentro de um projeto. O PERT é usado principalmente em projetos de pesquisa e desenvolvimento, enquanto o CPM é usado para projetos de construção e manufatura, em nosso caso utilizamos o CPM pois o tempo para cada tarefa já foi estimado pela equipe e acordado entre os stakeholders no TAP (Termo de Abertura do Projeto).

A revisão do tempo das atividades está ilustrada na tabela a seguir:

### PERT

Tabela 3 - Tabela PERT

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

Fonte: Feito pelo Autor, 2024.

### Caminho Crítico (CPM)

O caminho crítico traçado na rede apresentada na imagem abaixo, calculado somando as atividades de maior duração em sequência, totalizando 42 dias.

Figura 9-1 - Caminho Crítico

Gráfico

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

## Orçamento

Estimativa de Custos / Orçamento Inicial

Tabela 4 - Tabela Orçamento

|  |  |  |  |
| --- | --- | --- | --- |
| **Recurso** | **Hora** | **Valor Hora** | **Total** |
| Design – Identidade visual, Protótipo | 8 | 30,00 | 240,00 |
| Back-End + Banco de Dados | 110 | 30,00 | 3.300,00 |
| Front-end | 90 | 30,00 | 2.700,00 |
| Mobile | 200 | 30,00 | 6.000,00 |
| Integração Full-Stack | 120 | 30,00 | 3.600,00 |
| Documentação do projeto | 30 | 30,00 | 900,00 |
| Testes | 25 | 30,00 | 750,00 |
| Total |  |  | **17.490,00** |

# Documento de Arquitetura

Este documento tem como objetivo apresentar a arquitetura do sistema **LEKA**, destacando a estrutura e os componentes principais utilizados em sua construção. São descritos os elementos fundamentais da solução, visando proporcionar uma visão clara e detalhada de como o sistema foi projetado e organizado.

Os diagramas apresentados incluem:

**Diagrama de Classes**: Representa a estrutura das classes do sistema e suas relações.

**Diagrama de Atividades**: Mostra o fluxo de atividades e processos dentro do sistema.

**Diagrama MER-DER**: Modelo Entidade-Relacionamento e Diagrama de Entidade-Relacionamento que detalha as entidades do banco de dados e suas interações.

**Diagrama MER-DER (Dicionário de Dados)**: Especifica a modelagem dos dados, ilustrando como os dados são armazenados e manipulados.

Esses diagramas auxiliam no entendimento das interações entre os componentes do sistema e no planejamento da implementação e evolução do projeto.

## Diagrama de Classes

Figura 10-1 - Diagrama de Classes

Diagrama, Linha do tempo

Descrição gerada automaticamente

Fonte: Feito pelo Autor, 2024.

## Diagrama de Atividades

Placa branca com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança baixa

Figura 10-2 - Diagrama de Atividades

Fonte: Feito pelo Autor, 2024.

## MER – Der (Diagrama de Entidade Relacionamento)

Figura 10-3 - Diagrama de Entidade Relacionamento

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Fonte: feito pelo Autor, 2024.

## MER-DER (Dicionário de Dados)

Usuário = id(PK), nome VARCHAR(100), email (Unique) VARCHAR(100), data\_criacao DATETIME(3), nivel ENUM (iniciante, médio, avançado)

Progresso = id(PK), usuario\_id(KK Usuario.id) INTEGER, etapa\_atual INTEGER, metas\_diarias VARCHAR(100), progresso\_percentual INTEGER, data\_atualizacao DATETIME(3)

Questionário = id(PK), usuario\_id(FK Usuario.id), motivo\_aprendizado ENUM(profissional, diversão, trabalho,outro), categoria\_inicial ENUM(ensino\_medio, profissional, empregado), apecto\_favorito ENUM(aparencia, logico, ambos),data\_resposta DATETIME(3)

Atividade = id(PK), usuario\_id(FK Usuario.id), descricao TEXT, data\_cadastro DATETIME(3), data\_execucao DATETIME(3), etapa\_numero INTEGER

Resumo - id(PK), usuario\_id(FK Usuario.id), conteudo\_resumo TEXT, data\_criacao DATETIME(3)

Amigo = id(PK), usuario\_id(FK Usuario.id), amigo\_id INTEGER

Conquista = id(PK), usuario\_id(FK Usuario.id), descricao TEXT, data\_conquista DATETIME(3)

Sala = id(PK), professor\_id INTEGER, codigo\_sala VARCHAR, data\_criacao DATETIME(3),nome\_sala VARCHAR

Pergunta = id(PK), sala\_id INTEGER, pergunta\_text TEXT, resposta\_correta VARCHAR

Resposta Aluno = id(PK), aluno\_id INTEGER, pergunta\_id INTEGER, resposta\_aluno VARCHAR, acerto BOOLEAN

# CONCLUSÃO/ENTREGA

O aplicativo **LEKA** - Aprenda Jogando foi desenvolvido para oferecer uma plataforma interativa e educativa, focada em aprendizado de programação em níveis iniciante, intermediário e avançado. O sistema permite que os usuários adquiram habilidades em programação de maneira lúdica e envolvente, com funcionalidades como jogos educativos em diferentes categorias, acompanhamento de desempenho e sistema de recompensas, promovendo um ambiente estimulante e adaptado para o aprendizado progressivo.

Desenvolvido com tecnologias modernas como **React-native**, **Node.js** e **Expo**, **LEKA** proporciona uma experiência otimizada tanto no ambiente mobile quanto na web.

Embora o sistema esteja funcional, ele ainda passa por melhorias de usabilidade e segurança, que não estão completamente preparadas para um ambiente de produção. A apresentação do aplicativo incluirá uma avaliação para coleta de feedback, com o objetivo de aperfeiçoá-lo.

O desenvolvimento seguiu boas práticas de engenharia de software, com a realização de testes automatizados e a criação de diagramas UML para documentar a estrutura e o fluxo do sistema.

A entrega final do **LEKA** inclui:

Código-fonte completo, hospedado em um repositório versionado para revisões e validações.

Documentação técnica com instruções para configuração e implantação.

Dicionário de dados descrevendo o modelo do banco de dados.

Diagramas UML de classes, atividades e entidade-relacionamento, que oferecem uma visão clara da arquitetura.

Relatórios de testes automatizados para garantir a funcionalidade e consistência das APIs.

Manual do usuário com instruções detalhadas para uso das funcionalidades.

Com essa entrega, o **LEKA** está pronto para receber sugestões de aprimoramento, com o objetivo de se tornar um recurso robusto e seguro para ensino de programação em um ambiente de produção.

## Evidências

O **LEKA** - Aprenda Jogando documenta e demonstra o progresso e as realizações alcançadas no desenvolvimento da plataforma de aprendizado em programação. As evidências a seguir validam a implementação das funcionalidades, a conformidade com os requisitos e a qualidade geral do sistema. Abaixo estão as principais evidências fornecidas:

1. **Código-Fonte Completo**:

O código-fonte do **LEKA** está disponível em um repositório de controle de versão, como o GitHub. O repositório inclui a base de código para o Front-end, Back-end e versão Mobile, permitindo a revisão e validação técnica das implementações.

1. **Documentação Técnica**:

A documentação técnica abrange a descrição da arquitetura do sistema, os fluxos de dados e a configuração para ambientes de desenvolvimento e produção. Inclui também instruções para implantação local, facilitando a configuração do **LEKA**.

1. **Dicionário de Dados:**

Um dicionário de dados detalhado descreve a estrutura do banco de dados, auxiliando na compreensão do modelo de dados e das interações entre as diferentes entidades do sistema.

1. **Diagramas UML:**

Diagramas de classes, atividades e de entidade-relacionamento (MER) ilustram a estrutura e o fluxo do sistema LEKA. Estes diagramas proporcionam uma visão clara da arquitetura e dos processos de operação do aplicativo.

1. **Relatório de Testes:**

Testes foram realizados com o auxílio de ferramentas como Insomnia, assegurando a funcionalidade e integridade das APIs. Os relatórios dos testes comprovam o funcionamento adequado das principais funcionalidades do sistema.

1. **Apresentação e Feedback:**

Slides e documentos foram utilizados nas apresentações do projeto para docentes, com o objetivo de coleta de feedback e sugestões de melhorias. Esse retorno será essencial para ajustes e aprimoramentos do sistema, visando uma futura implantação em produção.

1. **Segurança e Criptografia:**

A documentação descreve as medidas de proteção implementadas, visando segurança dos dados do usuário. Embora a segurança do sistema ainda precise de aprimoramentos, essa seção evidencia o compromisso em proteger as informações dos usuários.

1. **Manual do Usuário:**

Um guia prático orienta os usuários finais sobre como utilizar o  **LEKA**. O manual fornece instruções detalhadas para cada funcionalidade principal, facilitando a navegação e o uso eficiente do sistema.

## Front-End (Print das Telas)

## Mobile (Print das Telas)

Figura 11-1 - Tela Inicial



Fonte: App LEKA – Início, feito pelo Autor, 2024.

Figura 11-2 - Tela de Login



Fonte: App LEKA – Login, feito pelo Autor, 2024.

Figura 11-3 - Tela de Cadastro



Fonte: App LEKA – Cadastro, feito pelo Autor, 2024.

Figura 11-4 - Tela Questionário 01



Fonte: App LEKA – Questionário 01, feito pelo Autor, 2024.

Figura 11-5 - Tela Questionário 02



Fonte: App LEKA – Questionário 02, feito pelo Autor, 2024.

Figura 11-6 - Tela Questionário 03



Fonte: App LEKA – Questionário 03, feito pelo Autor, 2024.

Figura 11-7 - Tela Questionário 04



Fonte: App LEKA – Questionário 04, feito pelo Autor, 2024.

Figura 11-8 - Tela Resultado



Fonte: App LEKA – Resultado, feito pelo Autor, 2024.

Figura 11-9 - Tela Tempo



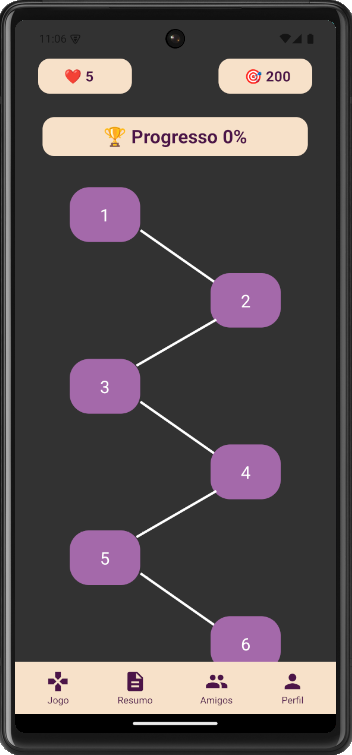
Fonte: App LEKA – Tempo, feito pelo Autor, 2024.

Figura 11-10 - Tela Gráfico Profissional



Fonte: App LEKA – Gráfico Profissional, feito pelo Autor, 2024.

Figura 11-11 - Tela Jogos



Fonte: App LEKA – Tela Jogos, feito pelo Autor, 2024.

Figura 11-12 - Tela Resumo



Fonte: App LEKA – Tela Resumo, feito pelo Autor, 2024.

Figura 11-13 - Tela Amigos (Chat)



Fonte: App LEKA – Tela Amigos (Chat), feito pelo Autor, 2024.

Figura 11-14 - Tela Comunidade



Fonte: App LEKA – Tela Comunidade, feito pelo Autor, 2024.

## Testes

**Tela de Login**

Realizar Login com e-mail e senha.

|  |
| --- |
|  |

O usuário sai da navegação e realiza o login verificando se o cadastro continua salvo.

|  |
| --- |
|  |

Se caso o usuário esquecer sua senha de acesso, poderá criar outra com base em seu e-mail cadastrado.

|  |
| --- |
|  |

Se o usuário ainda não possui uma conta, ao clicar em cadastrar ele é direcionado para uma página de cadastro.

|  |
| --- |
|  |

Após login se os dados forem compatíveis com o cadastro irá direcioná-lo para o aplicativo.

|  |
| --- |
|  |

**Tela de Cadastro:**

Criar um cadastro com base no e-mail e senha do usuário.

|  |
| --- |
|  |

Se os dados do cadastro forem compatíveis o usuário será direcionado para o login.

|  |
| --- |
|  |

# DOCUMENTO DE IMPLANTAÇÃO

**Introdução**

O presente documento de implantação do aplicativo **LEKA - Aprenda Jogando** visa fornecer um guia completo e detalhado para a configuração, instalação e execução do sistema em ambiente de produção. O LEKA é uma plataforma educativa interativa voltada ao ensino de programação, atendendo a usuários com diferentes níveis de habilidade: iniciante, intermediário e avançado. Combinando elementos lúdicos e um sistema de recompensas, o aplicativo promove uma experiência de aprendizado engajante e acessível para quem deseja desenvolver habilidades em programação.

## Tecnologias Utilizadas

O sistema **LEKA** foi desenvolvido utilizando diversas tecnologias que compõem seu **Front-end**, **Back-end**, ferramentas de teste e desenvolvimento, além das soluções adotadas para o banco de dados e a aplicação **mobile**. A seguir, são listadas as principais tecnologias utilizadas:

Tabela 5 - Tecnologias utilizadas

|  |  |
| --- | --- |
| **Tecnologias** | **Linguagens** |
| **Front-end** | JavaScript, Bootstrap, HTML, CSS |
| **Back-end** | Node.js, Express, Prisma, JWT (Json Web Token) |
| **Testes** | Insomnia |
| **Back e Front** | Visual Studio Code (VsCode) |
| **Banco de Dados** | Xampp, MariaDB e Prisma |
| **Mobile** | React-native, Yarn, Expo |

*Fonte: Feito pelo Autor, 2024.*

## Configuração do Ambiente de Desenvolvimento Web

1. Clone este repositório e abra com o VsCode:

git clone <https://github.com/Carla-coder/Leka.git>

1. Entre na pasta api:

cd api

1. Crie um arquivo .env contendo:

DATABASE\_URL="mysql://root:@localhost:3306/Leka"

KEY="base64q3

1. Abra o Xampp e inicie o mysql e o apache.
2. Inicie e instale as dependências e o banco de dados no VsCode:

npm i

npx prisma migrate dev --name leka init

1. Execute a api:

nodemon

## Instalação Mobile

Clone este repositório e abra com o VsCode:

git clone <https://github.com/Carla-coder/Leka.git>

2. Inicie o projeto:

yarn install

1. Instale as dependências:

yarn add @react-navigation/native

yarn add @react-navigation/bottom-tabs

yarn add @react-navigation/stack

yarn add @expo/vector-icons@react-native-picker/picker

yarn add @react-native-async-storage/async-storage

yarn add react-native-vector-icons/FontAwesome

yarn add validator

yarn add react-native-chart-kit

1. Execute o projeto:

yarn start

# MANUAL DO USUÁRIO

**Tutorial de Acesso ao Projeto LEKA – Aprenda Jogando**

Bem-vindo ao **LEKA**! Neste tutorial, você encontrará um guia passo a passo para aprender a acessar e utilizar o sistema para desenvolvimento de habilidades em programação. Este guia cobrirá desde o primeiro acesso até o acompanhamento de seu progresso e desempenho. Vamos começar!

## Acesso ao Sistema

**Acesso pelo Navegador**

1. Abra o navegador de sua preferência (Google Chrome, Firefox etc.).
2. Digite o seguinte endereço na barra de endereços:

http://127.0.0.1:5500/frontend/Login/login.html

1. A página de login do **LEKA** será exibida.

## Tela “Inicial”

**Visão geral:**

A primeira tela que você encontra ao abrir o aplicativo é a tela de boas-vindas. Esta tela tem como objetivo apresentar as principais funcionalidades e benefícios do aplicativo. Ela é composta por uma série de slides explicativos, cada um contendo:

* **Título:** Um resumo do aspecto abordado (ex: “Aprendizado Interativo”).
* **Subtítulo:** Uma descrição mais detalhada sobre como o LEKA tornará o aprendizado divertido e interativo.

**Descrição das Funcionalidades da Tela Inicial**

1. **Logotipo**: O logotipo do **LEKA** está centralizado na parte superior da tela para reforçar a identidade visual do aplicativo.
2. **Slides Interativos**: Você pode deslizar para a esquerda ou direita para visualizar as diferentes funcionalidades do **LEKA**, como:

* Aprendizado interativo.
* Jogos que ensinam programação de forma lúdica.
* Recompensas por completar desafios.

1. **Indicadores de Página**: Pequenos pontos na parte inferior indicam qual slide está ativo.
2. **Botão de Início**: Localizado logo abaixo dos slides, o botão “INICIAR” leva você para a tela de login do aplicativo.
3. **Opção de Login**: Um botão de texto abaixo do botão “INICIAR” permite que os usuários que já possuem uma conta acessem a tela de login diretamente.

**Como Navegar na Tela**

* **1.Deslizar entre Slides**: Arraste para a esquerda ou direita na área dos slides para alternar entre as páginas.
* **Selecionar Opções**: Toque no botão “INICIAR” para criar uma nova conta ou no botão de texto “EU JÁ TENHO UMA CONTA” para acessar a tela de login.

## Criando uma conta (Login)

**Para realizar o Login:**

1. O usuário deve preencher o campo de **Email e Nome** com as credenciais de sua conta.
2. Em seguida, deve preencher o campo de **Senha**.

* Se as credenciais estiverem corretas, o usuário será redirecionado para a tela **game**.
* Se as credenciais estiverem incorretas, será exibido um alerta informando o erro.

1. Após o cadastro, você será redirecionado automaticamente para a tela de login.

## Tela “Cadastro”

**Visão Geral**

A tela de cadastro permite que novos usuários se registrem no aplicativo. Para isso, eles devem preencher informações como nome, e-mail, celular e senha, além de confirmar a senha. Após o preenchimento e validação dos dados, o usuário pode criar sua conta.

**Funcionalidade**

**Campos de Entrada:**

**Nome**:

* 1. Campo obrigatório para o nome do usuário.
  2. Tipo: Texto.
  3. Exemplo: Leka.

**Email**:

* 1. Campo obrigatório para o e-mail do usuário.
  2. Tipo: Texto.
  3. O sistema valida se o e-mail contém "@".
  4. Exemplo: leka@email.com

**Celular**:

* 1. Campo obrigatório para o número de celular.
  2. Tipo: Texto numérico (com formatação automática).
  3. A validação exige que o número tenha 15 caracteres, incluindo DDD.
  4. Exemplo: (11) 91234-5678.

**Senha**:

* 1. Campo obrigatório para definir a senha de acesso.
  2. Tipo: Texto (mascarado).
  3. A senha deve atender aos seguintes requisitos de segurança:
     + No mínimo 6 caracteres.
     + Pelo menos 1 letra maiúscula, 1 letra minúscula, 1 número e 1 caractere especial.
     + Exemplo de senha válida: Senha@123.

**Confirmar Senha**:

* 1. Campo obrigatório para confirmação da senha.
  2. Tipo: Texto (mascarado).
  3. A senha inserida deve ser igual à senha definida no campo anterior.

**Ações:**

* **Botão "VAMOS"**:
  + Após preencher todos os campos corretamente, o usuário clica neste botão para enviar os dados e concluir o cadastro.
  + Caso algum dado esteja incorreto ou não atendendo às validações, o sistema exibirá uma mensagem de erro.
  + Se o cadastro for realizado com sucesso, o usuário será redirecionado para a tela de "Motivo".

## Tela “Motivo”

**Visão Geral**

A tela "Motivo" é apresentada após o cadastro do usuário e tem como objetivo entender o motivo pelo qual ele está aprendendo programação. O usuário pode selecionar uma das opções para prosseguir, o que ajudará a personalizar sua experiência no aplicativo.

**Funcionalidade**

**Elementos da Tela:**

**Título**: "Por que você está aprendendo?"

**Botões de Opções**:

A tela oferece 4 opções de respostas que o usuário pode selecionar. Cada botão representa um motivo para aprender programação:

* + - **Para me tornar profissional**
    - **Apenas por diversão**
    - **Para o meu trabalho atual**
    - **Por um motivo diferente**

Ao clicar em qualquer um dos botões, o usuário será direcionado para a tela **"Melhor Opção"**, independentemente da opção escolhida.

## Tela “Melhor Opção”

**Visão Geral**

A tela "**Melhor Opção**" permite que o usuário escolha qual das opções se aplica melhor a sua experiência ou preferências pessoais no contexto do aprendizado de programação. O objetivo é entender as preferências do usuário para oferecer uma experiência mais personalizada dentro do aplicativo.

**Funcionalidade**

**Elementos da Tela:**

**Título**: "Qual das opções melhor se aplica a você?"

**Botões de Opções**:

A tela oferece 3 opções para o usuário escolher, cada uma representando uma preferência sobre como ele se identifica no aprendizado de programação:

* + - **Eu gosto de resolver problemas**: Para usuários que preferem o aspecto lógico e analítico da programação.
    - **Eu gosto de criar interfaces bonitas**: Para usuários mais focados no design e estética das interfaces.
    - **Eu gosto de ambos**: Para usuários que se interessam tanto pelo lado lógico quanto pelo design.

Ao clicar em qualquer um dos botões, o usuário será direcionado para a tela **"Experiencia"**, passando o valor da opção selecionada como parâmetro de navegação.

## Tela “Experiência”

**Visão Geral**

A tela "**Experiência**" tem como objetivo perguntar ao usuário qual é o seu nível de experiência em programação. O nível escolhido ajudará a personalizar o conteúdo apresentado, de acordo com o conhecimento prévio do usuário.

**Funcionalidade**

**Elementos da Tela:**

**Título**: "Qual o seu nível de experiência?"

**Botões de Opções**:

A tela apresenta 3 opções para o usuário escolher o seu nível de experiência:

* + - **Baixo**: Para usuários iniciantes, que têm pouca ou nenhuma experiência com programação.
    - **Médio**: Para usuários que já têm algum conhecimento, mas não são experts.
    - **Alto**: Para usuários avançados, que possuem experiência significativa em programação.

Ao selecionar qualquer um dos botões, o usuário será direcionado para a tela **"Interessante"**, com o parâmetro do nível de experiência escolhido (baixo, médio ou alto) e o parâmetro da opção anterior (enviado da tela "Melhor Opção").

## Tela “Interesse”

**Visão Geral**

A tela "**Interessante**" é onde o usuário escolhe qual área da programação mais lhe interessa, com base nas opções apresentadas. Dependendo da escolha, o sistema determinará um perfil de desenvolvedor que se encaixa melhor com os interesses do usuário. Este perfil ajudará a personalizar o conteúdo e as dicas futuras dentro do aplicativo.

**Funcionalidade**

**Elementos da Tela:**

**Título**: "O que mais te interessa?"

**Botões de Opções**:

A tela apresenta 3 opções de interesse para o usuário escolher:

* + - **Eu gosto de lógica**: Para aqueles que têm interesse em resolver problemas de lógica, que se aplica mais a quem quer trabalhar com Backend.
    - **Eu gosto de design visual**: Para quem tem interesse em criar interfaces e focar no design visual, relacionado ao Frontend.
    - **Eu gosto de ambos**: Para os usuários que se interessam tanto por lógica quanto por design, com o objetivo de se tornar um desenvolvedor Fullstack.

**Ações:**

**Clicar no botão**:

Ao clicar em uma das opções, o usuário será redirecionado para a tela **"Resultado"** com o perfil de desenvolvedor calculado com base na escolha feita. O parâmetro resultado será enviado para a próxima tela, definindo o perfil que se aplica ao usuário.

## Tela “Resultado”

**Visão Geral**

A tela "**Resultado**" exibe o perfil de desenvolvedor mais adequado para o usuário, com base nas escolhas feitas nas telas anteriores. Esta tela apresenta o resultado final, mostrando a melhor opção de carreira ou área de desenvolvimento, além de fornecer um botão para seguir para a próxima etapa do aplicativo.

**Funcionalidade**

**Elementos da Tela:**

**Título**: "A melhor opção para você é:"

**Resultado**:

O **resultado** é o perfil de desenvolvedor recomendado, que pode ser:

* + - **Backend Developer** (se o usuário se interessou mais por lógica).
    - **Frontend Developer** (se o usuário se interessou por design visual).
    - **Fullstack Developer** (se o usuário escolheu ambos os interesses).

**Botão "Seguir"**:

**Texto do botão**: "Seguir"

Ao clicar no botão "Seguir", o usuário será redirecionado para a tela **"Tempo"**, onde provavelmente poderá avançar para a próxima fase do processo ou obter mais informações sobre a opção escolhida.

**Ações:**

**Clicar no botão "Seguir"**:

Ao clicar no botão, o usuário será direcionado para a tela "**Tempo**", que pode ser uma nova etapa do processo de aprendizagem ou a apresentação de um conteúdo específico relacionado à escolha do perfil de desenvolvedor.

## Tela “Tempo”

**Visão Geral**

A tela **"Tempo"** permite que o usuário selecione o tempo que deseja dedicar ao aprendizado. Dependendo da opção escolhida, o aplicativo ajustará a carga de estudo e oferecerá uma trilha personalizada com base no tempo disponível para aprendizado. O usuário poderá selecionar entre três opções de tempo e, em seguida, avançar para a tela seguinte clicando no botão "**CONTINUAR**".

**Funcionalidade**

**Elementos da Tela:**

**Título**: **"**Quanto tempo você quer gastar aprendendo?"

**Opções de Tempo**:

O usuário pode escolher entre três opções de tempo de aprendizado:

* + - **Casual**: 5 minutos.
    - **Regular**: 10 minutos.
    - **Sério**: 20 minutos.

As opções são apresentadas em botões, e a opção selecionada recebe um destaque visual, mudando a cor de fundo para indicar que foi escolhida.

Cada opção tem um texto explicativo ao lado, facilitando a compreensão do que a escolha representa.

**Botão "CONTINUAR"**:

Ao selecionar uma das opções, o usuário deve clicar no botão **"CONTINUAR"** para avançar para a próxima tela.

**Ações:**

**Selecionar uma Opção**:

O usuário pode clicar em uma das opções de tempo (**Casual**, **Regular** ou **Sério**). A opção selecionada será destacada com uma cor de fundo diferente.

**Clicar no Botão "CONTINUAR"**:

Após selecionar a opção de tempo desejada, o usuário deve clicar no botão **"CONTINUAR"** para avançar para a tela **"Trilha"**, onde o tempo selecionado será passado como parâmetro para a próxima etapa.

## Tela “Trilha”

**Visão Geral**

A tela **"Trilha"** exibe o progresso do usuário no processo de aprendizado, com base na opção de tempo selecionada anteriormente. Aqui, o usuário pode ver o tempo estimado até alcançar o nível profissional, representado visualmente por uma linha do tempo com três estágios: **Básico**, **Médio** e **Profissional**. O objetivo da tela é motivar o usuário mostrando o caminho até se tornar um profissional.

**Funcionalidade**

**Elementos da Tela:**

**Título**: **"**Você estará profissional em pouco tempo!"

**Representação Visual do Progresso**:

A linha do tempo é representada por uma linha diagonal com três pontos, cada um correspondente a um estágio do aprendizado:

* + - **Básico** (atualmente: "Agora").
    - **Médio** (aproximadamente: "1 Semana").
    - **Profissional** (tempo estimado: conforme a escolha do usuário, como "Abril de 2025", "Dezembro de 2024", ou "Julho de 2024").

Os pontos estão posicionados ao longo de uma linha diagonal que percorre a tela, criando uma sensação de progresso visual.

**Exibição do Tempo para se Tornar Profissional**:

Com base na escolha de tempo anterior (Casual, Regular ou Sério), o tempo estimado para o nível "Profissional" é calculado e exibido.

**Botão "CONTINUAR"**:

Após visualizar o progresso, o usuário pode clicar no botão **"CONTINUAR"** para avançar para a próxima tela do aplicativo, **"Game"**.

**Ações:**

**Visualizar Progresso**:

Ao acessar a tela **"Trilha"**, o usuário verá o progresso do aprendizado, representado por três estágios: **Básico**, **Médio** e **Profissional**.

O estágio **Profissional** mostrará o tempo estimado até que o usuário alcance esse nível, com base no tempo selecionado anteriormente.

**Clicar no Botão "CONTINUAR"**:

Ao clicar em **"CONTINUAR"**, o usuário será redirecionado para a tela **"Game"**. O tempo estimado para se tornar profissional (calculado com base na escolha anterior) será mantido para a próxima fase do aprendizado.

## Tela “Game”

**Visão Geral**

A tela **"Game"** exibe o progresso do usuário no jogo e oferece uma interface para avançar pelas fases do jogo. O usuário pode ver seu desempenho, acessar diferentes fases do jogo e navegar para outras seções do aplicativo, como **Resumo**, **Amigos** e **Perfil**, por meio de um menu inferior. A tela é dividida em três seções principais: **Cabeçalho** (com informações de vidas e pontuação), **Caminho do Curso** (onde o progresso nas fases é exibido), e o **Menu Inferior**.

**Funcionalidade**

1. **Cabeçalho (Vidas e Pontuação)**

* Exibe as **vidas** e a **pontuação** do usuário no topo da tela.
* **Vidas**: Representadas por um ícone de coração (❤️) com um número indicando quantas vidas o usuário tem disponíveis.
* **Pontuação**: Representada por um ícone de alvo (🎯) com a pontuação atual do usuário.

1. **Caminho do Curso (Fases do Jogo)**

Exibe as fases do jogo em um caminho visual.

**Fases**: O caminho do curso contém 10 fases, numeradas de 1 a 10. Cada fase é representada por um círculo que o usuário pode tocar para acessar o jogo correspondente.

* + O **caminho** está visualmente dividido por linhas diagonais entre as fases para representar o progresso.
  + A fase **1** é uma fase inicial de **pergunta**, e as fases subsequentes levam a diferentes jogos.
* O progresso é indicado ao clicar nas fases, com a tela navegando para a fase selecionada.

1. **Menu Inferior**

O menu inferior fornece acesso rápido a outras telas do aplicativo:

* + **Jogo**: Navega de volta para a tela principal do jogo.
  + **Resumo**: Leva o usuário para a tela de resumo do seu desempenho.
  + **Amigos**: Direciona para a tela de amigos, onde é possível interagir com outros jogadores.
  + **Perfil**: Acessa a tela de perfil do usuário, onde ele pode visualizar e editar suas informações.

**Ações:**

**1. Interagir com o Caminho do Curso**

**Clicar nas Fases**: Cada fase, representada por um círculo, pode ser tocada para avançar para o jogo correspondente.

* + A fase **1** leva o usuário à tela **Pergunta**.
  + As fases **2 a 10** levam o usuário para diferentes jogos.

**2. Visualizar Vidas e Pontuação**

As informações de **vidas** e **pontuação** são exibidas no topo da tela, permitindo ao usuário acompanhar seu progresso e desempenho no jogo.

**3. Navegar pelo Menu Inferior**

O menu inferior permite ao usuário navegar facilmente para diferentes seções do aplicativo:

* **Jogo**: Retorna à tela principal do jogo.
* **Resumo**: Acessa a tela de resumo, onde o usuário pode ver seu desempenho nas fases.
* **Amigos**: Leva à tela onde é possível interagir com outros usuários.
* **Perfil**: Permite ao usuário visualizar e editar seu perfil.

## Tela “Comunidade”

**Visão Geral**

A tela de **Comunidade** permite ao usuário criar novas comunidades, adicionar amigos, visualizar convites pendentes e gerenciar comunidades. A tela também oferece funcionalidades para visualizar amigos aceitos e interagir com eles de maneira simples e direta.

**Funcionalidades**

**1. Criar uma Comunidade**

**Objetivo:** Permite ao usuário criar uma nova comunidade.

**Passos:**

* 1. Toque no botão **"Criar Comunidade"** localizado na parte superior da tela.
  2. Preencha os campos obrigatórios:
     + **Nome da Comunidade:** Insira o nome desejado para sua comunidade.
     + **Descrição da Comunidade:** Descreva brevemente o propósito da comunidade.
  3. Selecione os amigos que deseja convidar para participar da comunidade. Toque sobre o nome de cada amigo para selecioná-lo.
  4. Após preencher as informações, toque no botão **"Criar"** para concluir a criação.
  5. A comunidade será salva e estará visível na lista de comunidades.

**2. Adicionar Amigos**

**Objetivo:** Permite ao usuário adicionar novos amigos inserindo o ID de um amigo já cadastrado.

**Passos:**

* 1. Toque no botão **"Adicionar Amigos"**.
  2. Insira o **ID do amigo** no campo de texto e toque no ícone de busca (lupa).
  3. Caso o ID seja válido e o amigo seja encontrado, um convite será enviado.
  4. O convite aparecerá na lista de **Convites Pendentes**.

**3. Visualizar Convites Pendentes**

**Objetivo:** Exibe todos os convites que o usuário recebeu para se tornar amigo de outros usuários.

**Passos:**

* 1. Ao abrir o modal de adicionar amigos, você verá a seção de **Convites Pendentes**.
  2. Nessa seção, será listado o nome de cada amigo que enviou um convite.

**4. Aceitar Convite de Amigo**

**Objetivo:** Aceitar convites pendentes de amigos.

**Passos:**

* 1. Ao visualizar um convite na lista de convites pendentes, toque sobre o convite.
  2. A amizade será confirmada e o usuário será adicionado à sua lista de amigos.

**5. Gerenciar Comunidades**

**Objetivo:** Gerenciar as comunidades criadas e deletá-las.

**Passos:**

* 1. Abaixo da seção de **Criar Comunidade**, há uma lista de comunidades já criadas.
  2. Para **entrar** em uma comunidade, toque no nome da comunidade.
  3. Para **excluir** uma comunidade, toque no ícone de lixeira ao lado da comunidade e confirme a exclusão.

**6. Visualizar Amigos Aceitos**

**Objetivo:** Exibir todos os amigos com quem o usuário tem amizade.

**Passos:**

* 1. No canto inferior direito da tela, toque no ícone de **Amigos**.
  2. Será aberta uma lista com todos os amigos aceitos, onde será possível ver o nome de cada amigo.

**Modais**

**Modal Criar Comunidade**

Usado para criar uma nova comunidade. Permite que o usuário insira o nome e a descrição da comunidade, além de selecionar participantes (amigos) que irão fazer parte da comunidade.

**Modal Adicionar Amigos**

Usado para adicionar amigos ao sistema, inserindo o ID de usuário. Exibe também os convites pendentes para aceitar amigos.

**Modal Amigos Aceitos**

Exibe a lista de amigos que foram aceitos. O usuário pode visualizar os amigos com quem já tem amizade estabelecida.

**Barra de Navegação Inferior**

A tela de **Comunidade** possui uma barra de navegação na parte inferior com os seguintes ícones:

* **Jogo:** Acessa a tela de jogo do aplicativo.
* **Resumo:** Acessa o resumo de atividades.
* **Amigos:** Acessa a tela de amigos para gerenciar amizades.
* **Perfil:** Acessa o perfil do usuário.

**Avisos e Erros**

* **Erro ao adicionar amigo:** Se o ID do amigo não for encontrado, será exibido um alerta com a mensagem "ID de amigo não encontrado".
* **Erro ao criar comunidade:** Se os campos de nome ou descrição da comunidade estiverem vazios, será exibido um alerta informando que esses campos precisam ser preenchidos.
* **Erro ao aceitar convite:** Se houver um erro ao tentar aceitar um convite, um alerta será exibido informando o erro.

## Tela “Resumo”

**Visão Geral**

A tela de **Resumo** do aplicativo permite ao usuário criar, editar, visualizar e excluir resumos de textos ou tópicos importantes. A seguir, você encontrará um guia passo a passo sobre como usar as funcionalidades dessa tela.

**Funcionalidades**

**1. Visualização de Resumos**

* Na tela inicial, você verá uma lista de resumos já criados.
* Cada item na lista exibe o **título** e a **descrição** do resumo.
* Para ver mais detalhes de um resumo, basta tocar no item desejado.

**2. Adicionar Novo Resumo**

Para adicionar um novo resumo, toque no botão **+** localizado no canto inferior direito da tela.

Um modal será exibido onde você poderá inserir as seguintes informações:

* **Título**: Nome do resumo.
* **Descrição**: Uma breve descrição do que o resumo trata.
* **Resumo**: O texto do resumo que você deseja registrar.

Após preencher os campos, você pode escolher entre:

* + **Salvar**: O resumo será salvo e aparecerá na lista.
  + **Descartar**: Cancela a criação do resumo e fecha o modal.

**3. Visualizar Detalhes do Resumo**

Ao clicar em qualquer resumo da lista, um modal será exibido mostrando:

* + O **título** do resumo.
  + O **conteúdo completo** do resumo.

Você pode fechar o modal tocando no botão **X** no canto superior direito.

**4. Editar Resumo**

* Para editar um resumo, toque no ícone de **lápis** (editar) ao lado do resumo desejado.
* O modal de adição de resumo será aberto com os campos já preenchidos com os dados atuais.
* Após editar, toque em **Salvar** para atualizar o resumo. Se preferir, pode **Descartar** as alterações.

**5. Excluir Resumo**

* Para excluir um resumo, toque no ícone de **lixeira** (excluir) ao lado do resumo desejado.
* O resumo será removido permanentemente da lista.

**6. Navegação**

Na parte inferior da tela, você tem um menu com ícones para acessar outras telas do aplicativo:

* + **Jogo**: Navega para a tela do jogo.
  + **Resumo**: Retorna à tela de resumos.
  + **Amigos**: Navega para a tela de amigos.
  + **Perfil**: Navega para a tela de perfil.

**Como Funciona o Armazenamento**

* Os resumos são armazenados localmente no dispositivo utilizando o **AsyncStorage**, garantindo que as informações persistam mesmo após o fechamento do aplicativo.
* Ao adicionar, editar ou excluir um resumo, as alterações são salvas imediatamente.

**Resumo das Ações e Atalhos**

|  |  |
| --- | --- |
| **Ação** | **Descrição** |
| **Adicionar Resumo** | Toque no botão **+** para criar um novo resumo. |
| **Visualizar Resumo** | Toque no título de um resumo para visualizar seus detalhes |
| **Editar Resumo** | Toque no ícone de lápis para editar um resumo existente. |
| **Excluír Resumo** | Toque no ícone da lixeira para excluír um resumo. |
| **Fechar Modal** | Toque no botão **X** para fechar os modais de visualização e edição |

**Conclusão**

Agora o usuário está pronto para explorar o **LEKA** e aprimorar seus conhecimentos em programação de forma interativa e divertida. O aplicativo oferece uma abordagem prática para aprender desde os conceitos mais básicos até os mais avançados, com desafios e exercícios em diferentes níveis.

Siga as etapas deste tutorial para aproveitar ao máximo todas as funcionalidades e tornar o processo de aprendizagem mais eficiente e envolvente. Lembre-se de que o progresso do usuário será continuamente monitorado e atualizado conforme avança nas atividades e nos cursos.

# REFERÊNCIAS BIBLIOGRÁFICAS

* **Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011).** From Game Design Elements to Gamefulness: Defining “Gamification.” In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9–15). ACM.
* **Hamari, J., Koivisto, J., & Sarsa, H. (2014).** Does Gamification Work? — A Literature Review of Empirical Studies on Gamification. In 2014 47th Hawaii International Conference on System Sciences (pp. 3025–3034). IEEE.
* **Brusilovsky, P., & Millán, E. (2007).** User Models for Adaptive Hypermedia and Adaptive Educational Systems. In The Adaptive Web (pp. 3–53). Springer.