PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Engenharia de *Software*

Projeto Integrado

Relatório Técnico

Gamer Library

Luiz Filipe Borges Lima Ferreira

Feira de Santana - BA
Outubro de 2023.

Projeto Integrado

Sumário

Projeto Integrado **2**
1. Cronograma de Trabalho 3
2. Introdução 3
3. Definição Conceitual da Solução 5
3.1 Diagrama de Casos de Uso 5
3.2 Requisitos Funcionais 6
3.3 Requisitos Não-funcionais 7
4. Protótipo Navegável do Sistema 7
5. Diagrama de Classes de Domínio 8
6. Padrão arquitetural 8
7. C4 model - Diagrama de Contexto 9
8. Tecnologias e Frameworks de Trabalho 9
9. Estrutura base do frontend 11
10. Modelo Relacional 14
11. Plano de testes 14
12. Apropriação de Horas no Projeto 16
13. Código da Aplicação 17
14. Avaliação retrospectiva 17
14.1 Objetivos Estimados 17
14.2 Objetivos Alcançados 17
14.3 Lições aprendidas 18

1. Cronograma de Trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	**Até**		
13 / 01 /23	20 / 01 /23	1. Leitura e entendimento dos materiais do Projeto integrado. Revisão de assuntos.	Entendimento geral do Projeto.
21 / 01 /23	25 / 01 /23	2. Definição do problema a ser resolvido.	Produto que será desenvolvido no Projeto.
26 / 01 / 23	03 / 02 /23	3. Testes de integração com APIs de terceiros que irão alimentar a base.	A API que apresentou os melhores resultados para ser integrada ao projeto foi a IGDB
04 / 02 /23	08 / 02 / 23	4. Relatório Técnico: Cronograma, objetivos do trabalho, apresentação do problema, descrição do produto que será desenvolvido.	Elaboração do Relatório Técnico - Descritivos.
09 / 02 / 23	14 / 02 / 23	5. Relatório Técnico: Identificação de atores, requisitos funcionais e não funcionais.	Elaboração do Relatório Técnico – Definição conceitual.
15 / 02 / 23	28 / 02 / 23	6. Relatório Técnico: Desenvolvimento do diagrama de casos de uso e seus descritivos.	Elaboração do Relatório Técnico – Casos de Uso.
01 / 03 / 23	05 / 03 / 23	7. Criação do protótipo da interface do produto.	Elaboração do Protótipo Navegável do Sistema
06 / 03 / 23	12 / 03 / 23	8. Relatório Técnico: Criação do Diagrama de Classes de domínio.	Elaboração do Relatório Técnico – Classes de Domínio
13 / 03 / 23	15 / 03 / 23	9. Finalização da primeira etapa do projeto	Primeira parte do Relatório Técnico finalizada
20 / 03 / 23	04 / 04 / 23	10. Relatório Técnico: Definição do Padrão Arquitetural do Projeto e Tecnologias.	Elaboração do Relatório Técnico – Descrição da Arquitetura escolhida e tecnologias utilizadas.
05 / 04 / 23	10 / 04 / 23	11. Relatório Técnico: Criação do Diagrama de Contexto do Projeto	Elaboração do Relatório Técnico – C4 Model
10 / 04 / 23	25 / 04 / 23	12. Relatório Técnico: Criação do Modelo Relacional do Banco de Dados	Elaboração do Relatório Técnico – Diagrama do Modelo Relacional
25/ 04 / 23	31 / 07 / 23	13. Desenvolvimento do produto	Produto funcional
01 / 08 / 23	15 / 08 / 23	14. Relatório Técnico: Apresentar Frameworks e Estrutura Base do projeto	Elaboração Relatório Técnico – Descrição dos Frameworks
01 / 09 / 23	14 / 10 / 23	15. Relatório Técnico: Criação do Plano e Relatório de Execução de Testes e Apropriação de Horas.	Finalização do relatório

2. Introdução

A popularização dos computadores, smartphones e da internet trouxeram soluções por meio de aplicações web que objetivam facilitar nossa vida e minimizar problemas do nosso dia a dia. Com o avanço da tecnologia, uma forma singular de colecionismo vem ganhando cada vez mais popularidade, o colecionismo de videogames e jogos de videogame. Colecionismo é uma prática que vem sendo seguida a vários anos no mundo inteiro e em diversos segmentos diferentes, indo desde veículos a moedas.

Como resultado, gerenciar essas coleções podem ser bastante trabalhosas à medida que crescem, tendo o colecionador que recorrer a planilhas de excel ou até caderninhos de anotações. Por consequência, o desenvolvimento de um sistema web, com o objetivo de gerenciar coleções de jogos, surge como uma solução para o problema recorrente dos colecionadores de consoles e jogos.

Sendo assim, este projeto tem como objetivo criar e registrar o processo de desenvolvimento de uma aplicação WEB. Para isso, será composta por uma API que irá oferecer recursos para o usuário interessado em gerenciar coleções de jogos, e uma interface web que irá usar os endpoints dessa API. A aplicação será uma maneira fácil e rápida do colecionador registrar a aquisição de jogos e consoles. Ele poderá registrar o valor que foi gasto, adicionar à venda ou a troca dos seus itens e consultar informações estatísticas com o valor total gasto na compra bem como histórico dos itens que estão presentes ou já passaram por sua coleção.

Objetivos específicos:

- Oferecer uma plataforma de gerenciamento para colecionadores de jogos.
- Facilitar o compartilhamento de informações de coleções de jogos e consoles.
- Oferecer de maneira rápida estatísticas de coleções aos usuários.

3. Definição Conceitual da Solução

3.1 Diagrama de Casos de Uso

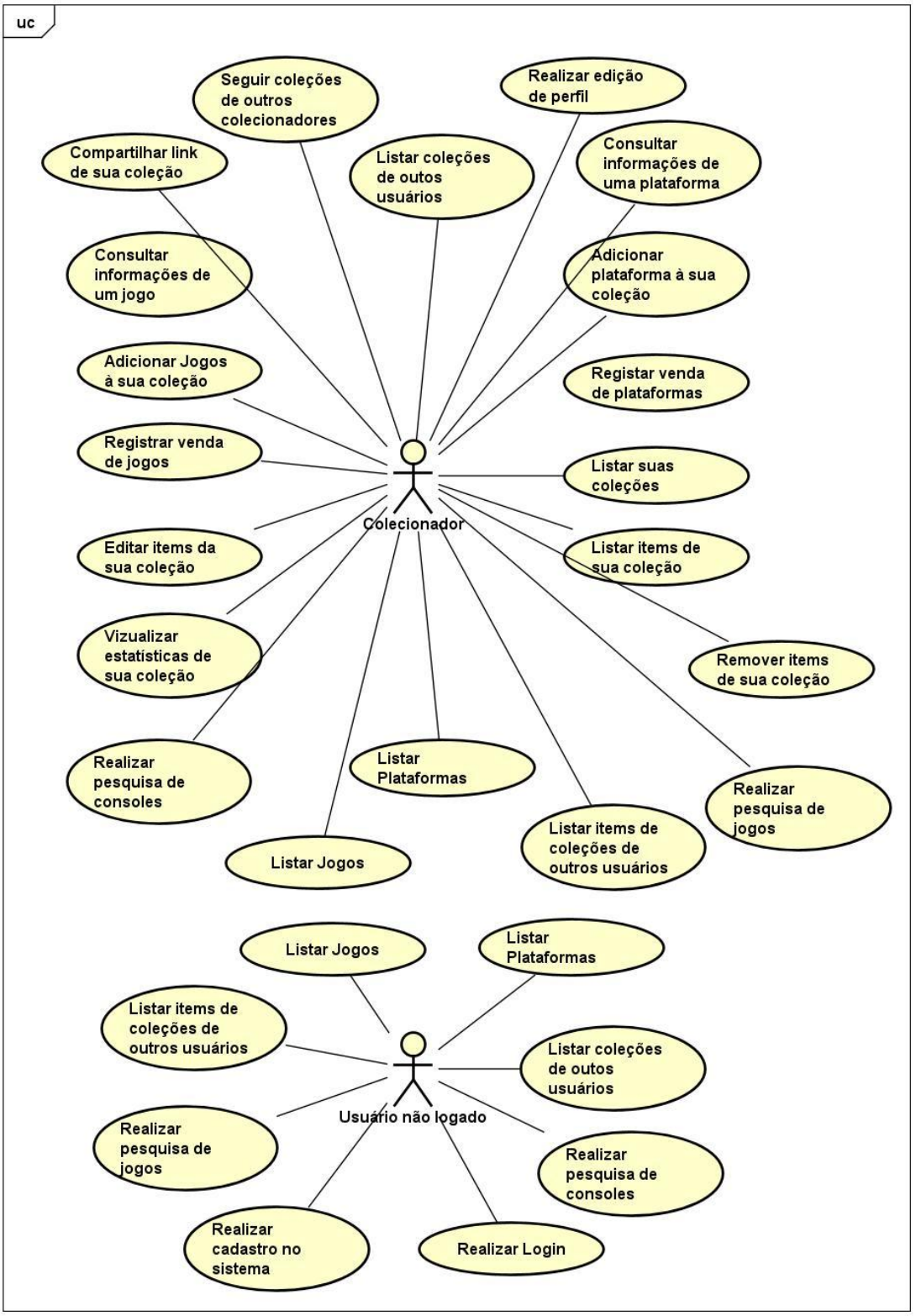


Imagem:
https://github.com/filipborgs/gamer-collection-puc/blob/main/docs/UseCase%20Diagram-v2.jpg

3.2 Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O usuário deve ser capaz de se cadastrar no sistema	B	A
RF02	O usuário deve ser capaz de listar todos os jogos	A	A
RF03	O usuário deve ser capaz de realizar pesquisas de jogos	M	M
RF04	O usuário deve ser capaz de adicionar jogos à sua coleção	M	A
RF05	O usuário deve ser capaz de abrir um jogo em detalhes	B	B
RF06	O usuário deve ser capaz de listar todos os consoles	A	A
RF07	O usuário deve ser capaz de realizar pesquisas de consoles	M	M
RF08	O usuário deve ser capaz de adicionar consoles à sua coleção de consoles	M	A
RF09	O usuário deve ser capaz de editar informações de um jogo da sua coleção de jogos	M	M
RF10	O usuário deve ser capaz de editar informações de um console na sua coleção de consoles	M	M
RF11	O usuário deve ser capaz de excluir um jogo de sua coleção de jogos	B	M
RF12	O usuário deve ser capaz de listar suas coleções de jogos	B	A
RF13	O usuário deve ser capaz de listar suas coleções de consoles	B	A
RF14	O usuário deve ser capaz de exibir os itens de sua coleção de jogos	M	A
RF15	O usuário deve ser capaz de exibir os itens de sua coleção de consoles	M	A
RF16	O usuário deve ser capaz de visualizar informações de suas coleções de jogos	M	B
RF17	O usuário deve ser capaz de visualizar informações de suas coleções de consoles	M	B
RF18	O usuário deve ser capaz de editar informações de seu perfil	B	B
RF19	O usuário deve ser capaz de compartilhar o link de sua coleção	B	M
RF20	O usuário deve ser capaz de seguir perfis de outros usuários	M	B

RF21	O usuário deve ser capaz de listar o perfil de usuários que segue	B	B
RF22	O usuário deve ser capaz de de visualizar coleções de outros usuários	M	M
RF23	O usuário deve ser capaz de registrar venda de jogos	M	M
RF24	O usuário deve ser capaz de registrar venda de consoles	M	M
RF25	O usuário deve ser capaz de se logar ao sistema	B	A

* B = Baixa, M = Média, A = Alta.

3.3 Requisitos Não-funcionais

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve estar disponível quando necessário.	A
RNF02	O sistema deve garantir a segurança das senhas dos usuários, criptografando-as antes de serem gravadas no banco de dados.	A
RNF03	O sistema web deve ser responsivo.	M
RNF04	O sistema deve atender aos requisitos do usuário.	A
RNF05	O sistema deverá utilizar codificação limpa e organizada.	A

4. Protótipo Navegável do Sistema

Link prototipo navegável:

https://www.figma.com/proto/AQQpnwUr16vrJliPK4afOV/Gamer-LIbrary?node-id=1-1040&scaling=scale-down&page-id=0%3A1

Vídeo do protótipo figma:

https://github.com/filipborgs/gamer-collection-puc/blob/main/prototipo-figma.mp4

Vídeo do protótipo html:

https://github.com/filipborgs/gamer-collection-puc/blob/main/prototipo-vue.mp4

5. Diagrama de Classes de Domínio

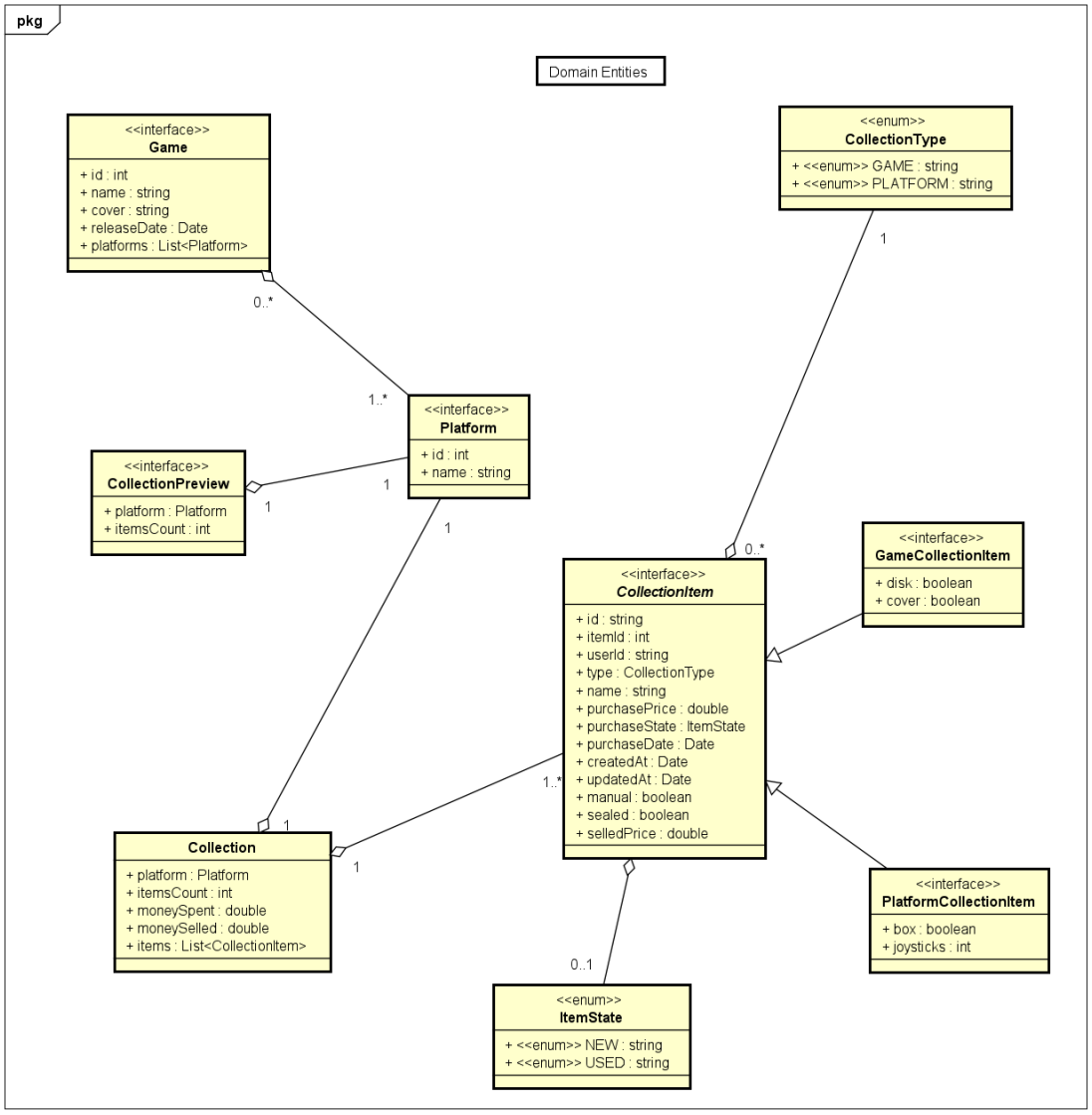
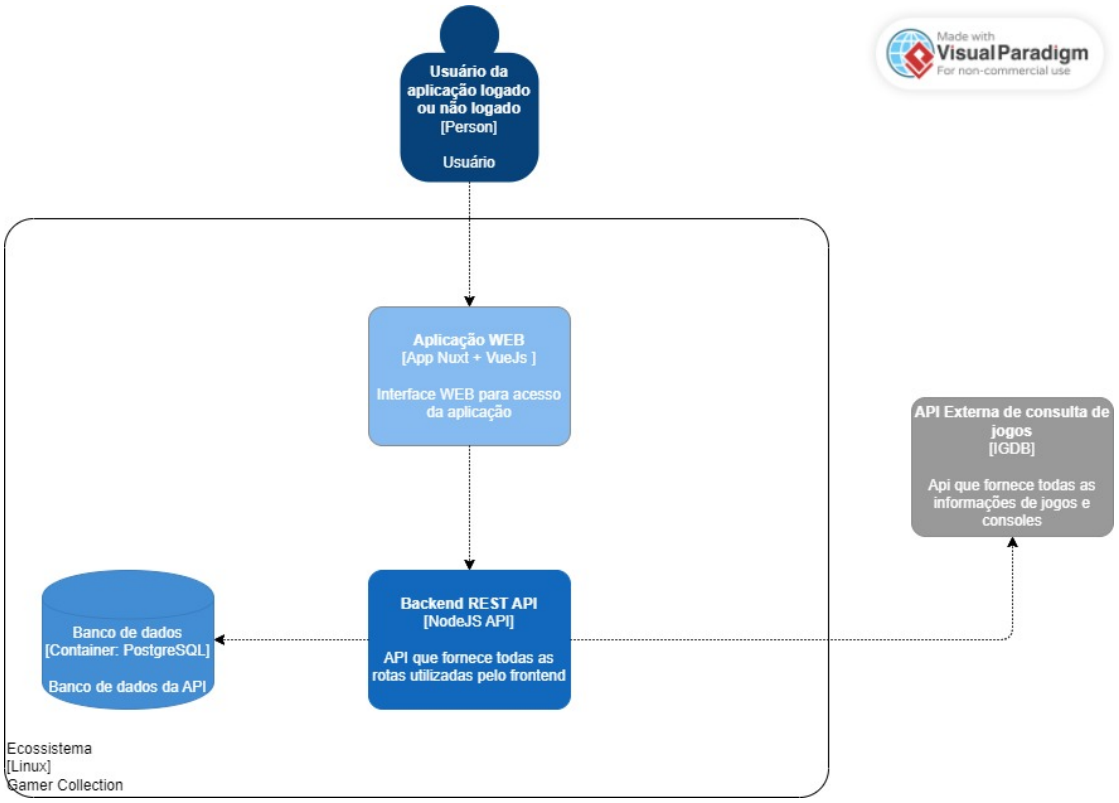


Imagem:
https://github.com/filipborgs/gamer-collection-puc/blob/main/docs/Class%20Diagram-v2.png

6. Padrão arquitetural

O padrão arquitetural escolhido para este projeto foi a Clean Architecture (Arquitetura limpa) para fazer a distribuição de responsabilidades em camadas, sempre seguindo os princípios do SOLID e, sempre que for necessário, aplicar soluções utilizando Design Patterns, como o factory, adapter, decorator, entre outros.

7. C4 model - Diagrama de Contexto



O usuário pode acessar a aplicação através de uma interface WEB. Ela pode ser utilizada a partir de um navegador de internet, em um dispositivo mobile ou de um computador. Esta usa como base os frameworks Nuxt e VueJS e se comunica com uma API desenvolvida em Javascript/Typescript (NodeJs). Para que a API interna conheça os jogos e consoles, a mesma é alimentada por uma API externa, chamada IGDB, que fornece todas as informações necessárias para cada item da coleção. E para o armazenamento de todos os dados da aplicação, é utilizado um banco PostgreSQL.

8. Tecnologias e Frameworks de Trabalho

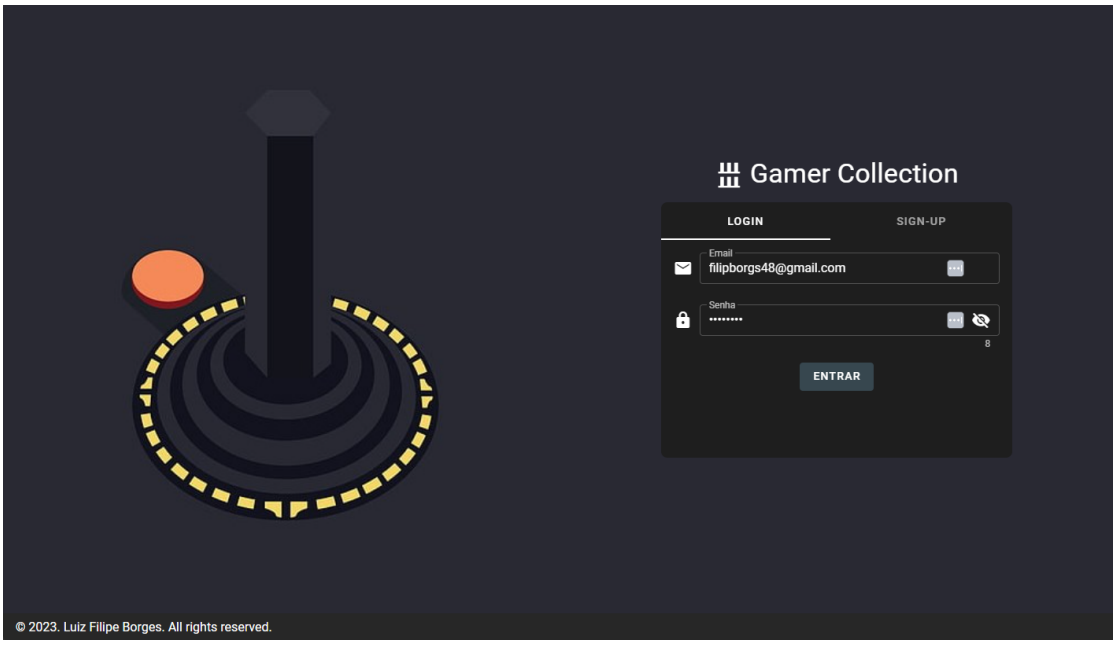
Tecnologias utilizadas:

- Backend: Para o desenvolvimento da API foi escolhido a linguagem Javascript e o runtime NodeJs, por ser bastante popular e ter um amplo suporte pela comunidade. Para adicionar uma camada de confiabilidade ao código e adicionar mais funcionalidades à linguagem, foi utilizado o superset Typescript. Para gerenciar o sistemas de rotas

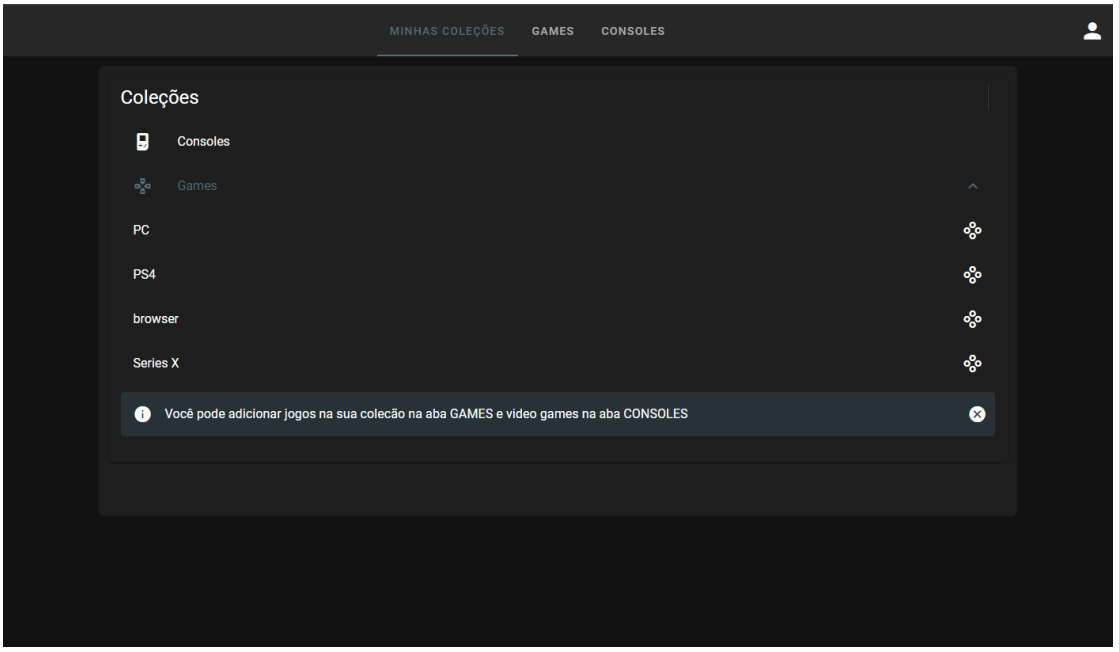
da aplicação foi utilizado o framework Express por ser limpo, objetivo, e ter um vasto suporte da comunidade. O banco de dados e ORM foram o Postgres, um banco extremamente robusto e confiável, e o TypeORM por trazer suporte nativo ao Typescript. E para testes foi utilizado a biblioteca Jest.

- Frontend: Como framework base foi utilizado o VueJs que é leve e de fácil utilização para os desenvolvedores. Para trazer mais features ao VueJs e facilitar toda a parte de configurações foi adicionado o Nuxt. E como biblioteca de componentes, buscou-se empregar o Vuetify, que utiliza o padrão material design, trazendo todos os componentes necessários nele sem precisar adicionar outras bibliotecas.
- Infra: Para hospedagem foram escolhidas plataformas com planos gratuitos, confiáveis e fáceis de serem utilizadas. Tais como:
    - Cyclic (Para o backend)
    - Netlify (Para o frontend)
    - ElephantSQL (Para o banco de dados)

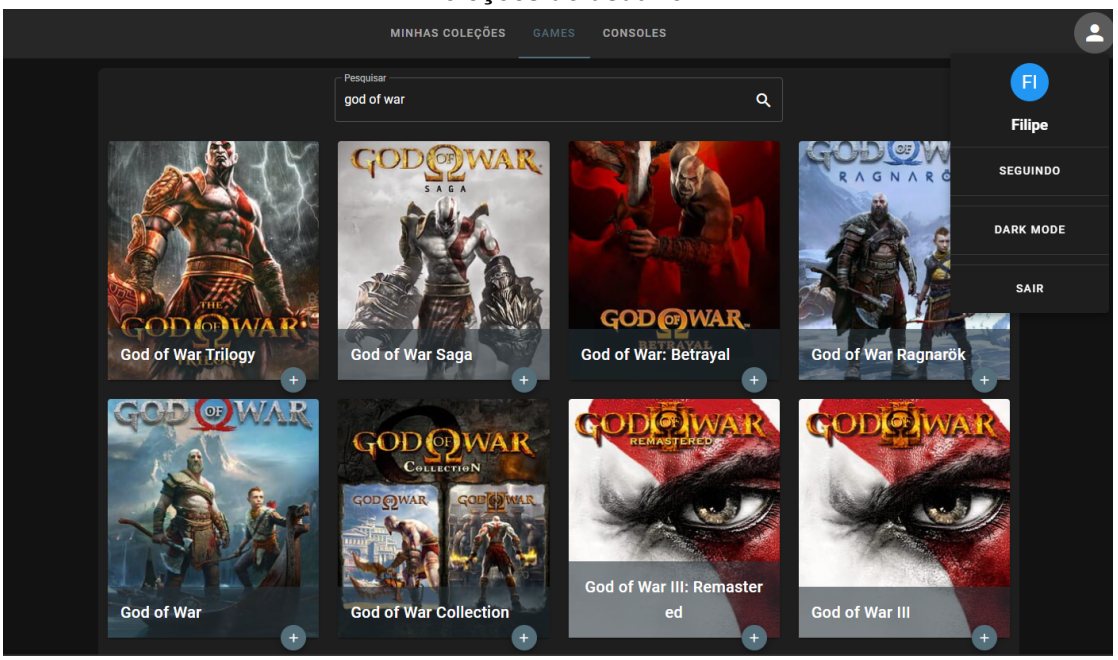
9. *Estrutura base do frontend*



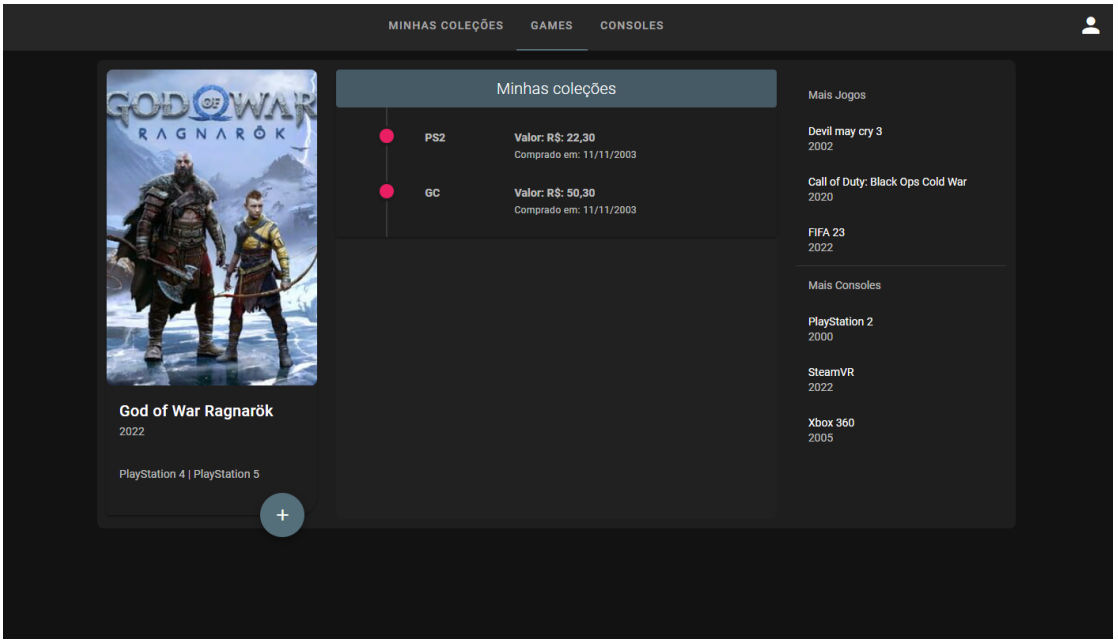
Tela de Login



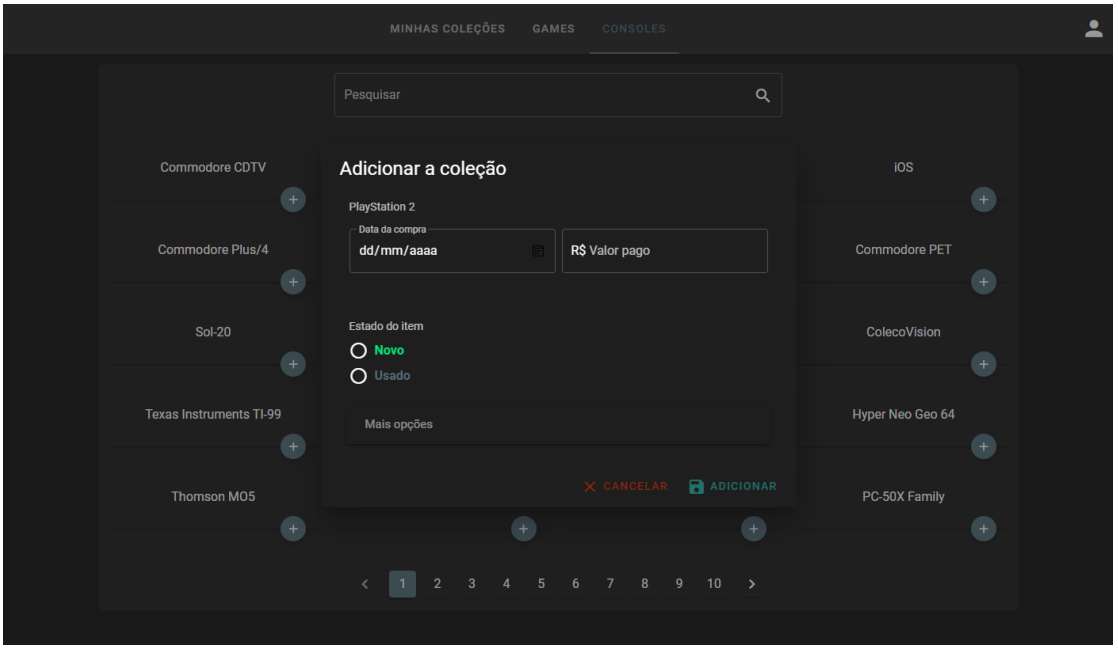
Coleções do usuário



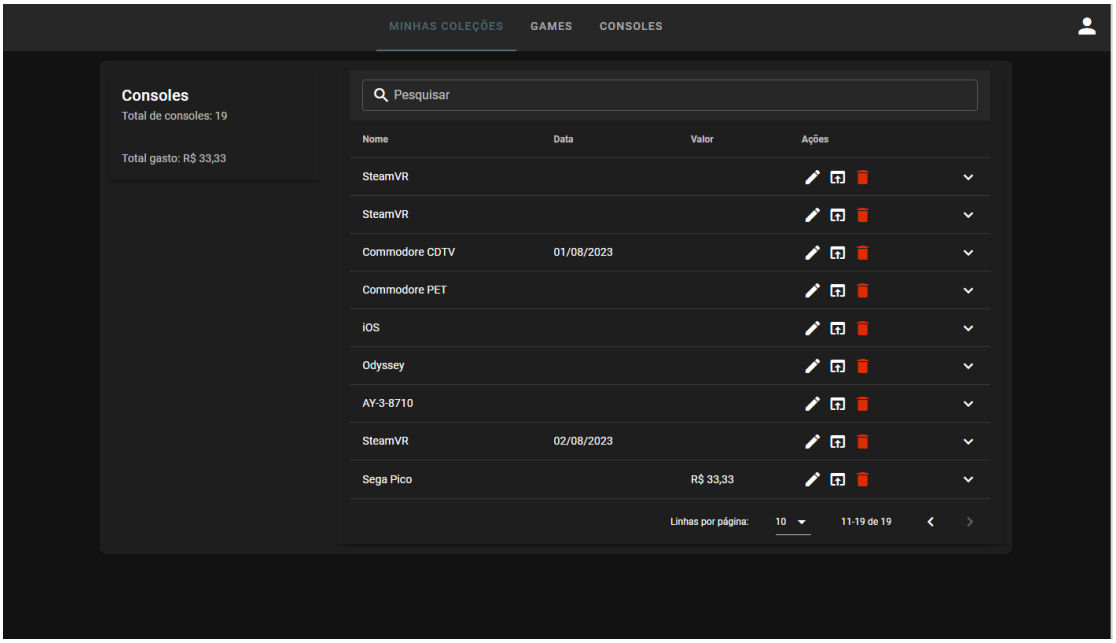
Listagem de jogos e menu de usuário



Tela com detalhes do jogo

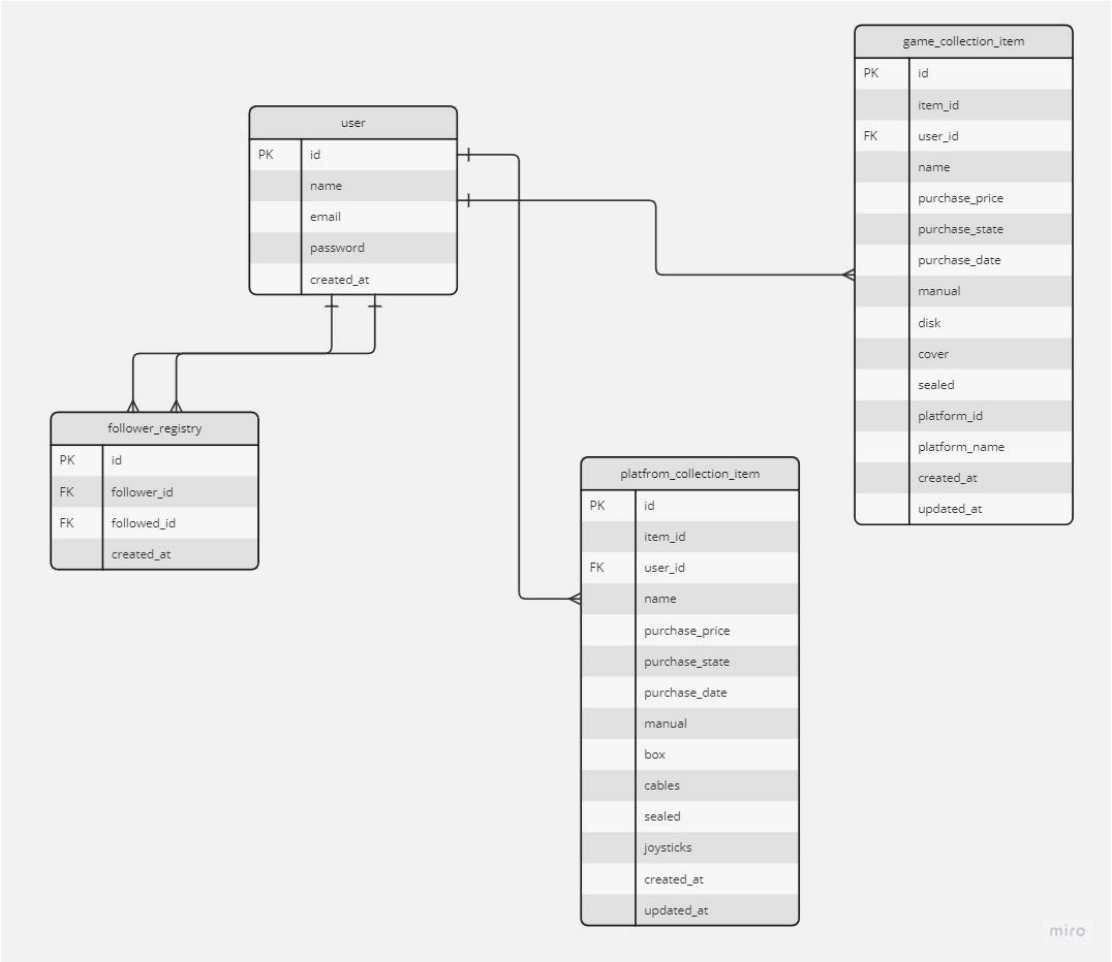


Modal de cadastro de consoles à coleção



Tela com os consoles cadastrados na coleção do usuário

10. *Modelo Relacional*



11. Plano de testes

N°	Caso de uso	Objetivo do teste	Entrada	Resultado esperado
1	Cadastrar um jogo a coleção	Adicionar um jogo a uma coleção	Preencher todos os dados	O sistema deverá exibir a mensagem de êxito e o jogo deve ser adicionado a coleção
	Cadastrar um jogo repetido a coleção	Adicionar múltiplas cópias de um jogo a coleção	Preencher todos os dados	O sistema deverá exibir a mensagem de êxito e o jogo deve ser adicionado a coleção
	Cadastrar um jogo a coleção sem selecionar a plataforma	Adicionar um jogo a coleção sem plataforma	Preencher todos os dados, exceto a seleção de plataforma	O sistema deverá mostrar uma mensagem de erro informando que a plataforma é obrigatória
	Cadastrar um jogo a coleção com dados inválidos	Adicionar um console a coleção com dados inválidos	Preencher todos os dados e informar a data de compra maior que a atual	O sistema deverá mostrar uma mensagem de erro informando que a data de compra deve ser menor ou igual que a atual
2	Cadastrar um console a coleção	Adicionar um console a coleção	Preencher todos os dados	O sistema deverá exibir a mensagem de êxito e o console deve ser adicionado a coleção
	Cadastrar um console repetido a coleção	Adicionar múltiplas cópias de um console a coleção	Preencher todos os dados	O sistema deverá exibir a mensagem de êxito e o console deve ser adicionado a coleção
	Cadastrar um console a coleção com dados inválidos	Adicionar um console a coleção com	Preencher todos os dados e informar a	O sistema deverá mostrar uma mensagem

		dados inválidos	data de compra maior que a atual	de erro informando que a data de compra deve ser menor ou igual que a atual
3	Realizar pesquisa de jogos	Obter jogos para que possam ser adicionados às coleções	Preencher campo de pesquisa com o nome do jogo	O sistema deverá exibir uma lista de jogos com o nome que foi pesquisado
	Realizar pesquisa de jogos que não existem ou não se encontra na base	Obter jogos para que possam ser adicionados às coleções	Preencher campo de pesquisa com o nome do jogo	O sistema deverá exibir uma mensagem dizendo que o jogo não foi encontrado
4	Realizar pesquisa de consoles	Obter consoles para que possam ser adicionados às coleções	Preencher campo de pesquisa com o nome do console	O sistema deverá exibir uma lista de consoles com o nome que foi pesquisado
	Realizar pesquisa de consoles que não existem ou não se encontra na base	Obter consoles para que possam ser adicionados às coleções	Preencher campo de pesquisa com o nome do console	O sistema deverá exibir uma mensagem dizendo que o console não foi encontrado

12. Apropriação de Horas no Projeto

Histórico de apropriação de horas		
Data do registro	**Atividade**	**Quantidade de horas**
20 / 01 / 23	Revisão dos materiais.	16
25 / 01 / 23	Definição da solução.	18
03 / 02 / 23	Testes de integração com APIs de terceiros.	22
08 / 02 / 23	Objetivos do Trabalho, Apresentação do Problema, Descrição do Software, Cronograma, Definição dos requisitos.	20
14 / 02 / 23	Identificação de atores, requisitos funcionais e não funcionais.	10
28 / 02 / 23	Desenvolvimento do diagrama de casos de uso e seus descritivos.	14
05 / 03 / 23	Protótipo da interface do produto.	25

12 / 03 / 23	Desenvolvimento do diagrama de classes.	10
15 / 03 / 23	Finalização e revisão da primeira etapa.	8
04 / 04 / 23	Definição do padrão arquitetural e tecnologias do projeto	20
10 / 04 / 23	Criação do diagrama de contexto	6
25 / 04 / 23	Criação do modelo relacional do banco de dados	18
31 / 05 / 23	Desenvolvimento da base do backend	60
30 / 06 / 23	Desenvolvimento da base do frontend e integrações com o backend	55
31 / 07 / 23	Refinamento de toda a aplicação e deploys	50
15 / 08 / 23	Registro no relatório sobre tecnologias, frameworks, imagens e links do projeto	20
14 / 09 / 23	Registro do plano de testes do projeto	18
14 / 10 / 23	Finalização do relatório e revisões	25

13. Código da Aplicação

Frontend: https://gamer-collection.netlify.app/

Backend: https://gamer-api.cyclic.app/api

Usuário padrão:

- Email: filipborgs48@gmail.com
- Senha: password

Github da Aplicação: https://github.com/filipborgs/gamer-collection-puc

Obs: Ainda estou trabalhando no desenvolvimento da Aplicação, então possíveis mudanças podem acontecer.

Vídeo guia utilizando a aplicação:

https://github.com/filipborgs/gamer-collection-puc/blob/main/docs/fluxo.mp4

14. Avaliação retrospectiva

Durante o desenvolvimento do projeto, pude colocar em prática todo meu conhecimento em desenvolvimento de softwares e tudo aquilo que foi adquirido durante as matérias do curso. Como por exemplo, a priorização de tarefas e o gerenciamento de tempo aprendidos na matéria de Métodos ágeis, questões de cores e usabilidades que pude aprender na matéria de design e experiência do usuário e/ou também os diagramas vistos em aulas diversas, principalmente na de Projeto (Design) de Software.

14.1 Objetivos Estimados

Esse projeto teve como objetivo o desenvolvimento de um sistema que melhor atendesse uma necessidade minha e de diversos colecionadores que passam pela dificuldade de gerenciar suas coleções. Bem como, a disponibilidade de uma base de jogos já cadastrados no sistema, evitando o usuário de ter que adicionar os jogos e consoles manualmente.

14.2 Objetivos Alcançados

Tanto meus objetivos pessoais, quanto os propostos no desenvolvimento do projeto foram alcançados com êxito. Todas as funcionalidades principais foram gradativamente entregues e eu planejo continuar expandindo o projeto após a conclusão do curso.

14.3 Lições aprendidas

	Descrição	**Classificação**
1	Aplicação de tempo aos cronogramas	Positiva
2	Conhecimento adquirido durante o projeto	Positiva
3	Aplicação de conceitos teóricos ao projeto	Positiva