Curso Técnico em Desenvolvimento de Sistemas – SENAI

RELATÓRIO DE SISTEMA

Equipe: Adrian, Fabiano, Gustavo, Jean, Lucas, Luis, Matheus, Miguel, Pedro  
Ano: 2025

1. Introdução

2. Metodologia   
 2.1 Início com arquitetura: Fluxograma, Diagrama de classes e Casos de uso  
 2.2 Modelagem do banco de dados  
 2.3 Desenvolvimento do Figma  
 2.4 Desenvolvimento Back-end e Front-end

3. Requisitos do sistema  
 3.1 Requisitos funcionais  
 3.2 Requisitos não funcionais   
 3.2. Regras de negócio

4. Descrição do sistema  
 4.1 Visão geral  
 4.2 Funcionalidades   
 4.2.1 Funcionalidades para usuários (Consumidores)   
 4.2.2 Funcionalidades para administradores  
 4.2.3 Segurança, acessibilidade e performance  
 4.2.4 Funcionalidade dos desenvolvedores Indie

5. Tecnologias utilizadas

6. Arquitetura do sistema   
 6.1 Diagrama de classe  
 6.2 Fluxograma   
 6.3 Caso de uso

7. Cronograma de desenvolvimento  
 7.1 Organização do Trello  
 7.2 Modelagem do banco de dados  
 7.3 Diagrama de classes  
 7.4 Figma — Protótipo visual  
 7.5 Fluxograma  
 7.6 Primeiros passos no Back-end  
 7.7 Primeiros passos no Front-end  
 7.8 Desenvolvimento do Back-end  
 7.9 Desenvolvimento do Front-end  
 7.10 Finalização do Back-end  
 7.11 Finalização do Front-end

8. Plano de testes  
 8.7 Testes funcionais  
 7.8 Testes de integração  
 7.9 Testes de banco de dados  
 7.10 Testes de interface e usabilidade

9. Considerações finais

10. Reuniões

**1. Introdução**

Este relatório descreve o desenvolvimento de uma plataforma de e-commerce voltada exclusivamente para empresas iniciantes no mercado de jogos independentes (indie). A plataforma tem como objetivo criar um ambiente digital que favoreça o lançamento e a venda de jogos, conectando desenvolvedores independentes a consumidores que buscam experiências inovadoras no mundo dos games. O sistema oferece uma estrutura que permite desde o cadastro dos jogos, passando por uma análise comercial, até a disponibilização para venda ao público.

**2. Me**t**odologia**

### 2.1. Início com arquitetura: Fluxograma, Diagrama de classes e Casos de uso

O primeiro passo do projeto foi liderado pelos nossos dois arquitetos de software, Luis e Miguel. Eles ficaram responsáveis por estabelecer a base estrutural do sistema, criando os artefatos essenciais para nortear todo o desenvolvimento. Assim, elaboraram primeiramente o **fluxograma**, que detalhou visualmente o fluxo de processos e funcionalidades principais da plataforma de gestão de jogos indie.

Em paralelo, deram continuidade à elaboração do **diagrama de classes**, definindo claramente as entidades que fariam parte do sistema, suas propriedades, métodos e as relações entre elas. Este trabalho foi fundamental para garantir a coesão e a consistência do modelo de dados e das regras de negócio.

Complementando esta etapa, os arquitetos também produziram os **casos de uso**, que documentaram os cenários de interação entre os usuários (administradores, desenvolvedores de jogos, avaliadores e clientes) e o sistema. Esses casos de uso permitiram mapear todos os requisitos funcionais e assegurar que as expectativas do produto fossem bem compreendidas por toda a equipe.

### 2.2. Modelagem do banco de dados

Enquanto os arquitetos conduziam a construção da estrutura conceitual do sistema, eu, Pedro, na função de Product Owner, iniciei simultaneamente a **modelagem do banco de dados**. Esta etapa foi conduzida em estreita colaboração com os arquitetos, de modo a garantir que a modelagem refletisse com precisão os elementos definidos no diagrama de classes.

A modelagem contemplou a definição das tabelas, atributos, chaves primárias, estrangeiras e índices necessários para assegurar a integridade e a eficiência no armazenamento e recuperação das informações, tão cruciais para o funcionamento do sistema. Essa modelagem serviu como a espinha dorsal para as próximas fases de desenvolvimento.

### 2.3. Desenvolvimento do Figma

Com a estrutura de dados bem definida e validada, avançamos para a fase de **desenvolvimento das interfaces visuais** utilizando a ferramenta Figma, a mesma foi conduzida pelo desenvolvedor front - end Lucas. Esta etapa teve início um pouco depois, já com uma base sólida que possibilitou aos designers criar protótipos e mockups alinhados com a lógica de dados e fluxos previamente estabelecidos.

O Figma foi essencial para ilustrar a experiência do usuário (UX) e a interface gráfica (UI) do sistema. As telas prototipadas ajudaram a materializar o produto antes mesmo do início da codificação, facilitando validações com stakeholders e ajustando pontos importantes relacionados à usabilidade e design.

### 2.4. Desenvolvimento Back-end e Front-end

Após a conclusão da modelagem visual no Figma e com a estrutura do banco de dados e arquitetura bem estabelecidas, iniciamos a etapa de **desenvolvimento simultâneo** das partes **back-end** e **front-end**.

Os nossos desenvolvedores de back-end, Jean e Gustavo, começaram a implementar a lógica de negócio, as APIs e a integração com o banco de dados. Garantiram que todas as operações de cadastro, avaliação e gerenciamento de jogos estivessem devidamente implementadas, seguindo as melhores práticas de segurança e performance.

Em paralelo, os desenvolvedores de front-end, Adrian, Matheus, Lucas e Fabiano, passaram a traduzir os protótipos do Figma em componentes funcionais e interativos, utilizando as tecnologias definidas previamente pela equipe de arquitetura. Este desenvolvimento simultâneo permitiu uma evolução rápida e coordenada do sistema, promovendo uma integração contínua entre as camadas de apresentação e de negócio.

**3. Requisitos do sistema**

**3.1 Requisitos funcionais**

RF001 – Exibir página inicial com propósito da plataforma e resultados obtidos com jogos já cadastrados.

RF002 – Exibir jogos já aprovados e disponíveis para compra.

RF003 – Disponibilizar botão “Faça Parte” na navbar com contatos corporativos.

RF004 – Permitir que desenvolvedores enviem propostas de jogos por meio de canais externos (e-mail, WhatsApp).

RF005 – Permitir que administradores cadastrem jogos aprovados no sistema, incluindo capa, descrição, vídeo, arquivo e preço.

RF006 – Exibir informações completas de cada jogo na página do produto.

RF007 – Permitir que usuários comprem jogos pela plataforma.

RF008 – Disponibilizar arquivo de instalação do jogo após confirmação da compra.

RF009 – Permitir avaliação do site após o processo de compra.

RF010 – Controlar acesso com permissões para usuários comuns, e administradores.

RF011 - Listar avaliações de usuários nas páginas dos jogos.

RF012 - Permitir que usuários visualizem seu histórico de compras.

RF013 - Permitir que o administrador consiga fazer alterações no sistema .

RF014 - Permitir que o administrador lance os jogos no sistema

**3.2 Requisitos não funcionais**

RNF001 - Implementar sistema de login e cadastro de usuários com autenticação segura.

RNF002 - O sistema deve ter textos claros e botões com contraste para atender acessibilidade básica.

RNF003 - As informações dos jogos e dos usuários devem ser salvas com segurança em um banco de dados simples.

RNF004 - O acesso ao arquivo do jogo deve ser protegido e liberado apenas para quem fez a compra.

RNF005 - O tempo de carregamento das páginas deve ser rápido (ideal: até 3 segundos).

**3.2 Regras de negócio**

RN001 – Apenas jogos aprovados pela equipe comercial podem ser cadastrados no sistema.

RN002 – O jogo só poderá ser baixado após a confirmação de pagamento.

RN003 – O jogo precisa conter obrigatoriamente capa, descrição, história, preço e vídeo demonstrativo.

RN004 – A comunicação com os desenvolvedores ocorrerá fora da plataforma, via e-mail ou WhatsApp corporativo.

RN005 – A avaliação do usuário é solicitada obrigatoriamente após a compra de um jogo.

RN006 – A proposta de jogo será analisada em até 30 dias úteis (prazo a ser definido).

RN007 – Os arquivos dos jogos devem ser armazenados em local seguro, com controle de acesso por autenticação.

**4. Descrição do sistema**

**4.1 Visão geral**

O sistema desenvolvido é uma plataforma de e-commerce dedicada à venda e divulgação de jogos indie, criada para resolver um dos principais desafios enfrentados por desenvolvedores independentes: a dificuldade de obter visibilidade e alcançar o público consumidor. Inspirado na trajetória de Leon e sua equipe, o sistema foi idealizado para ser um espaço justo, transparente e acessível, onde criadores podem apresentar, divulgar e vender seus jogos de maneira simplificada e segura.

A plataforma é gerida pelos próprios desenvolvedores, que também exercem o papel de administradores, com total autonomia sobre o cadastro, publicação e gerenciamento dos seus jogos. Essa decisão reforça o compromisso da plataforma com a descentralização e o empoderamento dos criadores, reduzindo intermediários e garantindo que cada desenvolvedor tenha controle direto sobre sua produção.

**4.2 Funcionalidades**

**4.2.1 Funcionalidades para Usuários (Consumidores)**

**Página Inicial:** Exibe de forma clara o propósito da plataforma, apresentando os resultados já alcançados e destacando o impacto positivo sobre os desenvolvedores indie (RF001).

**Catálogo de Jogos:** Disponibiliza uma lista de todos os jogos aprovados e publicados, com possibilidade de explorar detalhes como capa, descrição, história, vídeo demonstrativo e preço (RF002 e RF006).

**Processo de Compra:** Permite que usuários adquiram jogos diretamente pela plataforma, com segurança e simplicidade (RF007). Após a confirmação do pagamento, o arquivo de instalação é liberado exclusivamente para o comprador (RF008).

**Avaliações e Histórico:** Após a compra, os usuários são obrigatoriamente convidados a avaliar o site e sua experiência (RF009, RN005). Além disso, podem consultar o histórico completo de suas compras e acessar os jogos adquiridos (RF012).

**4.2.2 Funcionalidades para administradores**

Como desenvolvedores e administradores são o mesmo perfil, este usuário possui um conjunto mais amplo de responsabilidades e permissões:

* **Envio de Propostas:** O contato inicial com a plataforma ocorre via canais externos, como e-mail ou WhatsApp, através do botão “Faça Parte” na navbar (RF003 e RF004).
* **Cadastro de Jogos:** Após a aprovação, os desenvolvedores podem cadastrar seus jogos diretamente no sistema, incluindo os elementos obrigatórios: capa, descrição, vídeo demonstrativo, história e preço (RF005 e RN003).
* **Gestão e Manutenção:** Desenvolvedores-administradores podem também realizar alterações nos conteúdos cadastrados, atualizar informações ou ajustar preços, bem como lançar novos jogos no sistema (RF013 e RF014).
* **Controle de Acesso:** Apesar de acumularem funções, o sistema mantém uma estrutura clara de permissões, diferenciando usuários comuns (compradores) dos administradores/desenvolvedores que têm acesso às ferramentas de gestão (RF010).

**4.2.3 Segurança, acessibilidade e performance**

Login Seguro: O sistema exige autenticação segura para proteger os dados dos usuários e garantir que apenas quem realizou a compra tenha acesso ao arquivo do jogo (RNF001 e RNF004).

Armazenamento Seguro: Todos os arquivos dos jogos são armazenados de forma protegida, com controle rigoroso sobre quem pode acessar e baixar (RN007).

Acessibilidade: A interface foi desenvolvida com textos claros e botões de alto contraste, assegurando o atendimento às necessidades básicas de acessibilidade (RNF002).

Desempenho: O sistema é otimizado para carregar páginas rapidamente, com um tempo ideal de até 3 segundos (RNF005).

### 4.2.4 .Funcionalidade dos desenvolvedores Indie

A plataforma foi concebida para ser um espaço de acolhimento e valorização de **desenvolvedores independentes** de jogos digitais, promovendo uma oportunidade real para que criadores com recursos limitados possam divulgar e comercializar seus projetos.

Visando facilitar o acesso e a integração desses desenvolvedores ao ecossistema da plataforma, foi implementada a funcionalidade **"Faça Parte"**, posicionada de maneira estratégica na barra de navegação. Por meio desta opção, qualquer criador interessado pode obter informações sobre o funcionamento da plataforma, os critérios para publicação e os canais de comunicação com a equipe responsável.

O processo ocorre de forma simples e transparente: o desenvolvedor envia sua proposta utilizando canais externos, como e-mail ou WhatsApp corporativo, onde receberá orientações sobre os próximos passos. Após análise e aprovação da equipe de curadoria, o criador passa a ter acesso às funcionalidades administrativas, podendo cadastrar, gerenciar e comercializar seu jogo diretamente pelo sistema.

Esse fluxo foi idealizado para reduzir barreiras burocráticas e técnicas, proporcionando uma experiência acessível e colaborativa, fortalecendo a presença dos desenvolvedores independentes no mercado e ampliando a diversidade e qualidade do catálogo de jogos disponíveis para os usuários.

**5. Tecnologias utilizadas**

A divisão do projeto em diferentes tecnologias permitiu uma organização eficiente e o aproveitamento máximo das potencialidades de cada área. A escolha por separar o desenvolvimento em camadas específicas, utilizando ferramentas adequadas a cada função, garantiu uma distribuição equilibrada do trabalho entre os integrantes, além de otimizar o tempo e a qualidade das entregas.

Para a programação, o uso do IntelliJ Ultimate associado ao Spring Boot proporcionou agilidade na criação de uma arquitetura robusta e escalável no back-end, enquanto o MySQL Workbench foi responsável pela modelagem precisa do banco de dados. O front-end, por sua vez, foi desenvolvido com foco na experiência do usuário, utilizando JavaScript, HTML e CSS, contando ainda com o apoio do Cloudflare para estilização e desempenho.

A prototipação no Figma possibilitou visualizar e ajustar previamente as interfaces, garantindo que o layout estivesse alinhado com as funcionalidades. Já as ferramentas de documentação e gestão, como Google Drive, Google Docs e Trello, centralizaram a comunicação e o acompanhamento das tarefas, mantendo o fluxo de trabalho bem estruturado. O uso de Draw.io e LucidChart na modelagem da arquitetura também contribuiu para uma visão clara e compartilhada das estruturas do sistema.

Essa divisão estratégica permitiu que cada integrante focasse em uma área específica, utilizando a tecnologia mais adequada, promovendo assim um desenvolvimento mais eficiente, organizado e colaborativo.

Segue abaixo a lista de tecnologias utilizadas no desenvolvimento do projeto,

Programação:

IntelliJ Ultimate

MySQL Workbench

Insomnia

Google Chrome (Chat Gpt)

GitHub

Documentação:

Google Drive

Google Docs

Trello

Protótipo:

Figma

Arquitetura de Sistema

Draw.io ou LucidChart

Linguagens de programação:

Java 23

JavaScript

HTML

CSS

Framework:  
Springboot (para Java)

Cloudflare (para css)

**6. Arquitetura do sistema**

Nesta seção, apresentamos a arquitetura do sistema, com o objetivo de ilustrar visualmente a estrutura lógica, o fluxo de processos e as interações entre os principais componentes que compõem a plataforma de e-commerce de jogos indie.

A definição da arquitetura foi fundamental para garantir a organização, a clareza e a eficiência no desenvolvimento do projeto, servindo como referência para todas as etapas posteriores: modelagem de dados, desenvolvimento de interface e implementação das funcionalidades.

A seguir, estão descritos e apresentados, por meio de diagramas e fluxos, os principais artefatos produzidos durante a fase de planejamento arquitetural:

**6.1 Diagrama de Classe**

O **Diagrama de Classe** foi o primeiro modelo elaborado, com a finalidade de representar as principais entidades que compõem o sistema, bem como os relacionamentos existentes entre elas. A partir desse modelo, foi possível definir com clareza as estruturas de dados essenciais, os atributos que caracterizam cada entidade e os métodos necessários para a execução das operações previstas.

Este diagrama serviu como base para a **modelagem do banco de dados**, permitindo que a estrutura lógica fosse transformada posteriormente em uma estrutura física robusta e eficiente. Além disso, facilitou o entendimento das interdependências entre os componentes, otimizando o trabalho dos desenvolvedores, especialmente na fase de implementação das regras de negócio.

Entre as classes mais relevantes modeladas estão: **Usuário**, **Jogo**, **Compra**, **Avaliação** e **Administrador** (este último representando o papel adicional exercido pelos próprios desenvolvedores). As relações de associação, composição e herança foram cuidadosamente definidas para garantir a coesão e a reusabilidade do código.

**6.2 Fluxograma**

O **Fluxograma** foi elaborado com o objetivo de representar, de forma sequencial e lógica, os principais fluxos operacionais do sistema. Este artefato foi fundamental para visualizar o comportamento do sistema frente às ações dos usuários, identificando as etapas, as decisões críticas e os possíveis pontos de falha ou gargalos.

A construção do fluxograma permitiu que a equipe analisasse de maneira objetiva o **fluxo de processos**, desde o primeiro acesso à plataforma até a conclusão da compra e a disponibilização do arquivo do jogo para download. Além disso, ele orientou o trabalho dos times de desenvolvimento, garantindo que todas as operações fossem implementadas conforme o fluxo previamente definido.

Os fluxos modelados contemplam tanto as ações realizadas por usuários comuns (como navegação, compra e avaliação), quanto aquelas executadas pelos desenvolvedores-administradores (como cadastro e gestão de jogos).

A clareza proporcionada por este diagrama também foi essencial para facilitar o alinhamento entre as equipes de frontend e backend, permitindo que ambas atuassem simultaneamente, mas de forma coesa e sincronizada.

**6.3 Caso de uso**

O Diagrama de Caso de Uso foi desenvolvido com o propósito de mapear e documentar todas as interações possíveis entre os diferentes atores e o sistema, além de definir o escopo funcional de maneira clara e objetiva. Este artefato foi essencial para garantir que todos os requisitos funcionais identificados na fase de levantamento fossem efetivamente contemplados durante a implementação.

No caso desta plataforma, os dois principais atores modelados foram: Usuário (Comprador) e Desenvolvedor-Administrador, sendo que este último acumula funções tanto de criador de conteúdo quanto de gestor da sua própria produção. O diagrama explicita as funcionalidades disponíveis para cada perfil, como: cadastrar jogos, visualizar catálogo, realizar compras, gerenciar arquivos, efetuar downloads e registrar avaliações.

A elaboração deste diagrama contribuiu diretamente para a validação dos requisitos, auxiliando na detecção de possíveis lacunas ou redundâncias e orientando a priorização das funcionalidades mais críticas para o negócio.

**7. Cronograma de desenvolvimento**

### 7.1. Organização do Trello

* **Terça, 6 de maio:** Estruturação das colunas do Trello (Back-end, Front-end, Banco de Dados, Testes, Deploy).
* **Quarta, 7 de maio:** Definição de tarefas e subtarefas de cada etapa do projeto.
* **Quinta, 8 de maio:** Organização da documentação inicial no Google Drive e Google Docs.

### 7.2. Modelagem do banco de dados

* **Terça, 6 de maio:** Levantamento das entidades e requisitos do banco.
* **Quarta, 7 de maio:** Rascunho das tabelas e relacionamentos no MySQL Workbench.
* **Quinta, 8 de maio:** Estrutura inicial da modelagem.
* **Terça, 13 de maio:** Modelagem formal das entidades no MySQL Workbench.
* **Quarta, 14 de maio:** Definição de restrições, chaves primárias e estrangeiras.
* **Quinta, 15 de maio:** Criação dos scripts SQL.
* **Terça, 20 de maio:** Revisão e ajuste fino da modelagem.
* **Quarta, 21 de maio:** Verificação da integridade e normalização.
* **Quinta, 22 de maio:** Polimento final.
* **Quarta, 28 de maio:** Última revisão técnica para entrega.
* **Quinta, 29 de maio:** Confirmação e documentação final da modelagem.

### 7.3. Diagrama de classes

* **Terça, 6 de maio:** Esboço inicial no Draw.io ou LucidChart baseado na modelagem do banco.
* **Quarta, 7 de maio:** Definição das classes e relacionamentos principais.
* **Quinta, 8 de maio:** Ajuste e validação com a equipe.
* **Terça, 13 de maio:** Formalização do diagrama.
* **Quarta, 14 de maio:** Refinamento de atributos e métodos.
* **Quinta, 15 de maio:** Finalização do diagrama.
* **Terça, 20 de maio:** Ajustes de última hora conforme evolução do código.
* **Quarta, 21 de maio:** Integração do diagrama na documentação técnica.
* **Quinta, 22 de maio:** Revisão final.
* **Quarta, 28 de maio:** Conferência e polimento do diagrama.
* **Quinta, 29 de maio:** Entrega final junto à documentação.

### 7.4. Figma — Protótipo visual

* **Terça, 6 de maio:** Criação dos layouts iniciais: Home, Catálogo e Detalhes.
* **Quarta, 7 de maio:** Protótipos de Checkout, Avaliação e Histórico de Compras.
* **Quinta, 8 de maio:** Protótipo da aba **"Faça Parte"**.
* **Terça, 13 de maio:** Ajustes finos nos protótipos com base na modelagem.
* **Quarta, 14 de maio:** Revisão de responsividade e interação.
* **Quinta, 15 de maio:** Finalização da versão navegável.
* **Terça, 20 de maio:** Refinamento visual conforme desenvolvimento do front.
* **Quarta, 21 de maio:** Alinhamento com padrões de estilo.
* **Quinta, 22 de maio:** Últimos ajustes.
* **Quarta, 28 de maio:** Polimento estético final.
* **Quinta, 29 de maio:** Entrega definitiva dos protótipos.

### 7.5. Fluxograma

* **Terça, 6 de maio:** Esboço do fluxo principal: navegação, compra e cadastro.
* **Quarta, 7 de maio:** Fluxos secundários: avaliação e histórico de compras.
* **Quinta, 8 de maio:** Ajustes e validação do fluxo.
* **Terça, 13 de maio:** Formalização no Draw.io ou LucidChart.
* **Quarta, 14 de maio:** Revisão lógica do fluxo.
* **Quinta, 15 de maio:** Finalização do fluxograma.
* **Terça, 20 de maio:** Ajustes conforme novos requisitos ou alterações.
* **Quarta, 21 de maio:** Revisão técnica.
* **Quinta, 22 de maio:** Integração com a documentação.
* **Quarta, 28 de maio:** Revisão final e polimento do fluxograma.
* **Quinta, 29 de maio:** Entrega definitiva.

### 7.6. Primeiros passos no Back-end

* **Terça, 6 de maio:** Estruturação do projeto Spring Boot no IntelliJ.
* **Quarta, 7 de maio:** Criação da arquitetura de camadas (Controller, Service, Repository).
* **Quinta, 8 de maio:** Implementação das primeiras classes e entidades.

### 7.7. Primeiros passos no Front-end

* **Terça, 6 de maio:** Estruturação de pastas e arquivos (HTML, CSS, JS).
* **Quarta, 7 de maio:** Desenvolvimento do layout base com Cloudflare.
* **Quinta, 8 de maio:** Configuração da comunicação com a API.

### 7.8. Desenvolvimento do Back-end

* **Terça, 13 de maio:** Desenvolvimento de endpoints RESTful (GET, POST).
* **Quarta, 14 de maio:** Implementação de PUT, DELETE e validações básicas.
* **Quinta, 15 de maio:** Testes iniciais com Insomnia.
* **Terça, 20 de maio:** Ampliação das funcionalidades da API.
* **Quarta, 21 de maio:** Integração completa com MySQL.
* **Quinta, 22 de maio:** Ajustes e melhorias.

### 7.9. Desenvolvimento do Front-end

* **Terça, 13 de maio:** Desenvolvimento das páginas principais com HTML e CSS.
* **Quarta, 14 de maio:** Desenvolvimento das páginas secundárias.
* **Quinta, 15 de maio:** Integração inicial com o back via JS.
* **Terça, 20 de maio:** Ajustes na comunicação front-back.
* **Quarta, 21 de maio:** Implementação de interações dinâmicas.
* **Quinta, 22 de maio:** Melhoria da estilização com Cloudflare.

### 7.10. Finalização do Back-end

* **Quarta, 28 de maio:** Testes completos com Insomnia, tratamento de erros e validação.
* **Quinta, 29 de maio:** Análise final de qualidade do banco e documentação do back-end.

### 7.11. Finalização do Front-end

* **Quarta, 28 de maio:** Integração total entre front e back, revisão de UX.
* **Quinta, 29 de maio:** Ajustes finais de estilização e arquitetura.

**8. Plano de testes**

Na etapa de testes, iremos validar todos os fluxos principais do sistema, garantindo que as funcionalidades estejam funcionando conforme planejado, e que a integração entre as diferentes partes da aplicação — back-end, front-end e banco de dados — seja estável, eficiente e sem falhas.

### 8.1 Testes funcionais

Inicialmente, realizaremos testes em todas as funcionalidades principais, como o cadastro e login de usuários, o cadastro de jogos pelos desenvolvedores, a realização de compras, o processo de download, e a avaliação de produtos. Também testaremos a navegação geral da plataforma, incluindo o acesso à aba "Faça Parte", que direciona desenvolvedores interessados a se comunicar com a equipe.

### 8.2 Testes de integração

Após a validação das funções individuais, iremos testar a comunicação entre o front-end e o back-end, assegurando que todas as informações enviadas pelo usuário sejam corretamente processadas e armazenadas no banco de dados. Será importante garantir que o catálogo de jogos seja exibido corretamente, que os processos de compra sejam realizados sem falhas, e que o histórico de compras do usuário seja atualizado automaticamente.

### 8.3 Testes de banco de dados

Também realizaremos uma verificação completa da estrutura do banco de dados, garantindo que os registros de usuários, jogos e compras estejam sendo armazenados corretamente, que os relacionamentos entre as tabelas estejam funcionando, e que não haja problemas de integridade ou redundância de dados.

### 8.4 Testes de interface e usabilidade

Na sequência, vamos validar a interface da plataforma, verificando se as telas seguem o protótipo do Figma, se a aplicação é responsiva, e se o design está consistente em diferentes dispositivos. A navegabilidade será analisada, garantindo que o usuário consiga acessar facilmente todas as áreas da plataforma e realizar suas ações com clareza e eficiência.

**9. Considerações finais**

Este projeto foi desenvolvido com o objetivo de suprir uma demanda significativa no mercado de jogos independentes, oferecendo uma plataforma especializada que facilita a comercialização e a visibilidade de títulos criados por desenvolvedores pequenos. A ausência de espaços acessíveis e justos para esses criadores limita o potencial de inovação e diversidade no setor, o que torna esta solução ainda mais necessária.

O sistema proposto destaca-se pela combinação entre simplicidade, segurança e transparência, atendendo aos requisitos funcionais e não funcionais essenciais para uma experiência de compra e gestão eficaz. A curadoria interna assegura a qualidade dos jogos disponibilizados, enquanto os controles de acesso e proteção dos arquivos garantem a segurança para compradores e desenvolvedores.

Além disso, a plataforma foi pensada para ser acessível, com foco em usabilidade e desempenho, assegurando que os usuários tenham uma navegação rápida e intuitiva. A flexibilidade para administrar conteúdos, somada à facilidade no processo de compra e avaliação, torna o sistema uma ferramenta robusta para o mercado indie.

Em suma, este projeto contribui para a profissionalização e o fortalecimento do segmento de jogos independentes, proporcionando um ambiente confiável, justo e eficiente tanto para desenvolvedores quanto para consumidores.

**10. Reuniões**

### Reunião – 7 de maio - (Adrian, Fabiano, Jean, Gustavo, Lucas, Luis, Matheus, Miguel e Pedro) - 19:45

Nesta reunião, discutimos o levantamento de requisitos, focando em identificar quais seriam os mais plausíveis para a aplicação prática do sistema. Também abordamos o desenvolvimento do protótipo visual no Figma, preparando o time para o início da execução do front-end, garantindo que o design estivesse alinhado às necessidades do projeto.

### Reunião – 20 de maio - (Adrian, Fabiano, Jean, Gustavo, Lucas, Luis, Matheus, Miguel e Pedro) - 21:00

Conversamos sobre a estrutura geral do projeto, visando simplificar etapas e resolver problemas. Fizemos uma revisão das tarefas no Trello e conseguimos resolver as tasks que estavam atrasadas, o que permitiu alinhar o andamento das atividades e garantir uma organização mais eficiente no processo de desenvolvimento.

### Reunião – 28 de maio - (Adrian, Fabiano, Jean, Gustavo, Lucas, Luis, Miguel e Pedro) - 21:00

Neste encontro, realizamos a finalização dos testes do back-end, incluindo o tratamento de erros e as validações necessárias. Também revisamos e polimos os diagramas, a modelagem do banco de dados, os fluxogramas e os protótipos visuais, garantindo a coerência técnica de todos os artefatos. Além disso, avaliamos a integração entre o front-end e o back-end, com foco especial na experiência do usuário e no desempenho do sistema.

### Reunião – 29 de maio (Adrian, Fabiano, Jean, Gustavo, Lucas, Luis, Matheus, Miguel e Pedro) - 22:00

Na última reunião, realizamos a análise final da qualidade do banco de dados e da documentação técnica do back-end. Entregamos os protótipos visuais definitivos e efetuamos os ajustes finais na estilização e arquitetura do front-end, visando melhorar a usabilidade e a coerência visual da plataforma. Por fim, discutimos o fechamento das etapas de desenvolvimento e a preparação para a implantação do sistema.