

**Pixel Mart: Trabalho Integrado dos Componentes de Banco de Dados II, Engenharia de Software I e Programação II**

Diego Rafael Muller\*

Guilherme M. De Souza\*\*

Luis R. Holscher\*\*\*

Otilia Donato Barbosa\*\*\*\*

Thiago H. Venturini\*\*\*\*\*

Thiago J. De O. Scaravonatto\*\*\*\*\*

Roberson Junior Fernandes Alves\*\*\*\*\*

Rubens G. Voivoda\*\*\*\*\*

---

\* Discente do Curso de Ciência da Computação  
Unoesc - Campus de São Miguel do Oeste  
Rua Oiapoc, 211. São Miguel do Oeste-SC  
diegomuller089@gmail.com

\*\* Discente do Curso de Ciência da Computação  
Unoesc - Campus de São Miguel do Oeste  
Rua Oiapoc, 211. São Miguel do Oeste-SC  
gmassinhani@gmail.com

\*\*\* Discente do Curso de Ciência da Computação  
Unoesc - Campus de São Miguel do Oeste  
Rua Oiapoc, 211. São Miguel do Oeste-SC  
luisricardoholscher@gmail.com

\*\*\*\* Mestre em Engenharia  
Biomédia e Informática Industrial  
Docente do Curso de Ciência da  
ComputaçãoUnoesc - Campus de São  
Miguel do Oeste Rua Oiapoc, 211.  
São Miguel do Oeste-SC  
otilia.barbosa@unoesc.edu.br

\*\*\*\*\* Discente do Curso de Ciência da Computação  
Unoesc - Campus de São Miguel do Oeste  
Rua Oiapoc, 211. São Miguel do Oeste-SC  
thethiagoveturini@gmail.com

\*\*\*\*\* Discente do Curso de Ciência da Computação  
Unoesc - Campus de São Miguel do Oeste  
Rua Oiapoc, 211. São Miguel do Oeste-SC  
thiago.ctecnico@gmail.com

\*\*\*\*\* Mestre em Computação Aplicada (UEPG)  
Professor do Curso de Bacharelado em Ciência da  
Computação Unoesc - Campus de São Miguel do  
Oeste Rua Oiapoc, 211 – São Miguel do Oeste – SC  
roberson.alves@unoesc.edu.br

\*\*\*\*\*Discente do Curso de Ciência da Computação  
Unoesc - Campus de São Miguel do Oeste  
Rua Oiapoc, 211. São Miguel do Oeste-SC  
rubensgvgarciavoivoda@gmail.com

## Resumo

Este artigo apresenta um estudo de caso sobre a implementação de um Sistema de Gerenciamento de Banco de Dados (SGBD) em uma loja de jogos, com o objetivo de modernizar e otimizar os processos administrativos do estabelecimento. A proposta visa integrar funcionalidades essenciais para o gerenciamento do negócio, incluindo o cadastro de clientes, controle de produtos e gerenciamento de pagamentos. Além de resolver as limitações de sistemas manuais ou desatualizados, o projeto busca criar uma solução tecnológica escalável e eficiente, adaptada às necessidades específicas de uma loja de jogos. Para alcançar esses objetivos, foram utilizadas ferramentas modernas, como PostgreSQL para a criação e gestão do banco de dados, Eclipse e Spring Boot para o desenvolvimento do backend, e Visual Paradigm para a modelagem de dados e processos. O planejamento e a organização das tarefas foram realizados por meio do Trello, enquanto o controle de versão foi feito com o GitHub, garantindo a rastreabilidade e a integridade do código ao longo do desenvolvimento. Durante o desenvolvimento, foram aplicadas práticas de engenharia de software, como a normalização do banco de dados, a definição de chaves primárias e estrangeiras, e a utilização de cláusulas SQL avançadas, como JOINS e views, para melhorar o desempenho das consultas. Scripts de backup e triggers foram implementados para aumentar a segurança e a funcionalidade do sistema, enquanto testes contínuos garantiram a consistência e a confiabilidade da aplicação. O resultado foi a criação de um sistema integrado, capaz de atender às principais demandas de gestão da loja de jogos, simplificando tarefas administrativas e proporcionando uma experiência eficiente para os usuários. Este projeto destaca não apenas a importância do uso de tecnologias modernas, mas também a relevância do trabalho em equipe e da aplicação de metodologias sólidas para alcançar soluções robustas e adaptáveis. A conclusão reforça como a implementação de um SGBD bem projetado pode transformar processos de negócios, promovendo eficiência operacional e atendendo às expectativas do mercado.

**Palavras-chave:** Engenharia. Software. Programação. Dados. Informações. Loja de Jogos. SGBD. Implementação.

## 1 INTRODUÇÃO

Com o avanço tecnológico, diversas áreas têm se beneficiado da automação e otimização de processos, em que a eficiência na gestão de informações desempenha um papel crucial, a implementação de soluções tecnológicas torna-se essencial para o sucesso e reconhecimento de estabelecimentos especializados. Este artigo discute o desenvolvimento de um Sistema Gerenciador de Banco de Dados para o controle de uma loja de jogos. Neste contexto, surge a proposta de desenvolvimento do software "Pixel Mart", dedicado a atender às demandas específicas de uma loja de games.

O Pixel Mart, busca ir além da simples comercialização de produtos, representando um ambiente dinâmico onde a eficiência operacional é fundamental. O sistema proposto oferece uma solução integrada, otimizando processos como, cadastro de pessoas, controle de jogos e controle de pagamentos.

Assim, diante desse contexto, o artigo apresentará a abordagem do Pixel Mart, destacando sua proposta inovadora para a gestão integrada, desde a concepção inicial até os resultados alcançados no desenvolvimento do sistema.

Uma solução abrangente para otimizar a gestão da loja de jogos "Pixel Mart" seria a implementação de um sistema integrado que tem como foco aprimorar a gestão da loja de jogos, integrando funcionalidades como cadastro de clientes, gerenciamento de jogos e controle de pagamentos. Além disso, o sistema busca oferecer uma experiência fluida para os usuários, adaptando-se às necessidades do mercado e elevando a eficiência operacional da loja.

## 2 MATERIAIS E MÉTODOS

O problema abordado neste artigo é a falta de um sistema gerenciador de banco de dados eficiente e adequado às necessidades de uma loja de jogos. Atualmente, muitas lojas de jogos ainda dependem de processos manuais ou sistemas desatualizados para realizar tarefas como cadastro de clientes, gerenciamento de jogos e controle de pagamentos.

Para a realização deste trabalho utilizamos várias ferramentas, sendo elas o dBeaver para a criação da base de dados em PostgreSQL, o Visual Paradigm para realização modelagem do banco de dados, além disso foi utilizado também para confecção do modelo de caso de uso, diagrama de sequência, diagrama de atividade, diagrama de estado e diagrama de classe, no ambiente de programação foi utilizado o Eclipse, onde realizamos métodos de criação, exclusão, listagem, busca e atualização, além do fluxo de cadastro de usuário, foi utilizada a linguagem de programação Java, e a biblioteca Jakarta e o framework Springboot, utilizamos ainda o GitHub como ferramenta para o versionamento. Trello para organização de cartões e tarefas realizadas pelo grupo, e o Visual Paradigm para realização do modelo conceitual.

dBeaver: É uma ferramenta de administração e desenvolvimento de banco de dados de código aberto, que oferece uma interface intuitiva para executar consultas, gerenciar esquemas, tabelas e visualizar dados (PEDRO PINTO, 2017).

Visual Paradigm: É uma ferramenta de modelagem visual de dados e processos (VISUAL PARADIGM, 2023).

GitHub: É uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo (WIKIPÉDIA, 2022).

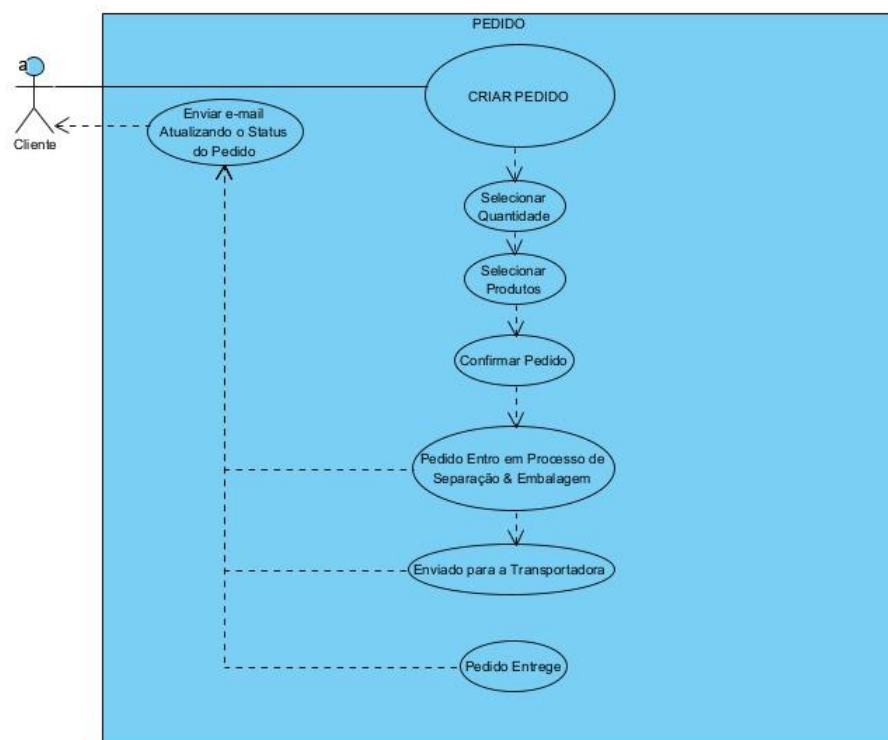
Trello: É uma ferramenta visual que possibilita ao time o gerenciamento de qualquer tipo de projeto, fluxo de trabalho ou monitoramento de tarefas. (TRELLO, 2023).

SpringBoot: É um framework bastante conhecido de nível empresarial, de software livre, para criar aplicativos independentes de nível de produção que são executados na Java Virtual Machine (JVM). (IBM, 2023)

## 2.1 MODELAGEM

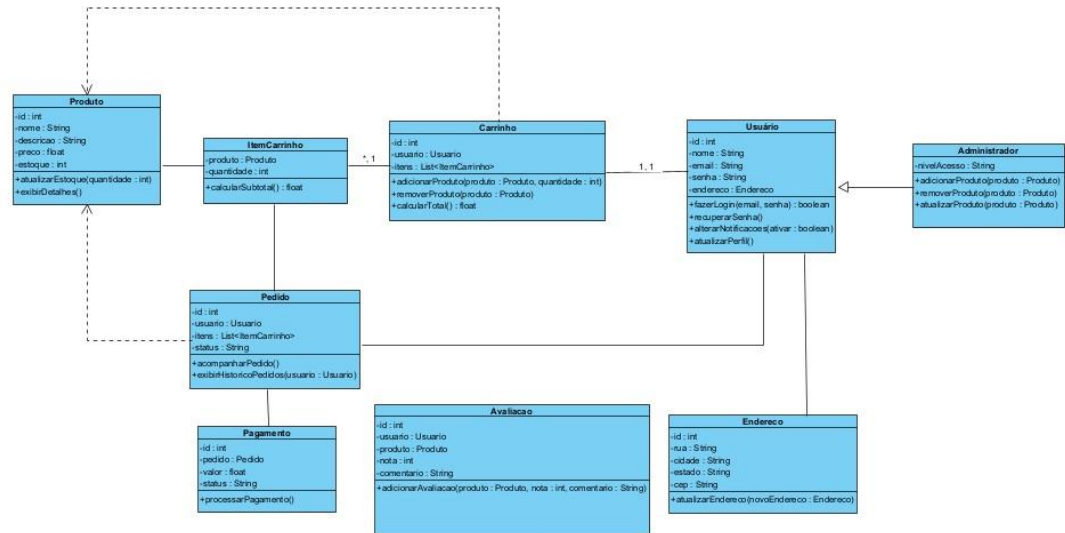
A partir dos requisitos já levantados, foi dado início a construção dos diagramas de casos de uso e de classes utilizando a ferramenta Visual Paradigm. Abaixo as Figuras 1 ilustra um diagrama de Caso de Uso, já a Figura 2 mostram o diagrama de classe de um Sistema de loja de jogos e a Figura 3 mostra o diagrama de atividades.

Figura 1 - Diagrama de Caso de Uso



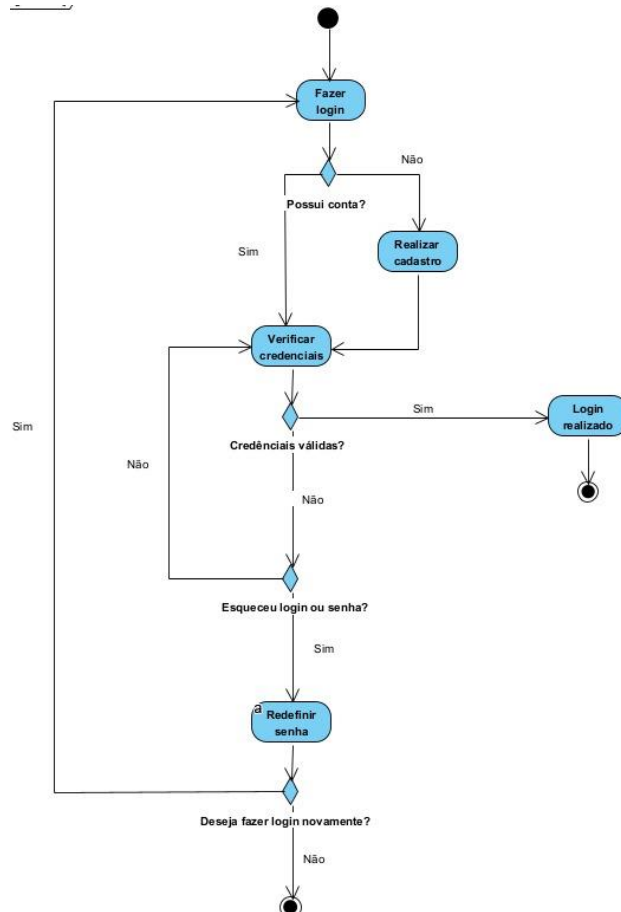
Fonte: Autoria própria (2024).

Figura 2 - Diagrama de Classes



Fonte: Autoria própria (2024).

Figura 3 - Diagrama de atividades

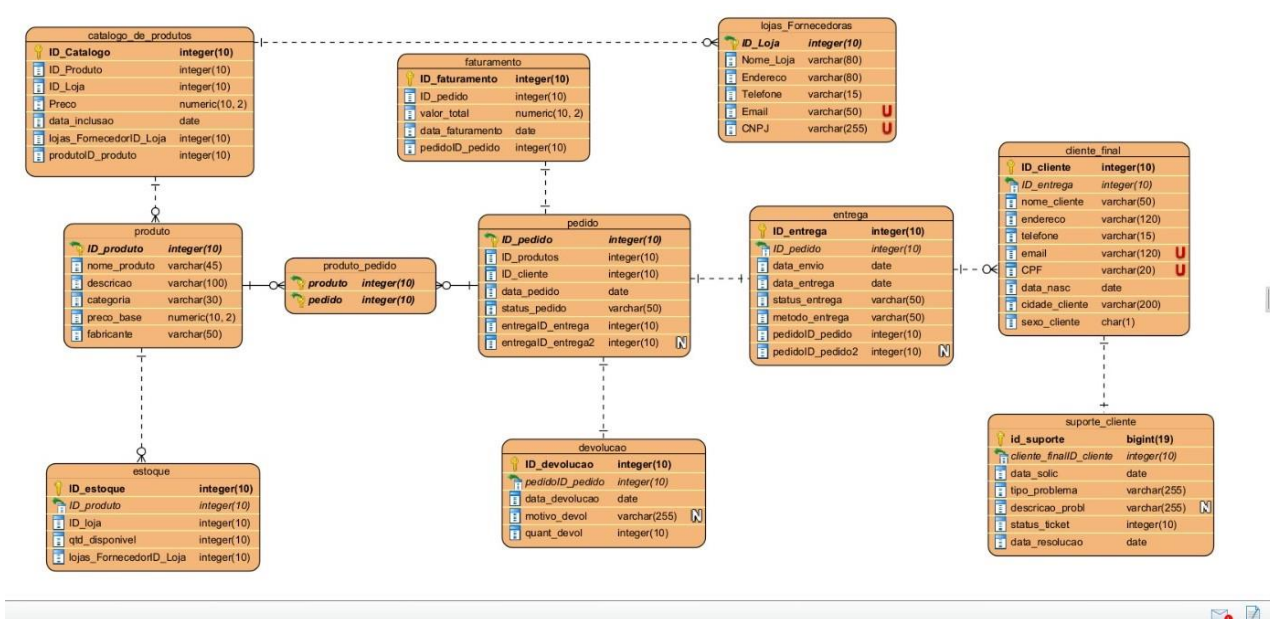


Fonte: Autoria própria (2024).

### 3 DESENVOLVIMENTO

Inicialmente reconstruímos nosso banco para que seu modelo fique normalizado. A Figura 3 ilustra a modelagem completa do banco.

Figura 4 - Banco de Dados da loja de jogos



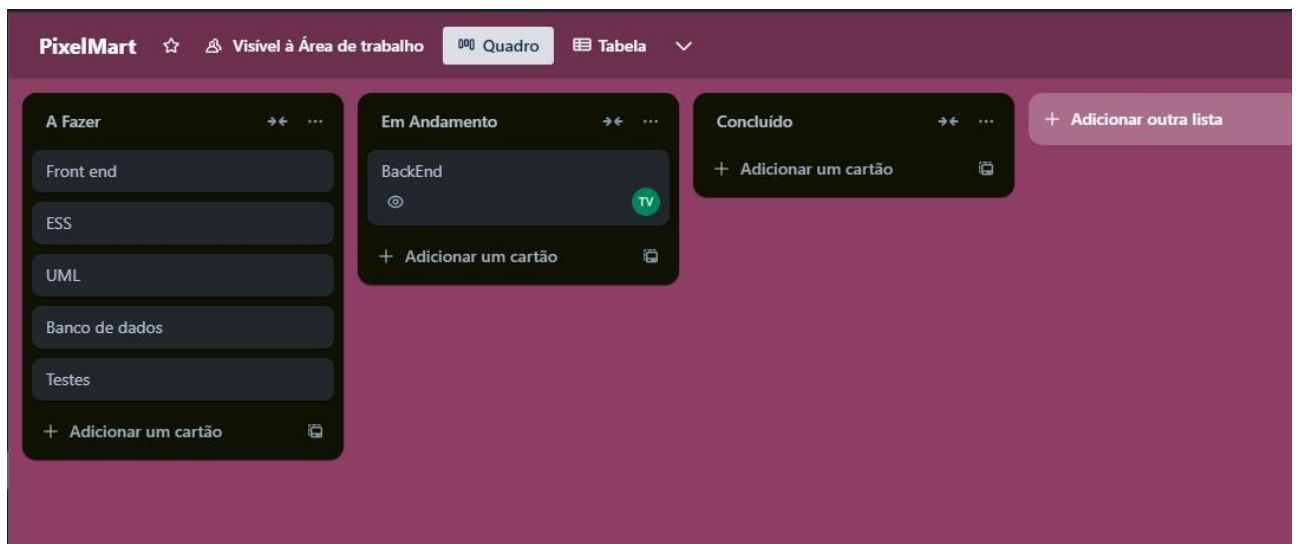
Fonte: Autoria própria (2024).

O desenvolvimento do banco de dados foi por meio do Dbeaver/PgAdmin, para a criação domesmo utilizamos a ferramenta de criação de script do VisualParadigm após a conclusão da modelagem, após os scripts serem revisados, realizamos a criação de tabelas, inserção, atualização e exclusão de registros, além de consultas complexas para recuperar e filtrar informações. Também foram utilizadas cláusulas SQL avançadas, como JOINS para combinardados de diferentes tabelas, além disso foram criados scripts de backup, criação de usuários e restrições, índices e views.

Para o desenvolvimento back-end usamos o Eclipse para escrita e desenvolvimento das funcionalidades do programa, por meio da linguagem de programação Java. Utilizamos como principais referências a biblioteca Jakarta e ainda utilizamos o framework Java Spring Boot que permitiu que os objetos tivessem suas próprias dependências, foi utilizado também para ligação de dados, conversão de tipo, validação, gerenciamento de exceções e gerenciamento de eventos e recursos, aumentando a flexibilidade, facilidade e deixando desenvolvimento mais rápido.

Posteriormente foi criado um KANBAN na ferramenta Trello, para organizar as tarefas e decidir quais atividades cada integrante ficou responsável. A Figura 4 mostra o KANBAN inicialmente em seus primeiros dias. É perceptível que no começo as atividades eram básicas e com o objetivo de organização do projeto. Somente mais tarde que as modelagens e implementações foram adicionadas como atividades como mostra a Figura 5 .

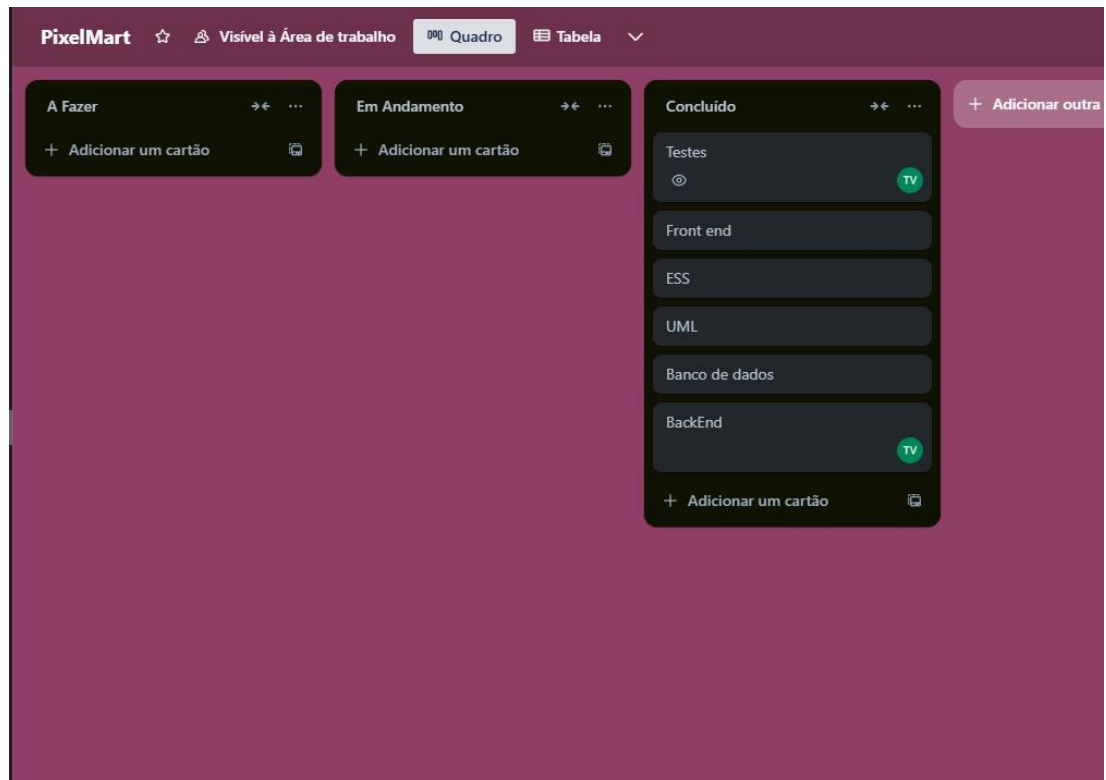
Figura 5 - Estado inicial do KANBAN



Fonte: Autoria própria (2024).



Figura 6 - KANBAN das atividades e estágio final



Fonte: Autoria própria (2024).

A organização de tarefas foi essencial para a equipe e para nosso projeto, e o uso de quadros visuais, inspirado no KANBAN, foi uma solução prática e eficiente. A estrutura clara das colunas e a divisão das tarefas permitiu uma visão geral do progresso, ajudando a manter o foco e facilitar a colaboração.

## 4 SISTEMA PIXEL MART

Com a modelagem concluída e tarefas já estabelecidas, foi dado início a parte de Banco de Dados, onde a estrutura principal já foi elaborada semestre passado, sendo acrescentados pequenos ajustes nas colunas, relatórios e script de inserts do banco de dados, além disso foram criados grupos e usuários com seus próprios privilégios, índices de colunas das principais tabelas do banco, views para facilitar pesquisa dos relatórios já estabelecidos, e implementado dois gatilhos, e criado políticas e configuração do backup e restore.

### 4.1 CÓDIGOS

Segue Figuras 7, 8 e 9 dos códigos realizados em Java para implementação dos códigos:

Figura 7 - SpringApplication

```
package br.com.PixelMart.projeto;

import org.springframework.boot.SpringApplication;

@EntityScan(basePackages = "br.com.PixelMart.projeto.model")
@SpringBootApplication
public class ProjetoApplication {

    public static void main(String[] args) {
        SpringApplication.run(ProjetoApplication.class, args);
    }

}
```

Fonte: Autoria própria (2024).

Figura 8 – LoginController

```
package br.com.PixelMart.projeto.controller;

import java.util.Optional;

@RestController
@CrossOrigin("*")
@RequestMapping("/auth")
public class LoginController {

    @Autowired
    private ILogins loginRepository;

    @PostMapping("/login")
    public ResponseEntity<Logins> login(@RequestBody Logins loginData) {
        Optional<Logins> login = loginRepository.findByUsuarioAndSenha(loginData.getUsuario(), loginData.getSenha());

        if (login.isPresent()) {
            return ResponseEntity.ok(login.get());
        } else {
            return ResponseEntity.status(HttpStatus.UNAUTHORIZED).build();
        }
    }
}
```

Fonte: Autoria própria (2024).

Figura 9 – DAO

```
package br.com.PixelMart.projeto.DAO;

import java.util.Optional;

public interface ILogins extends JpaRepository<Logins, Integer> {
    Optional<Logins> findByUsuarioAndSenha(String usuario, String senha);
}
```

Fonte: Autoria própria (2024).

## **5 CONCLUSÃO**

O desenvolvimento do Pixel Mart demonstra a importância de um Sistema Gerenciador de Banco de Dados (SGBD) eficiente para otimizar a gestão de lojas de jogos. Com o uso de ferramentas modernas e metodologias ágeis, foi possível criar um sistema robusto e adaptável às necessidades do mercado. A modelagem cuidadosa e a implementação detalhada permitiram que o sistema integrasse funções essenciais, como cadastro de clientes, controle de produtos e gerenciamento de pagamentos, além de garantir a escalabilidade e a usabilidade. Este projeto não apenas aprimorou os processos internos da loja, mas também destacou o papel do trabalho em equipe e do domínio técnico na entrega de soluções tecnológicas inovadoras.

## *Pixel Mart (DBMS)*

### *Abstract*

This article presents a case study on the implementation of a Database Management System (DBMS) in a game store, with the goal of modernizing and optimizing the establishment's administrative processes. The proposal aims to integrate essential functionalities for business management, including customer registration, product control, and payment management. In addition to addressing the limitations of manual or outdated systems, the project seeks to create a scalable and efficient technological solution tailored to the specific needs of a game store. To achieve these objectives, modern tools such as PostgreSQL for database creation and management, Eclipse and Spring Boot for backend development, and Visual Paradigm for data and process modeling were used. Task planning and organization were carried out through Trello, while version control was managed using GitHub, ensuring traceability and code integrity throughout development. During the development process, software engineering practices were applied, such as database normalization, the definition of primary and foreign keys, and the use of advanced SQL clauses like JOINS and views to enhance query performance. Backup scripts and triggers were implemented to increase the security and functionality of the system, while continuous testing ensured the application's consistency and reliability. The result was the creation of an integrated system capable of meeting the main management demands of the game store, simplifying administrative tasks and providing an efficient user experience. This project highlights not only the importance of using modern technologies but also the relevance of teamwork and the application of solid methodologies to achieve robust and adaptable solutions. The conclusion emphasizes how the implementation of a well-designed DBMS can transform business processes, promoting operational efficiency and meeting market expectations.

*Keywords: Engineering, Software, Programming, Data, Information, Game Store, DBMS, Implementation.*

## REFERÊNCIAS

ALVEZ, Roberson J. F. **Apostila de Banco de Dados**. São Miguel do Oeste: Unoesc, 2023. Material didático em PDF.

Autor Desconhecido: **Guia de tipos de diagramas UML: aprenda sobre todos os tipos de diagramas UML com exemplos**. Disponível em: <https://creately.com/blog/pt/diagrama/guia-de-tipos-de-diagramas-uml-aprenda-sobre-todos-os-tipos-de-diagramas-uml-com-exemplos/>. Acesso em: 17 nov, 2023.

BARBOSA, Otilia D. **Anotações, JavaDoc e JUnit**. São Miguel do Oeste: Unoesc, 2023. Material didático em PDF.

BARBOSA, Otilia D. **Spring Boot API**. São Miguel do Oeste: Unoesc, 2023. Material didático em PDF.

**CASOS DE USO**. 2023. 34 slides. Apresentação de slides.

CATUNDA, Heitor. **TRIGGER EM SQL: APLICAÇÕES, EXEMPLOS E BOAS**

IBM. **API (JPA)**. Disponível em: [https://www.ibm.com/docs/pt-br/was/8.5.5?topic=SSEQTP\\_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/cejbpersistence.htm](https://www.ibm.com/docs/pt-br/was/8.5.5?topic=SSEQTP_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/cejbpersistence.htm). Acesso em: 14 nov, 2023.

IBM. **O que é Java Spring Boot?**. Disponível em: <https://www.ibm.com/br-pt/topics/java-spring-boot>. Acesso em: 07 nov, 2023.

LISBOA, Paulo. **Como Criar um Banco de Dados Em Java**. Disponível em: [https://awari.com.br/como-criar-um-banco-de-dados-em-java/?utm\\_source=blog&utm\\_campaign=projeto+blog&utm\\_medium=Como%20Criar%20um%20Banco%20de%20Dados%20Em%20Java](https://awari.com.br/como-criar-um-banco-de-dados-em-java/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Como%20Criar%20um%20Banco%20de%20Dados%20Em%20Java). Acesso em: 17 nov, 2023.

MARTINS, C. et al. **O MERCADO DE JOGOS ELETRÔNICOS E SEUS IMPACTOS NA SOCIEDADE**. Disponível em: <https://multivix.edu.br/wp-content/uploads/2018/08/o-mercado-de-jogos-eletronicos-e-seus-impactos-na-sociedade.pdf>.

PETRY, Franciele C. **Modelagem UML: DIAGRAMA DE CLASSES**. 2023. 26 slides. Apresentação de slides.

PostgreSQL. **Documentation**. Disponível em: <https://www.postgresql.org/docs/current/sql-createview.html>. Acesso em: 09 nov, 2023.

**PRÁTICAS**. Disponível em: <https://www.hashtagtreinamentos.com/trigger-em-sql>. Acesso em: 24 nov, 2023.

SELEM, Cristiane. **Engenharia de Software em Destaque: Diagramas Comportamentais da UML**. Disponível em: <https://www.estrategiaconcursos.com.br/blog/engenharia-software-diagramas-comportamentais-uml/>. Acesso em: 23 nov, 2023.

VENTURA, Plinio. **Entendendo definitivamente o que é um Caso de Uso**. Disponível em: <https://www.ateomomento.com.br/o-que-e-caso-de-uso/>. Acesso em: 18 nov, 2023.