

UNIVERSIDADE DE RIO VERDE (UniRV)
ENGENHARIA DE SOFTWARE

PEDRO GARCIA AUGUSTO PEREIRA
ORIENTADOR: PROF. ESP. VINICIUS FÉLIX DE PAULA

RELATÓRIO DE DESENVOLVIMENTO DE SOFTWARE
LOGIC-ENCRYPTED: JOGO EDUCACIONAL
PARA ENSINO DE LÓGICA DE PROGRAMAÇÃO

RIO VERDE – GO
2025

RESUMO

Este relatório apresenta o desenvolvimento do projeto *Logic-Encrypted*, um jogo educacional digital voltado ao ensino introdutório de lógica de programação. O trabalho parte do reconhecimento das dificuldades enfrentadas por estudantes iniciantes para compreender conceitos fundamentais como estruturas condicionais, laços de repetição e operadores lógicos, especialmente quando expostos a metodologias tradicionais centradas em teoria expositiva. A proposta busca mitigar esses desafios por meio de uma abordagem gamificada, com uso de elementos lúdicos que promovem o aprendizado ativo, o engajamento contínuo e a motivação dos alunos. O jogo foi concebido com base em princípios da aprendizagem significativa e no uso pedagógico de jogos digitais, alinhando-se às diretrizes da Base Nacional Comum Curricular (BNCC), que estabelece o pensamento computacional como uma competência essencial na formação básica. A aplicação foi desenvolvida em GameMaker Studio 2 e exportada para HTML5, permitindo sua execução direta em navegadores modernos sem necessidade de instalação, o que amplia sua acessibilidade, especialmente em ambientes educacionais com infraestrutura limitada. Entre as funcionalidades implementadas destacam-se: correção interativa de trechos de código com erros propositalmente inseridos; sistema de progressão adaptativa por fases; fornecimento de feedback imediato contextualizado; caderno virtual com glossário técnico e anotações do jogador; e salvamento automático do progresso. Além disso, o jogo oferece relatórios de desempenho que permitem o acompanhamento do aprendizado por parte de professores e alunos. O presente relatório detalha desde a concepção teórica até os aspectos técnicos da implementação, utilizando o Game Design Document (GDD) como principal estrutura metodológica de desenvolvimento. O projeto demonstra como uma solução digital, interativa e acessível pode contribuir efetivamente para o ensino de lógica de programação de forma mais dinâmica, envolvente e alinhada aos contextos reais dos estudantes brasileiros.

Palavras-chave: Jogo educacional; Lógica de programação; Gamificação; Aprendizagem ativa; HTML5.

ABSTRACT

This report presents the development of the project *Logic-Encrypted*, a digital educational game designed to support the introductory teaching of programming logic. The work is based on the recognition of the challenges faced by beginner students in understanding basic concepts such as conditionals, loops, and logical operators, particularly when taught through traditional expository methods. The proposal aims to overcome these barriers by applying gamification strategies that promote active learning, sustained engagement, and student motivation. The game was conceived according to the principles of meaningful learning and educational game design, following the guidelines of Brazil's National Common Curricular Base (BNCC), which includes computational thinking as an essential competence in early education. The application was developed in GameMaker Studio 2 and exported to HTML5, ensuring it runs directly in modern browsers without requiring installation, which enhances its accessibility, especially in environments with limited resources. Key features include: interactive correction of code snippets with intentional errors; adaptive level-based progression; immediate contextual feedback; a virtual notebook with a glossary and personal notes; and automatic saving of game progress. Additionally, the game provides performance reports that allow both students and educators to track the learning process. This report covers the entire process from theoretical conception to technical implementation, using the Game Design Document (GDD) as the main development framework. The project demonstrates how an interactive, accessible, and gamified digital solution can effectively contribute to making programming logic education more engaging, meaningful, and aligned with real student contexts.

Keywords: Educational game; Programming logic; Gamification; Active learning; HTML5.

Índice de Figuras

1	Cronograma de desenvolvimento do jogo	13
2	Diagrama de Classes – Logic-Encrypted	25
3	Diagrama Entidade-Relacionamento – Firebase para Logic-Encrypted	26
4	Diagrama de Sequência – Edição de Código	27
5	Diagrama de Sequência – Salvamento Automático do Progresso	27
6	Diagrama de Casos de Uso – Funcionalidades do Sistema	28
7	Diagrama de Componentes – Manter Cadastro (RF001)	29
8	Diagrama de Componentes – Recuperar Senha (RF002)	29
9	Diagrama de Componentes – Manter Perfil (RF003)	30
10	Diagrama de Componentes – Editar Código da Fase (RF004)	30
11	Diagrama de Componentes – Caderno Virtual (RF005)	31
12	Diagrama de Componentes – Salvamento Automático (RF006)	32
13	Diagrama de Componentes – Seleção de Fases Desbloqueadas (RF007)	33
14	Diagrama de Componentes – Feedback Imediato (RF008)	34
15	Protótipo da Tela Inicial do Jogo	35
16	Protótipo da Tela de Seleção de Fases	36
17	Protótipo da Tela do Caderno Virtual	36

Índice de Tabelas

1	RF001 – Manter cadastro	15
2	RF002 – Recuperar senha	15
3	RF003 – Manter perfil	15
4	RF004 – Editar código da fase	16
5	RF005 – Visualizar caderno virtual	16
6	RF006 – Salvar automaticamente o progresso do jogador	16
7	RF007 – Exibir interface de seleção de fases desbloqueadas	17
8	RF008 – Fornecer feedback imediato	17
9	Cadastro de Jogador	17
10	Recuperação de Senha	18
11	Edição de Perfil do Jogador	19
12	Edição de Código na Fase	20
13	Interface do Caderno Virtual	21
14	Itens Salvos Automaticamente	22
15	Interface de Seleção de Fases	22
16	Feedback Imediato	23

Sumário

1	Introdução	6
2	Game Design Document (GDD)	9
2.1	Propósito do Projeto	9
2.1.1	Regras de Negócio	9
2.1.2	Abordagem Pedagógica Inovadora	9
2.1.3	Tecnologia e Acessibilidade	10
2.1.4	Componentes Estruturais	10
2.1.5	Interessados no Projeto	10
2.2	Público-Alvo, Plataforma e Restrições	11
2.2.1	Público-Alvo Principal	11
2.2.2	Plataforma e Requisitos	11
2.2.3	Principais Restrições	11
2.2.4	Restrições Obrigatórias	12
2.3	Produção	13
2.3.1	Cronograma de Elaboração	13
2.3.2	Prioridade dos Requisitos	13
2.3.3	Requisitos Funcionais	14
2.3.4	Tabelas de Fluxo	17
2.3.5	Requisitos Não-Funcionais	23
2.4	Diagramas do Sistema	25
2.4.1	Diagrama de Classe	25
2.4.2	Diagrama Entidade-Relacionamento	26
2.4.3	Diagrama de Sequência	27
2.4.4	Diagrama de Caso de Uso	28
2.4.5	Diagrama de Sequência	29
2.5	Protótipos das Interfaces	35
2.6	Fatos e Suposições Relevantes	37
2.6.1	Fatos Relevantes	37
2.7	Escopo do Trabalho	38
	REFERÊNCIAS	40

1 Introdução

Com o crescente papel da tecnologia na sociedade, o ensino de lógica de programação tornou-se um pilar essencial para a formação de estudantes em diversas áreas do conhecimento. Essa habilidade constitui a base para o pensamento computacional, sendo considerada fundamental pela Base Nacional Comum Curricular (BNCC), que propõe o desenvolvimento de competências voltadas à resolução de problemas e ao raciocínio lógico desde os primeiros anos da educação básica (BRASIL, 2018).

Apesar de sua importância, o ensino de lógica de programação ainda enfrenta desafios significativos. Muitos estudantes iniciantes demonstram dificuldades em compreender conceitos abstratos, como variáveis, estruturas condicionais e repetições. Essa situação é agravada pelo uso de metodologias tradicionais centradas na exposição teórica e na resolução de exercícios formais, os quais nem sempre despertam o interesse e a motivação dos aprendizes (RODRIGUES et al., 2019). Como consequência, observa-se um alto índice de evasão e desmotivação em disciplinas introdutórias da área de computação.

Nesse contexto, surgem estratégias pedagógicas alternativas capazes de transformar a experiência de aprendizado em algo mais envolvente e significativo. Dentre essas estratégias, destaca-se a gamificação, definida como o uso de elementos de jogos em contextos não lúdicos, com o objetivo de potencializar o engajamento, a motivação e a aprendizagem (MCGONIGAL, 2012). Recompensas simbólicas, feedback imediato, desafios progressivos e narrativas interativas são recursos comumente utilizados em jogos para captar a atenção dos jogadores, e sua aplicação na educação tem gerado resultados positivos, conforme apontado por Shell (2017).

Diversas iniciativas vêm explorando o uso de jogos digitais como ferramenta de apoio ao ensino de programação, como "Human Resource Machine" e "CodeCombat". No entanto, essas abordagens geralmente não consideram aspectos curriculares locais nem oferecem suporte em português, o que pode representar uma barreira para sua adoção no contexto brasileiro. O jogo "Logic-Encrypted", proposto neste trabalho, busca preencher essa lacuna ao oferecer uma solução educacional gratuita, acessível via navegador, desenvolvida com base nas diretrizes da BNCC e na realidade dos estudantes brasileiros.

A principal contribuição deste trabalho consiste no desenvolvimento de um jogo educacional que utiliza os princípios da gamificação para facilitar a compreensão de conceitos funda-

mentais da lógica de programação. Por meio de uma proposta interativa e acessível, pretende-se oferecer uma alternativa didática que una teoria e prática em um ambiente digital lúdico e imersivo.

Dessa forma, parte-se da hipótese de que o uso de um jogo digital baseado em elementos de gamificação pode contribuir de maneira significativa para o engajamento e a aprendizagem de estudantes iniciantes em lógica de programação, ao apresentar os conteúdos de forma prática, visual e interativa. O propósito deste projeto, portanto, é investigar e demonstrar como uma solução lúdica, acessível e alinhada às diretrizes educacionais brasileiras pode servir como ferramenta pedagógica complementar, promovendo não apenas o entendimento conceitual, mas também o interesse contínuo dos alunos pela área da computação.

O objetivo geral deste trabalho é desenvolver um jogo educacional em 2D, denominado "Logic-Encrypted", com o intuito de promover o ensino de lógica de programação para iniciantes. O jogo deve proporcionar uma experiência interativa, acessível via navegador, que estimule o raciocínio lógico, a resolução de problemas e a correção de códigos com erros propositais, utilizando elementos de gamificação como narrativa, progressão por fases e feedback imediato.

Para atingir esse objetivo, busca-se identificar os principais desafios enfrentados por iniciantes no processo de aprendizagem da lógica de programação; planejar e estruturar um jogo com base em requisitos funcionais e não-funcionais bem definidos; desenvolver uma interface amigável e responsiva, acessível via navegador; implementar uma abordagem pedagógica ativa com feedback contextual.

Como resultado principal deste trabalho, espera-se a entrega de um jogo educacional funcional, acessível via navegador e estruturado em fases progressivas que abordem os conceitos fundamentais da lógica de programação, como estruturas condicionais, laços de repetição, operadores lógicos e variáveis. Cada fase será apresentada como um desafio interativo que exigirá do jogador a identificação e correção de trechos de código com erros propositais, promovendo, assim, a aplicação prática do conhecimento adquirido. Essa abordagem busca tornar o processo de aprendizado mais leve e lúdico, contribuindo para o engajamento e a retenção dos conteúdos.

O projeto também inclui a integração de um caderno virtual, que servirá como um repositório dinâmico de conceitos estudados, exemplos práticos e anotações personalizadas feitas pelo próprio jogador. Essa ferramenta permitirá a revisão do conteúdo em qualquer momento,

reforçando o aprendizado de forma contextual e contínua, mesmo fora do ambiente de jogo.

Além disso, será implementado um sistema de relatório de desempenho, que possibilitará o acompanhamento do progresso do jogador por meio de dados como tempo de resolução, fases concluídas e tópicos dominados. Com isso, espera-se não apenas fomentar a autorreflexão e a autogestão do aprendizado, mas também fornecer subsídios para que educadores possam utilizar o jogo como uma ferramenta de apoio didático em sala de aula ou em ambientes de ensino a distância.

2 Game Design Document (GDD)

2.1 Propósito do Projeto

O projeto *Logic-Encrypted* surge como resposta a dois desafios contemporâneos: (1) a crescente demanda por capacitação em pensamento computacional e (2) as barreiras de engajamento nos métodos tradicionais de ensino de programação. Seu objetivo central é desenvolver um jogo educacional imersivo que traduza conceitos abstratos de lógica de programação em desafios tangíveis, utilizando mecânicas de puzzle em um ambiente 2D top-down tematizado como uma "cripta digital".

2.1.1 Regras de Negócio

As regras de negócio são restrições lógicas e operacionais que devem ser aplicadas ao sistema para garantir sua coerência com os objetivos pedagógicos e funcionais definidos. No caso do *Logic-Encrypted*, destacam-se:

- O jogador só poderá avançar para a próxima fase após corrigir corretamente os erros do código apresentado na fase atual.
- O sistema deve fornecer feedback imediato (positivo ou negativo) a cada tentativa de correção feita pelo jogador.
- O conteúdo pedagógico (conceitos, exemplos e explicações) será disponibilizado no caderno virtual de forma progressiva, conforme o jogador avança nas fases.
- Não será permitido o acesso a fases ainda não desbloqueadas.

2.1.2 Abordagem Pedagógica Inovadora

A proposta diferencia-se por três pilares fundamentais:

- **Aprendizado Ativo:** O jogador atua como "decifrador" de códigos-base com erros propositalmente inseridos, desenvolvendo habilidades de lógica de programação e raciocínio lógico.
- **Progressão Adaptativa:** A estrutura não-linear permite que jogadores com ritmos diferentes avancem mediante domínio dos conceitos, não por sequência fixa.

- **Feedback Contextual:** Cada solução é acompanhada de explicações integradas ao enredo, reforçando a correlação entre teoria e prática.

2.1.3 Tecnologia e Acessibilidade

Desenvolvido no *GameMaker Studio 2*, o jogo emprega um sistema baseado em *grid* com tilesets em pixel art, otimizado para exportação em HTML5. Essa escolha garante:

- **Universalidade:** Execução direta em navegadores sem requisitos de hardware avançado.
- **Integração Educacional:** Facilidade de adoção em laboratórios escolares ou ambientes EAD.

2.1.4 Componentes Estruturais

Para maximizar a eficácia educacional, o projeto incorpora:

- **Caderno Virtual:** Funciona como um *codex* dinâmico, registrando conceitos aprendidos e permitindo revisão contextual. Inclui:
 - Exemplos de aplicação prática.
 - Glossário de termos técnicos.
 - Anotações personalizáveis pelo jogador.
- **Sistema de Dicas Hierárquicas:** Oferece auxílio escalonável (de dicas conceituais a soluções passo a passo), preservando a autonomia do aprendiz.

2.1.5 Interessados no Projeto

O projeto *Logic-Encrypted* foi concebido para atender principalmente estudantes iniciantes no aprendizado de lógica de programação, especialmente aqueles matriculados em cursos técnicos e de graduação na área de computação. Além deles, o jogo também é voltado a professores e educadores que buscam metodologias alternativas e interativas para apoiar o ensino da disciplina, tornando suas aulas mais dinâmicas e atrativas.

Outro grupo de interesse contempla instituições de ensino que pretendem adotar ferramentas digitais compatíveis com os princípios da Base Nacional Comum Curricular (BNCC), oferecendo experiências significativas de aprendizagem com base em tecnologias acessíveis. Por fim, desenvolvedores independentes e pesquisadores da área de jogos educacionais também podem se beneficiar do projeto, como referência de aplicação prática de gamificação no contexto pedagógico.

2.2 Público-Alvo, Plataforma e Restrições

2.2.1 Público-Alvo Principal

O jogo **Logic-Encrypted** será desenvolvido especialmente para iniciantes em programação que ainda não possuem conhecimento prévio, oferecendo uma introdução amigável e interativa aos conceitos fundamentais da lógica computacional. Além disso, é voltado para estudantes de cursos técnicos e de graduação na área de computação, servindo como ferramenta complementar ao conteúdo teórico. Professores também compõem o público-alvo, já que o jogo pode ser utilizado como um recurso didático inovador para dinamizar o processo de ensino.

2.2.2 Plataforma e Requisitos

A plataforma de execução do jogo é inteiramente baseada em navegadores web modernos, por meio da tecnologia HTML5. Isso garante ampla compatibilidade com sistemas operacionais como Windows, macOS, Linux e até mesmo dispositivos baseados em navegador, como os Chromebooks. O jogo não requer instalação, sendo executado diretamente do navegador com desempenho satisfatório em máquinas com configurações básicas. Para uma melhor experiência, recomenda-se o uso de navegadores atualizados como Google Chrome ou Mozilla Firefox, com conexão à internet apenas para o carregamento inicial.

2.2.3 Principais Restrições

Por se tratar de uma aplicação web focada em ambientes desktop, o jogo não possui versão adaptada para dispositivos móveis. Sua interface e jogabilidade foram otimizadas para uso em desktops e notebooks, onde os controles são realizados exclusivamente por teclado,

utilizando as setas direcionais e uma tecla de ação principal. Isso limita o acesso por dispositivos com entrada apenas por toque, como smartphones e tablets.

2.2.4 Restrições Obrigatórias

Desde a concepção do projeto, algumas restrições foram definidas como obrigatórias e inegociáveis para garantir a viabilidade técnica, pedagógica e institucional da solução. Entre elas:

- **Execução exclusivamente em navegador web:** O jogo deve funcionar integralmente sem a necessidade de instalação ou dependência de servidores externos, garantindo acessibilidade mesmo em ambientes com infraestrutura limitada.
- **Foco pedagógico:** Todas as mecânicas e conteúdos devem ser pensados com fins educativos. Recursos visuais, trilha sonora e narrativa devem sempre apoiar o objetivo de ensino de lógica de programação.
- **Compatibilidade com ambientes escolares:** A aplicação deve ser compatível com computadores de laboratórios escolares básicos, evitando tecnologias que exijam alto desempenho gráfico ou hardware avançado.

2.3 Produção

2.3.1 Cronograma de Elaboração

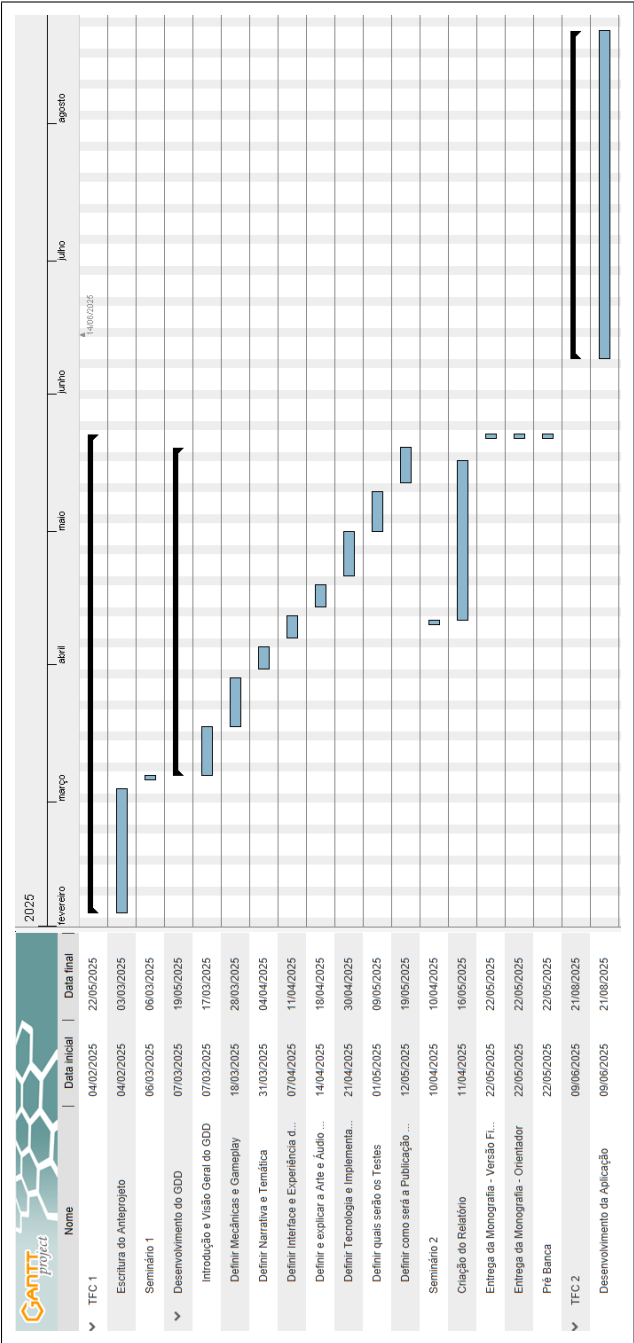


Figura 1: Cronograma de desenvolvimento do jogo

2.3.2 Prioridade dos Requisitos

A priorização dos requisitos é fundamental para guiar o desenvolvimento do sistema, garantindo que as funcionalidades mais críticas sejam implementadas primeiro. A seguir, estão

as categorias de prioridade utilizadas:

- **Essencial:** Requisitos indispensáveis para o funcionamento mínimo do sistema. Sem eles, o sistema não atende aos seus objetivos principais.
- **Importante:** Requisitos que agregam valor significativo ao sistema, mas cuja ausência não compromete o funcionamento básico. São implementados após os essenciais.
- **Desejável:** Requisitos que proporcionam melhorias ou funcionalidades adicionais. Podem ser incluídos caso haja tempo ou recursos disponíveis.

2.3.3 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, ou seja, as funcionalidades essenciais para o funcionamento do jogo e para o alcance dos objetivos educacionais propostos. No caso do *Logic-Encrypted*, os principais requisitos funcionais são:

- Manter Cadastro;
- Recuperar de Senha;
- Manter Perfil;
- Editar Código da fase;
- Visualizar caderno virtual;
- Salvar automaticamente o progresso do jogador;
- Exibir interface de seleção de fases desbloqueadas;
- Fornecer feedback imediato;

Tabela 1: RF001 – Manter cadastro

Requisito n°	RF001 – Manter cadastro
Descrição	Permitir registrar novos usuários informando nome, e-mail e senha, além de gerenciar os dados da conta.
Prioridade: Essencial	Dependências: Nenhuma Conflitos: Nenhum
Material de Apoio:	Diagrama de casos de uso

Fonte: O autor (2025)

Tabela 2: RF002 – Recuperar senha

Requisito n°	RF002 – Recuperar senha
Descrição	Permitir que o usuário recupere a senha por meio de e-mail.
Prioridade: Importante	Dependências: RF001 Conflitos: Nenhum
Material de Apoio:	Fluxo de autenticação

Fonte: O autor (2025)

Tabela 3: RF003 – Manter perfil

Requisito n°	RF003 – Manter perfil
Descrição	Permitir ao usuário visualizar e editar informações pessoais do perfil, como nome, foto e preferências.
Prioridade: Importante	Dependências: RF001 Conflitos: Nenhum
Material de Apoio:	Protótipo da tela de perfil

Fonte: O autor (2025)

Tabela 4: RF004 – Editar código da fase

Requisito n°	RF004 – Editar código da fase para desbloquear a próxima fase
Descrição	Permitir ao jogador corrigir trechos de código com erros para completar a fase e desbloquear a próxima.
Prioridade: Essencial	Dependências: RF001 Conflitos: Nenhum
Material de Apoio:	Protótipo da interface do puzzle

Fonte: O autor (2025)

Tabela 5: RF005 – Visualizar caderno virtual

Requisito n°	RF005 – Visualizar caderno virtual
Descrição	Armazenar e exibir os conceitos já aprendidos pelo jogador, permitindo revisões posteriores.
Prioridade: Essencial	Dependências: RF004 Conflitos: Nenhum
Material de Apoio:	Esboço do caderno virtual

Fonte: O autor (2025)

Tabela 6: RF006 – Salvar automaticamente o progresso do jogador

Requisito n°	RF006 – Salvar automaticamente o progresso do jogador
Descrição	Salvar automaticamente o progresso do jogador.
Prioridade: Essencial	Dependências: RF006 Conflitos: Nenhum
Material de Apoio:	Diagrama de sequência

Fonte: O autor (2025)

Tabela 7: RF007 – Exibir interface de seleção de fases desbloqueadas

Requisito n°	RF007 – Exibir interface de seleção de fases desbloqueadas
Descrição	Apresentar uma tela onde o jogador possa visualizar e acessar as fases desbloqueadas.
Prioridade: Essencial	Dependências: RF007 Conflitos: Nenhum
Material de Apoio:	Protótipo da tela de seleção

Fonte: O autor (2025)

Tabela 8: RF008 – Fornecer feedback imediato

Requisito n°	RF008 – Fornecer feedback imediato
Descrição	Após a submissão de código pelo jogador, o sistema deve indicar de forma visual e sonora se houve sucesso ou falha.
Prioridade: Essencial	Dependências: RF005 Conflitos: Nenhum
Material de Apoio:	Protótipo do editor com validação

Fonte: O autor (2025)

2.3.4 Tabelas de Fluxo

Fluxo – RF001: Manter Cadastro

Fluxo Básico

1. O jogador acessa a tela de cadastro por meio da interface inicial do jogo.
2. A tela exibe os campos para preenchimento, conforme o Quadro 1:

Tabela 9: Cadastro de Jogador

Campo	Domínio de Valores	Tipo de Campo na Tela	Obrigatório?
Nome	Texto livre	Campo texto editável	Sim
E-mail	Formato de e-mail válido	Campo texto editável	Sim
Senha	Alfanumérico com mínimo de 6 caracteres	Campo texto editável (senha)	Sim
Confirmar Senha	Deve coincidir com o campo Senha	Campo texto editável (senha)	Sim
Registrar	-	Botão de ação	-

Fonte: O autor (2025)

3. O jogador preenche os campos obrigatórios, aceita os termos e clica no botão "Registrar".
4. O sistema valida os dados informados.
5. Se os dados forem válidos e o e-mail ainda não estiver cadastrado:
 - (a) O sistema salva o novo jogador no banco de dados.
 - (b) O jogador é redirecionado automaticamente para a tela de login.
6. O fluxo termina após o redirecionamento para a tela de login.

Fluxo – RF002: Recuperar Senha

Fluxo Básico

1. O jogador acessa a opção "Esqueci minha senha" na tela de login.
2. O sistema exibe os campos para recuperação de senha, conforme o Quadro 2:

Tabela 10: Recuperação de Senha

Campo	Domínio de Valores	Tipo de Campo na Tela	Obrigatório?
E-mail	Formato de e-mail válido cadastrado	Campo texto editável	Sim
Recuperar Senha	-	Botão de ação	-

Fonte: O autor (2025)

3. O jogador informa o e-mail previamente cadastrado e clica no botão "Recuperar Senha".
4. O sistema verifica se o e-mail está vinculado a uma conta existente.
5. Se o e-mail for válido:
 - (a) O sistema envia um e-mail com link para redefinição de senha.
 - (b) O jogador é informado de que deve acessar seu e-mail para prosseguir.
6. O fluxo termina após a confirmação do envio do e-mail de recuperação.

Fluxo – RF003: Manter Perfil

Fluxo Básico

1. O jogador acessa o menu de configurações ou o ícone de perfil dentro do jogo.
2. A tela de edição de perfil é exibida com os campos preenchidos com os dados atuais, conforme o Quadro 3:

Tabela 11: Edição de Perfil do Jogador

Campo	Domínio de Valores	Tipo de Campo na Tela	Obrigatório?
Nome	Texto livre	Campo texto editável	Sim
E-mail	Formato de e-mail válido	Campo texto não editável	Sim
Senha Atual	Alfanumérico	Campo texto (senha)	Sim
Nova Senha	Alfanumérico com mínimo de 6 caracteres	Campo texto (senha)	Não
Confirmar Nova Senha	Igual ao campo Nova Senha	Campo texto (senha)	Não
Salvar Alterações	-	Botão de ação	-

Fonte: O autor (2025)

3. O jogador altera os campos desejados e clica no botão "Salvar Alterações".
4. O sistema valida as informações fornecidas.
5. Se os dados forem válidos:
 - (a) As alterações são salvas no banco de dados.
 - (b) O sistema exibe uma mensagem de confirmação de sucesso.
6. O fluxo termina com o retorno automático à tela anterior ou com a atualização dos dados exibidos.

Fluxo – RF004: Editar Código da Fase

Fluxo Básico

1. O jogador acessa uma fase disponível por meio da tela de seleção.
2. A interface da fase é carregada com um trecho de código contendo erros propositais, conforme o Quadro 4:

Tabela 12: Edição de Código na Fase

Campo	Domínio de Valores	Tipo de Campo na Tela	Obrigatório?
Código Base	Código com erro	Editor de código	Sim
Botão "Executar"	-	Botão de ação	Sim
Botão "Validar"	-	Botão de ação	Sim
Mensagem de Feedback	Texto automático	Campo de leitura	Não
Avançar Fase	Condicionado à correção correta	Botão de ação	Não

Fonte: O autor (2025)

3. O jogador analisa o código e realiza as correções no editor.
4. Ao clicar em "Executar", o sistema simula a execução e exibe uma saída visual.
5. O jogador clica em "Validar".
6. O sistema verifica se a correção está correta:
 - (a) Se estiver correta, é exibida uma mensagem positiva com explicação do erro corrigido.
 - (b) A fase é marcada como concluída e a próxima fase é desbloqueada.
 - (c) O botão "Avançar Fase" é ativado.
7. O fluxo termina com a transição para a próxima fase ou retorno à seleção de fases.

Fluxo – RF005: Módulo de Caderno Virtual

Fluxo Básico

1. O jogador acessa o caderno virtual a partir do menu principal ou durante uma fase.
2. A interface do caderno é exibida com abas de navegação, conforme o Quadro 5:

Tabela 13: Interface do Caderno Virtual

Campo	Domínio de Valores	Tipo de Campo na Tela	Obrigatório?
Lista de Conceitos	Textos previamente salvos	Campo de leitura com navegação por tópicos	Não
Anotações do Jogador	Texto livre	Campo texto editável	Não
Botão "Salvar Anotação"	-	Botão de ação	Não
Glossário Técnico	Termos e definições	Campo de leitura por ordem alfabética	Não
Voltar	-	Botão de ação	Sim

Fonte: O autor (2025)

3. O jogador pode revisar os conceitos desbloqueados e ler seus conteúdos.
4. O jogador pode escrever anotações pessoais relacionadas ao conteúdo estudado.
5. Ao clicar em "Salvar Anotação", o sistema armazena o texto no histórico pessoal do jogador.
6. O jogador pode consultar o glossário técnico a qualquer momento.
7. O fluxo termina quando o jogador clica em "Voltar" para retornar ao jogo ou menu anterior.

Fluxo – RF006: Salvamento Automático do Progresso do Jogador

Fluxo Básico

1. O jogador interage com o jogo normalmente, como por exemplo: completa uma fase, acessa o caderno virtual ou altera configurações do perfil.
2. A cada ação relevante, o sistema identifica automaticamente se há dados que precisam ser salvos.
3. Se houver alterações, o sistema executa o salvamento automático no armazenamento local do navegador.
4. O salvamento abrange: fase atual, fases desbloqueadas, anotações no caderno virtual e configurações do jogador.
5. O sistema exibe um ícone ou mensagem sutil confirmando que os dados foram salvos com sucesso.

6. O fluxo se repete periodicamente sempre que mudanças são detectadas.

Tabela 14: Itens Salvos Automaticamente

Item	Tipo de Informação	Momento de Salvamento
Fase atual	Identificador da fase em andamento	Ao iniciar uma nova fase
Fases desbloqueadas	Lista de identificadores de fases liberadas	Após finalização de fase
Progresso no caderno	Conceitos lidos, anotações inseridas	Após edição ou leitura
Configurações de usuário	Preferências visuais e idioma	Após alteração de configurações

Fonte: O autor (2025)

Fluxo – RF007: Interface de Seleção de Fases Desbloqueadas

Fluxo Básico

1. O jogador acessa o menu principal do jogo.
2. O jogador seleciona a opção "Selecionar Fase".
3. O sistema exibe uma grade com todas as fases disponíveis, conforme o Quadro 7:

Tabela 15: Interface de Seleção de Fases

Elemento	Descrição	Estado	Ação
Fase (ícone)	Representação visual da fase	Desbloqueada / Bloqueada	Clique inicia a fase (se desbloqueada)
Título da fase	Nome da fase (ex: Introdução às Variáveis)	Sempre visível	Identificação informativa
Indicador de progresso	Estrela ou porcentagem	Se fase foi completada	Exibe desempenho do jogador
Botão "Voltar"	Retorna ao menu principal	Ativo	Fecha seleção de fases

Fonte: O autor (2025)

4. O jogador visualiza quais fases estão desbloqueadas ou bloqueadas.
5. Ao clicar em uma fase desbloqueada, o jogo inicia diretamente nessa fase.
6. Se o jogador clicar em uma fase bloqueada, o sistema exibe uma mensagem informando os pré-requisitos para desbloqueio.
7. O fluxo termina com o carregamento da fase ou o retorno ao menu principal.

Fluxo – RF008: Fornecer Feedback Imediato

Fluxo Básico

1. O jogador acessa uma fase e edita o trecho de código proposto.
2. Após concluir a edição, o jogador clica no botão "Executar Código".
3. O sistema valida a sintaxe e a lógica do código inserido.
4. Com base na validação, o sistema apresenta o feedback, conforme o Quadro 8:

Tabela 16: Feedback Imediato

Cenário	Feedback Visual	Feedback Sonoro
Código correto	Mensagem de sucesso em verde; animação de desbloqueio	Som de conquista suave
Código incorreto	Mensagem de erro em vermelho; destaque em trechos inválidos	Som breve de alerta
Erro de sintaxe	Destaque em vermelho nos erros; dica textual exibida	Som de erro suave

Fonte: O autor (2025)

5. Se o código estiver correto:
 - (a) O sistema registra o progresso e desbloqueia a próxima fase.
 - (b) O jogador pode avançar ou revisar a solução.
6. Se o código estiver incorreto:
 - (a) O sistema apresenta o erro e permite nova tentativa.
 - (b) O jogador pode editar novamente e reenviar.
7. O fluxo termina após o jogador corrigir e obter feedback positivo ou abandonar a fase.

2.3.5 Requisitos Não-Funcionais

Os requisitos não funcionais definem restrições e qualidades desejadas do sistema, como desempenho, compatibilidade e usabilidade. Para o *Logic-Encrypted*, os requisitos não funcionais estabelecem parâmetros técnicos e de acessibilidade importantes para a experiência do usuário:

- Interface responsiva para diferentes resoluções de tela;
- Tempo de carregamento entre cenas inferior a 2 segundos;
- Compatibilidade com os principais navegadores modernos;
- Acessibilidade básica (ajuste de fonte e esquemas de cor);
- Desempenho estável em máquinas com configuração básica.
- Acompanhamento do progresso e desempenho do jogador.

2.4 Diagramas do Sistema

2.4.1 Diagrama de Classe



Figura 2: Diagrama de Classes – Logic-Encrypted

2.4.2 Diagrama Entidade-Relacionamento

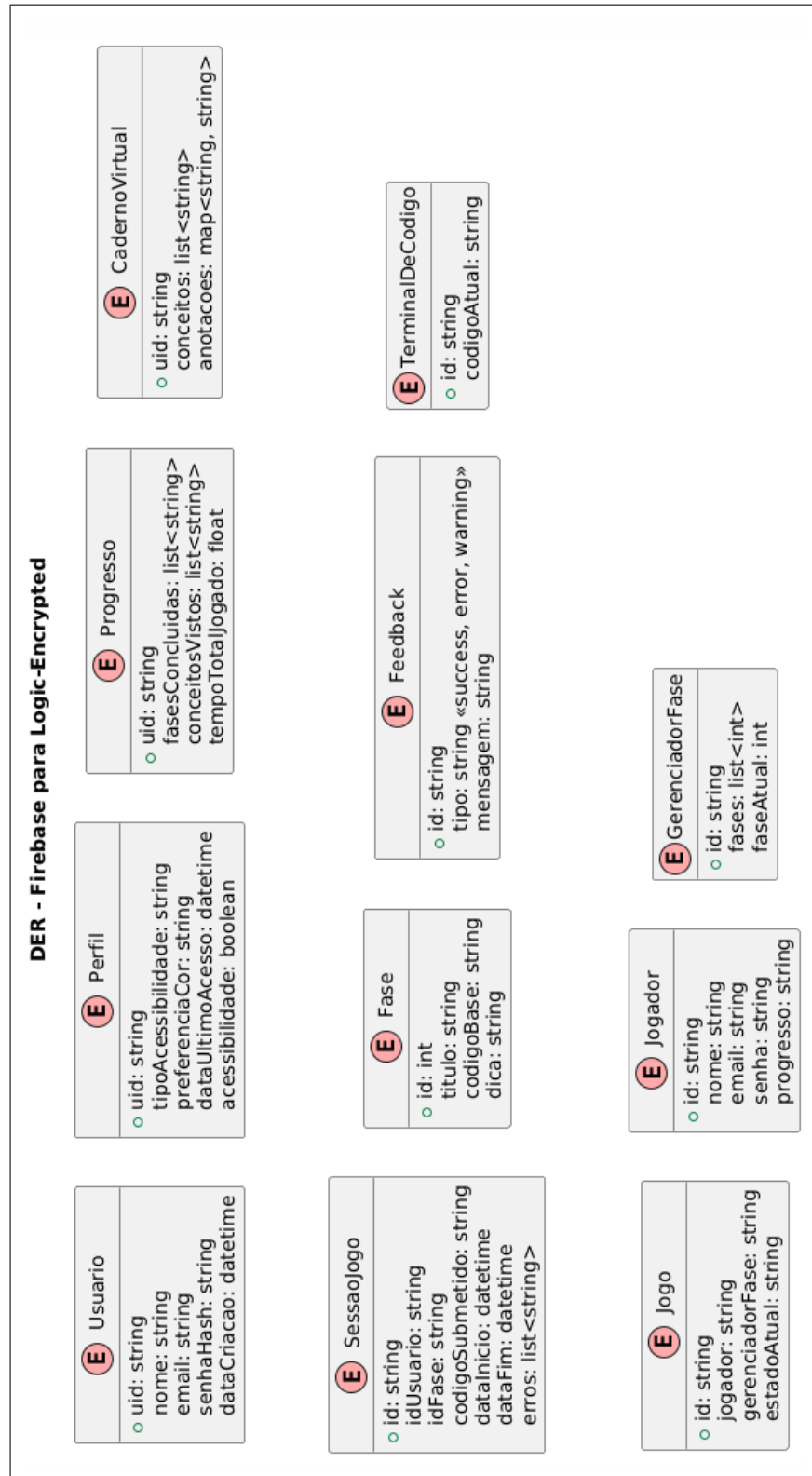


Figura 3: Diagrama Entidade-Relacionamento – Firebase para Logic-Encrypted

2.4.3 Diagrama de Sequência

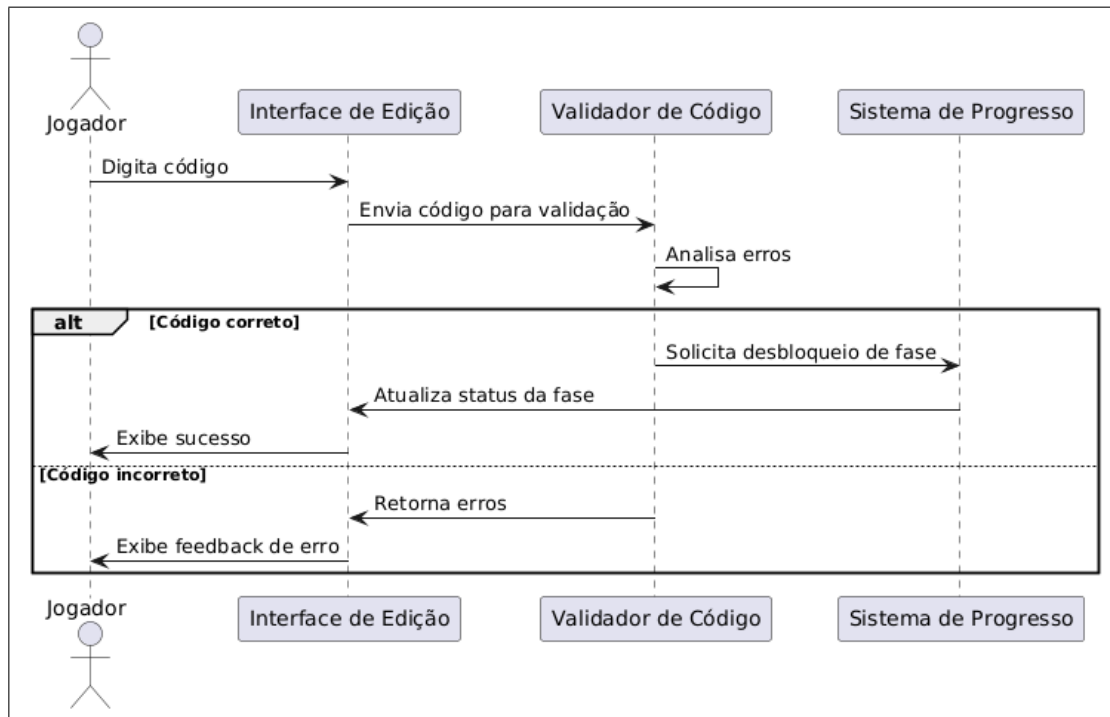


Figura 4: Diagrama de Sequência – Edição de Código

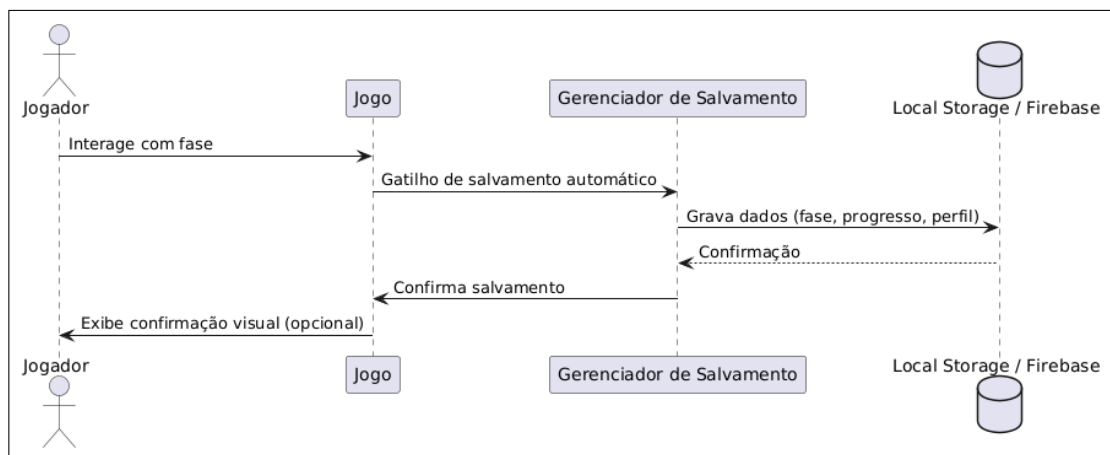


Figura 5: Diagrama de Sequência – Salvamento Automático do Progresso

2.4.4 Diagrama de Caso de Uso

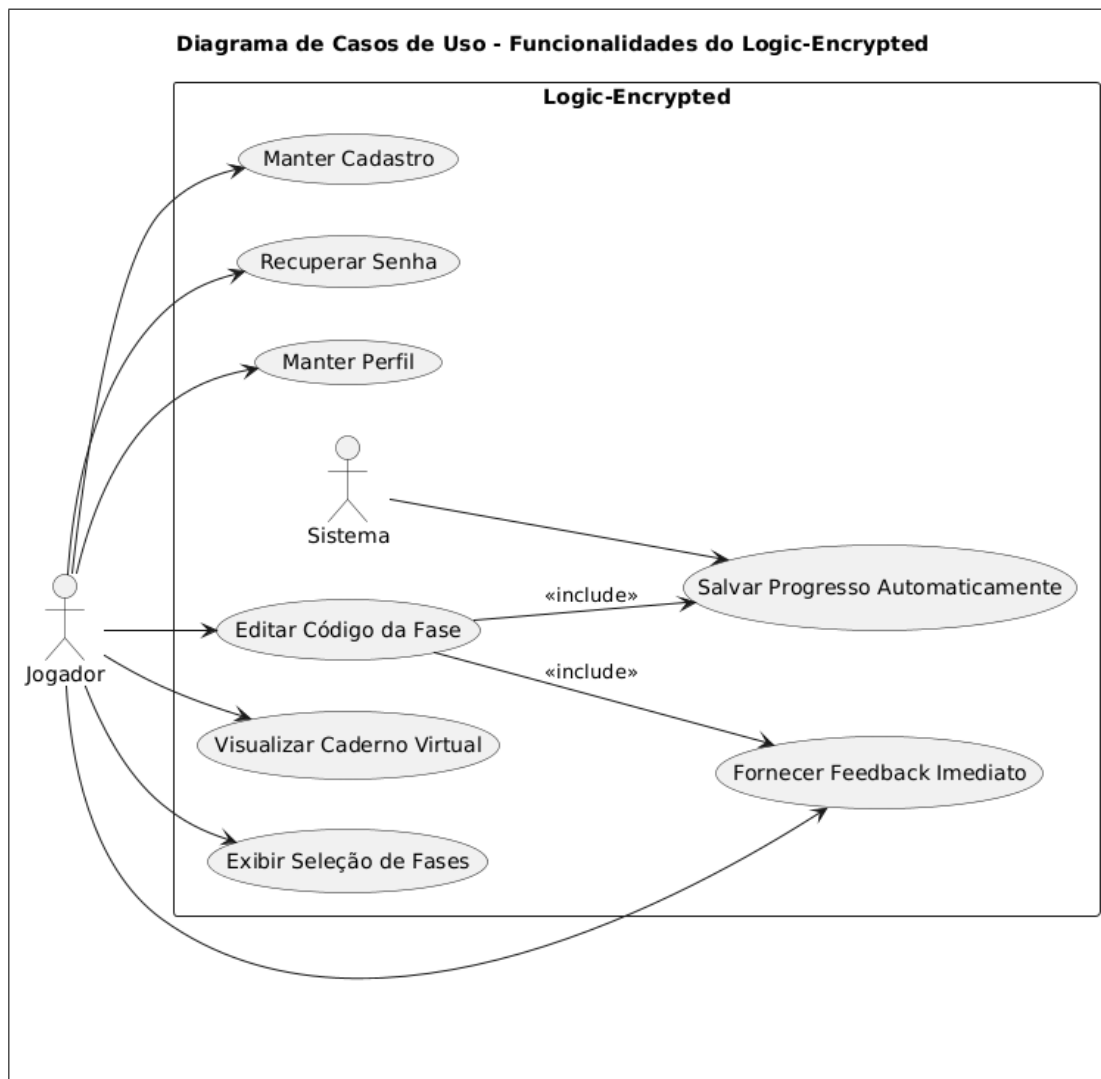


Figura 6: Diagrama de Casos de Uso – Funcionalidades do Sistema

2.4.5 Diagrama de Sequência

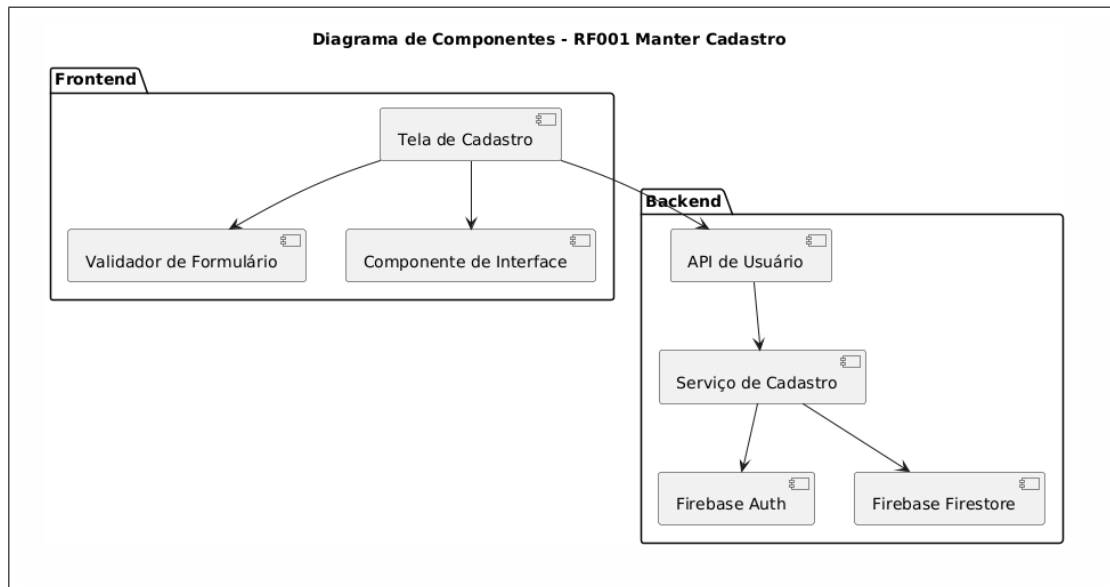


Figura 7: Diagrama de Componentes – Manter Cadastro (RF001)

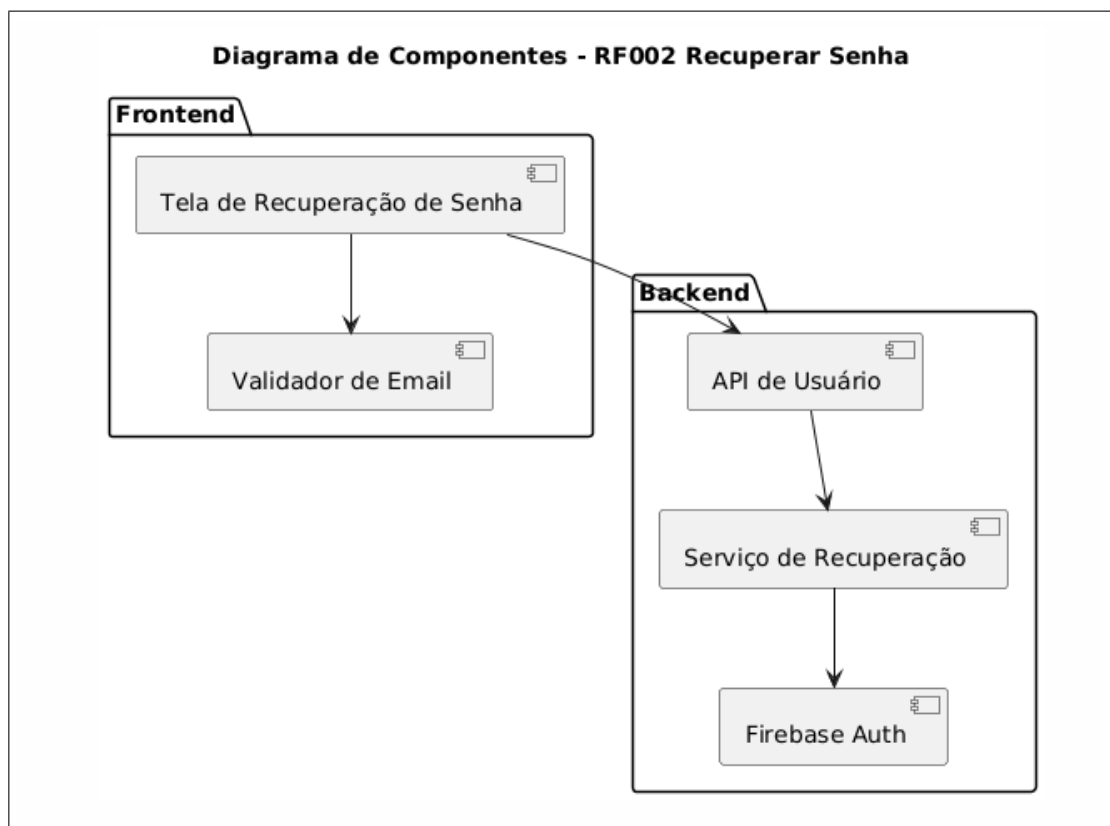


Figura 8: Diagrama de Componentes – Recuperar Senha (RF002)

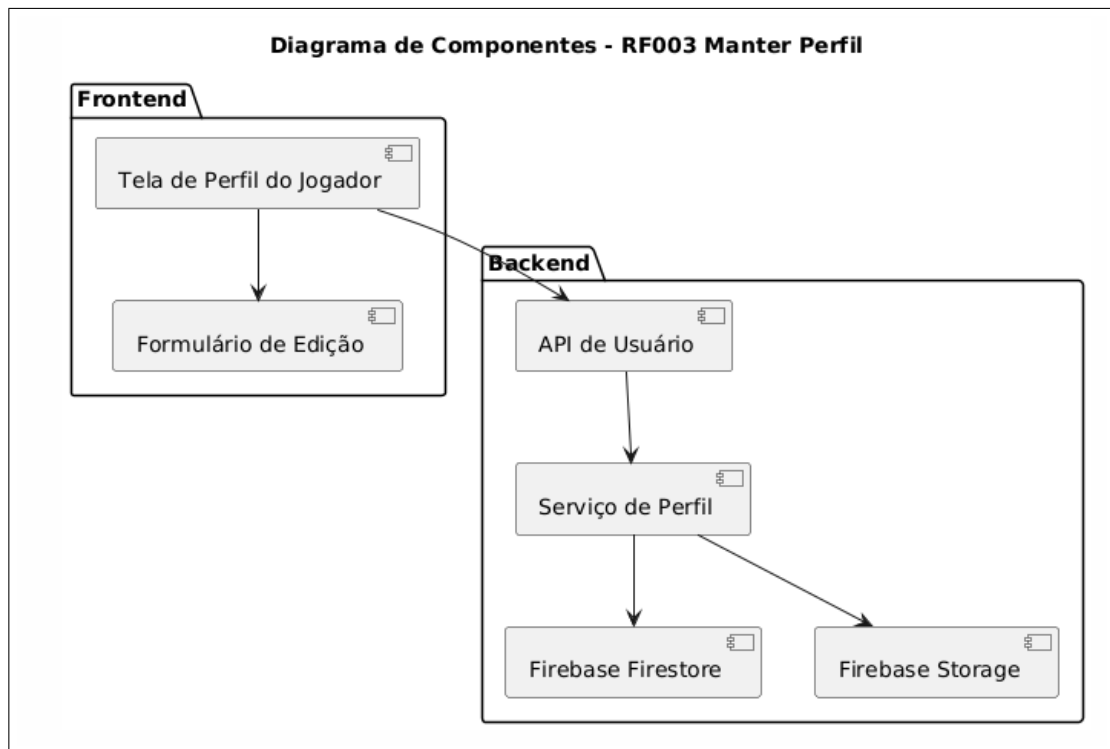


Figura 9: Diagrama de Componentes – Manter Perfil (RF003)

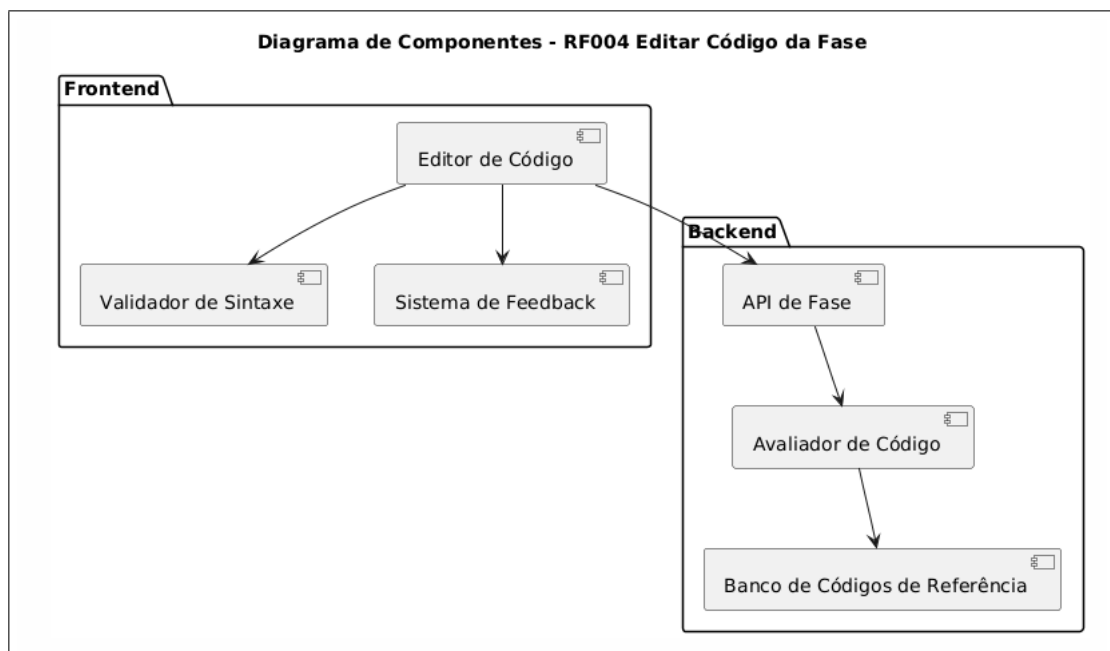


Figura 10: Diagrama de Componentes – Editar Código da Fase (RF004)

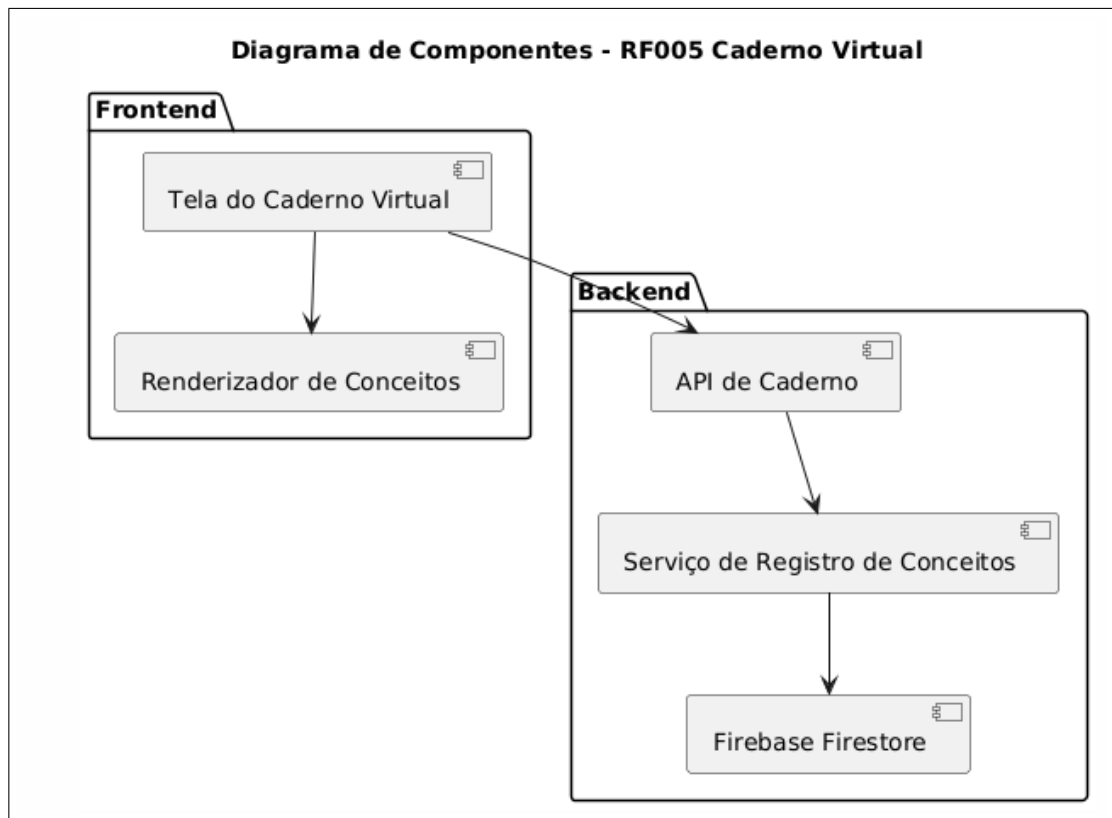


Figura 11: Diagrama de Componentes – Caderno Virtual (RF005)

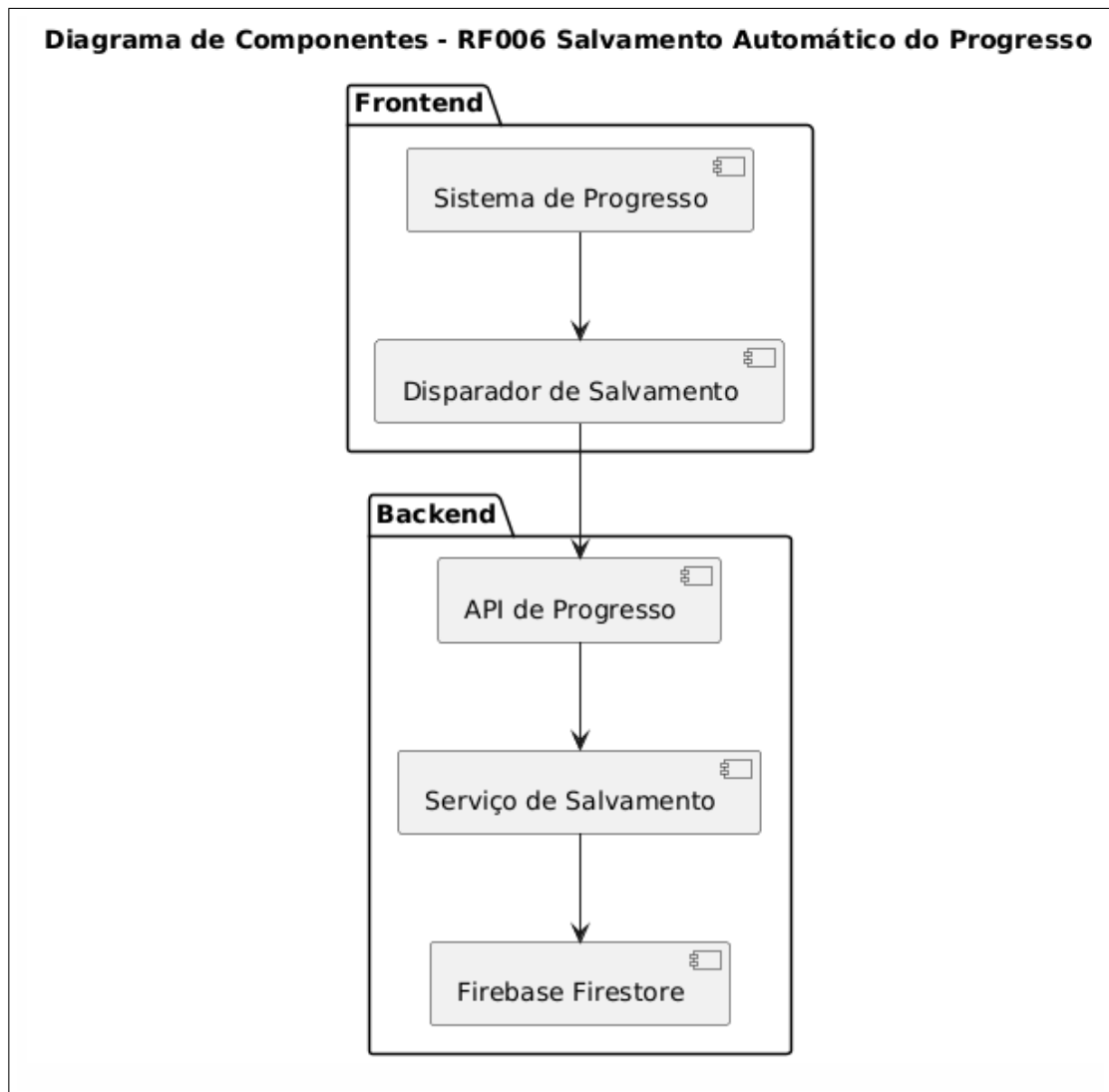


Figura 12: Diagrama de Componentes – Salvamento Automático (RF006)

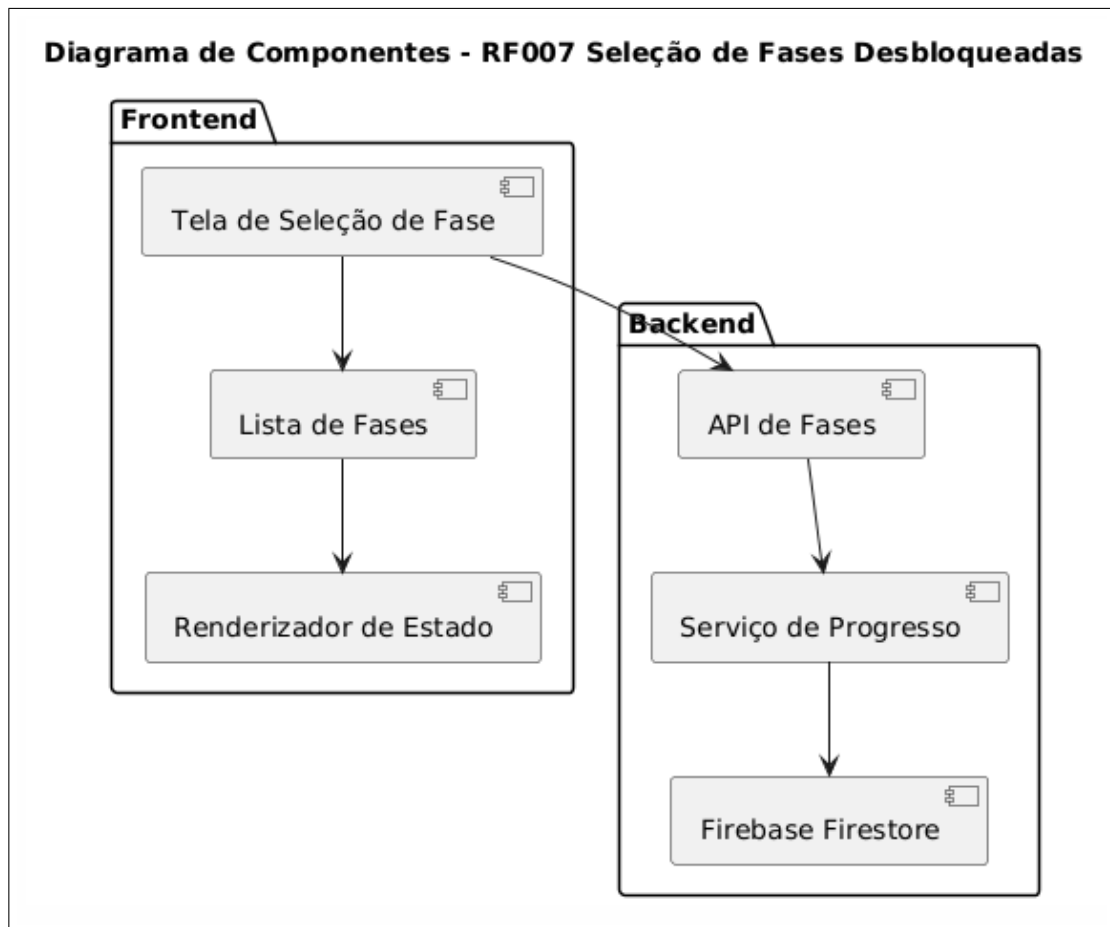


Figura 13: Diagrama de Componentes – Seleção de Fases Desbloqueadas (RF007)

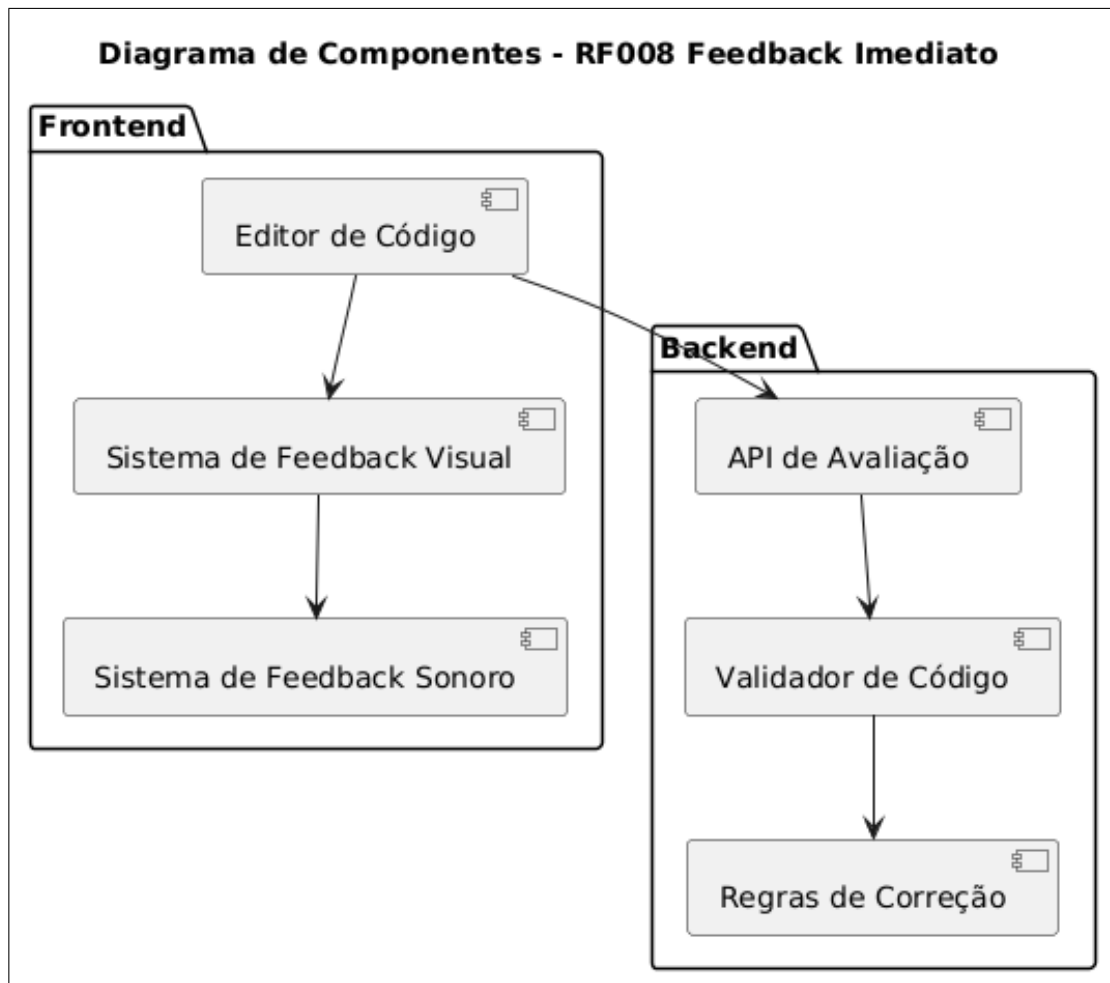


Figura 14: Diagrama de Componentes – Feedback Imediato (RF008)

2.5 Protótipos das Interfaces

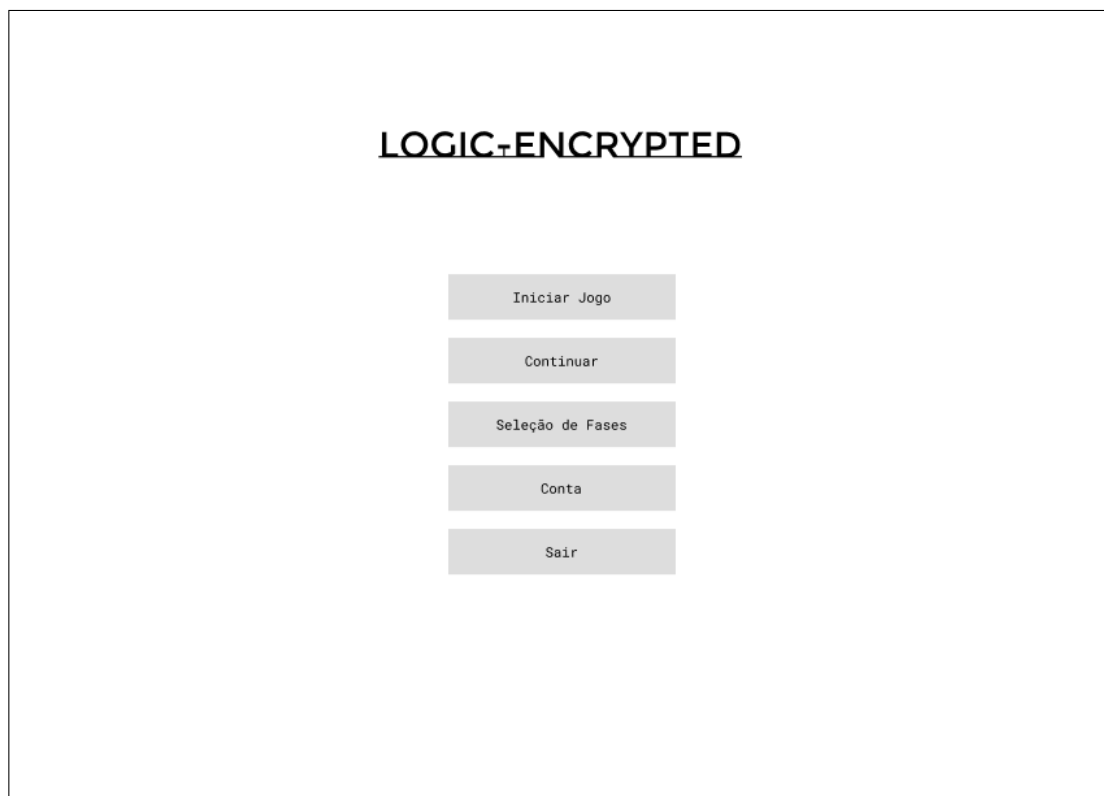


Figura 15: Protótipo da Tela Inicial do Jogo

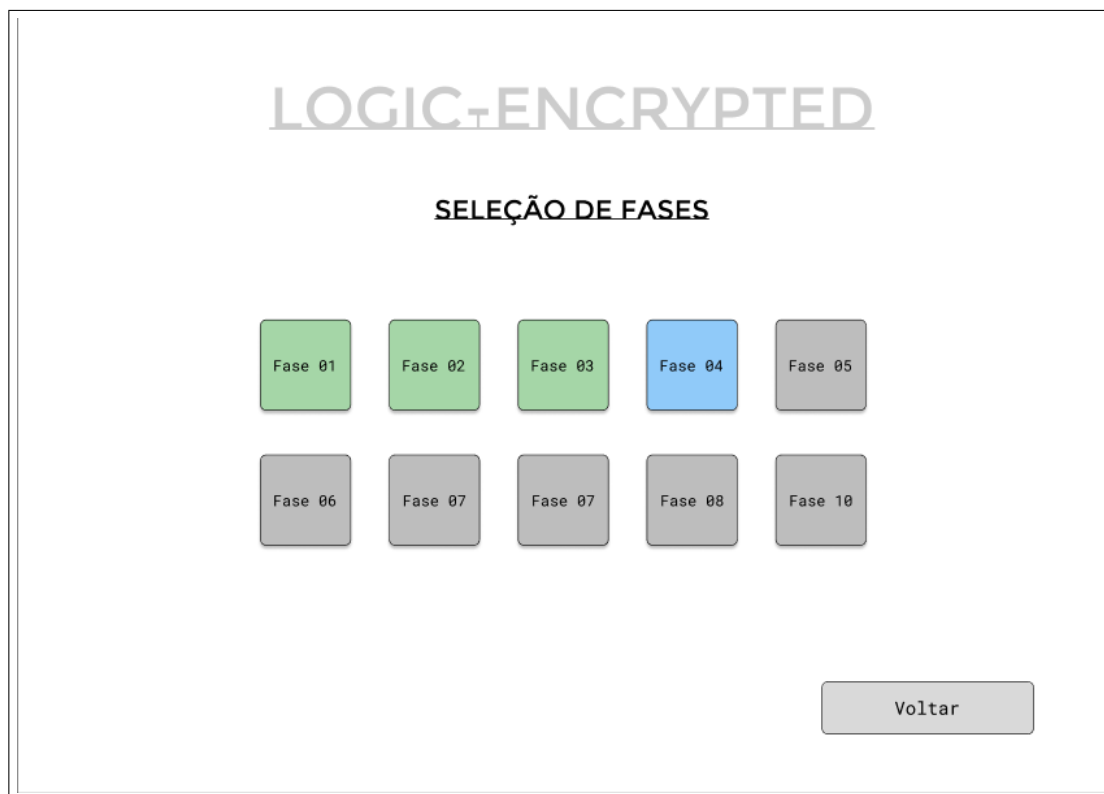


Figura 16: Protótipo da Tela de Seleção de Fases

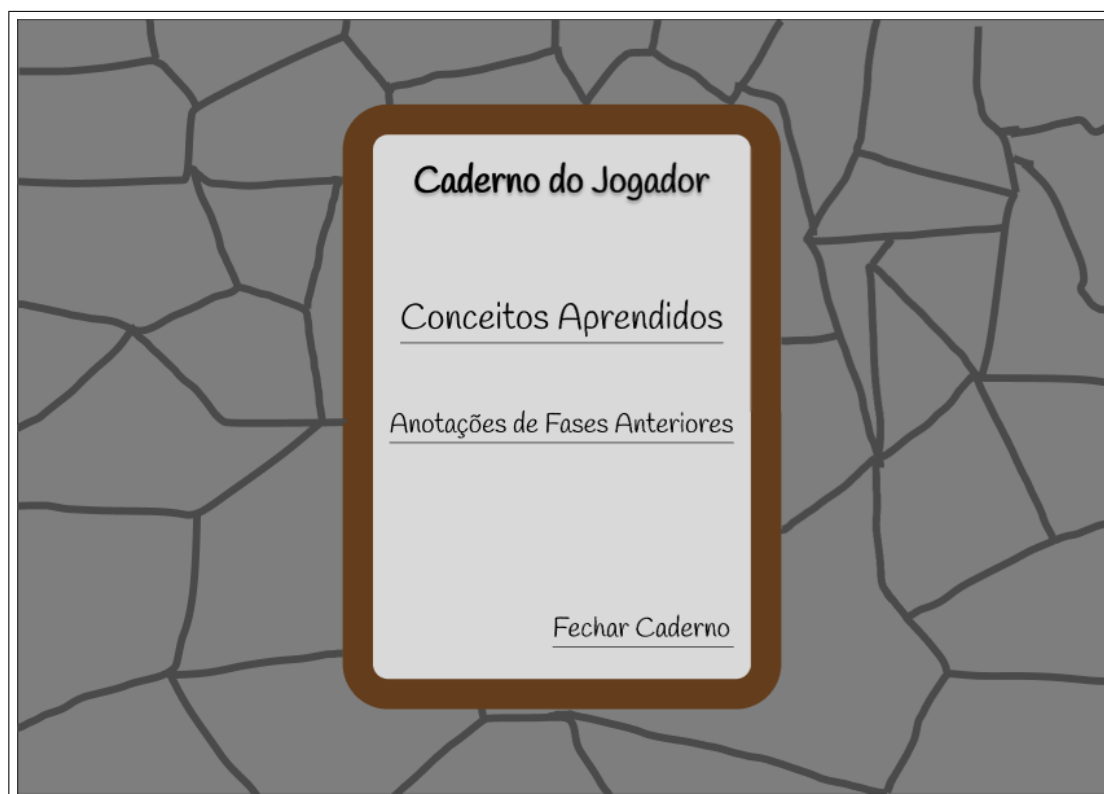


Figura 17: Protótipo da Tela do Caderno Virtual

2.6 Fatos e Suposições Relevantes

2.6.1 Fatos Relevantes

Durante a definição e planejamento do projeto *Logic-Encrypted*, foram identificados alguns fatos que influenciam diretamente as decisões de desenvolvimento e o escopo da solução. Esses fatos são considerados verdadeiros e inquestionáveis para os propósitos deste projeto:

- Estudantes iniciantes em programação apresentam dificuldades em compreender conceitos abstratos da lógica computacional sem apoio visual ou prático.
- A evasão em disciplinas introdutórias de programação é significativa, especialmente quando ministradas com metodologias tradicionais.
- A BNCC (Base Nacional Comum Curricular) reconhece o pensamento computacional como competência fundamental a ser desenvolvida durante a formação básica.
- Há uma tendência crescente na adoção de recursos digitais gamificados como ferramenta de apoio ao ensino em instituições de ensino públicas e privadas.
- A infraestrutura de muitos laboratórios escolares ainda é limitada, com computadores de baixo desempenho e acesso restrito à internet.

2.7 Escopo do Trabalho

O presente trabalho tem como escopo o desenvolvimento de um jogo educacional digital, denominado *Logic-Encrypted*, voltado especificamente para o ensino de lógica de programação a estudantes iniciantes. O projeto será implementado como uma aplicação web, acessível diretamente via navegadores modernos, dispensando a necessidade de instalação ou dependência de servidores externos. Essa característica amplia significativamente o alcance e a aplicabilidade do sistema, especialmente em contextos educacionais com limitações de infraestrutura tecnológica.

O foco principal está na construção de uma experiência interativa, lúdica e pedagógica que possibilite ao jogador aprender conceitos fundamentais da lógica computacional, como estruturas de decisão, repetições, operadores e sintaxe básica de código, por meio da resolução de desafios em formato de puzzle. O escopo também abrange o desenvolvimento de um caderno virtual integrado, onde os jogadores poderão consultar conteúdos explicativos, revisar conceitos e fazer anotações personalizadas, promovendo um ciclo contínuo de aprendizado teórico e prático.

Além do aspecto funcional, o projeto contempla o acompanhamento do progresso do jogador por meio de um sistema de relatório de desempenho. Esse relatório é projetado para fornecer dados sobre a evolução individual do estudante, contribuindo para a autorreflexão e permitindo que professores possam adotar o jogo como instrumento complementar de avaliação formativa.

Importa destacar que o escopo deste trabalho está restrito ao ambiente de ensino-aprendizagem de lógica de programação, sem abordar conteúdos relacionados a linguagens de programação específicas, algoritmos avançados ou conceitos de engenharia de software. O desenvolvimento concentra-se na etapa inicial do processo formativo do estudante, respeitando os limites pedagógicos de uma ferramenta introdutória.

A decisão por utilizar jogos digitais como meio de ensino encontra respaldo em estudos que demonstram sua eficácia na promoção do engajamento e da motivação dos estudantes. Segundo McGonigal (2012), os jogos têm o potencial de transformar atividades desafiadoras em experiências significativas e satisfatórias, estimulando a persistência, a colaboração e o pensamento estratégico. Dessa forma, o escopo do trabalho busca alinhar os objetivos técnicos e

pedagógicos do projeto à proposta de inovação educacional por meio da gamificação.

REFERÊNCIAS

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília: MEC, 2018.

MCGONIGAL, Jane. **A realidade em jogo: por que os games nos tornam melhores e como eles podem mudar o mundo**. Rio de Janeiro: BestSeller, 2012.

RODRIGUES, G. L. et al. Gamificação no ensino da computação: uma revisão sistemática. *Revista Brasileira de Informática na Educação*, v. 27, n. 1, p. 25–48, 2019.

SHELL, Jesse. **A Arte do Game Design: o livro original**. São Paulo: Cengage Learning, 2017.