

FACULDADE DE TECNOLOGIA DE MOGI DAS CRUZES

ANDRÉ LUIS COSME SANTOS

ISRAEL ANDRELINO CURVELO

LUCIO KENJI NAKA

**RAK GAMESHOP - DESENVOLVIMENTO DE
SISTEMA PARA VENDA DE MÍDIAS
DIGITAIS**

**Mogi das Cruzes – SP
Junho - 2022**

FACULDADE DE TECNOLOGIA DE MOGI DAS CRUZES

ANDRÉ LUIS COSME SANTOS

ISRAEL ANDRELINO CURVELO

LUCIO KENJI NAKA

**RAK GAMESHOP - DESENVOLVIMENTO DE
SISTEMA PARA VENDA DE MÍDIAS
DIGITAIS**

Relatório Técnico-Científico
apresentado à FATEC Mogi das
Cruzes como parte das
exigências do Curso Superior de
Tecnologia em Análise e
Desenvolvimento de Sistemas
para a obtenção do título de
Tecnólogo.

Orientador: Prof. Fabio Codo

Área de Concentração: Sistemas de
Informação

Linha de estudo: Programação Web

**Mogi das Cruzes – SP
Junho - 2022**

CURVELO, Israel Andreolino; NAKA, Lucio Kenji; SANTOS, André Luis Cosme. RAKGAMESHOP: Desenvolvimento de sistema para venda de mídias digitais. Mogi das Cruzes, 2022. 26 páginas. Relatório Técnico-Científico - Trabalho de Graduação, Faculdade de Tecnologia de Mogi das Cruzes.

RESUMO

Segundo dados de pesquisas recentes, muitas empresas estão migrando para o modelo de vendas online, não diferente disso a empresa RAK GAMES decidiu criar uma plataforma de vendas de produtos digitais, utilizando ferramentas populares no mercado de tecnologia, com um design simplista e intuitivo. Para o desenvolvimento dessa plataforma foi necessário levantar os requisitos e regras de negócio, definição do escopo, elaboração do documento de visão de projeto, elaboração de design e criação de protótipo, codificação das validações e persistências e da parte visual do sistema. Levantados os requisitos, foram definidas as tecnologias a serem utilizadas, para o protótipo foi utilizado o software Figma, para o desenvolvimento do módulo lógico do sistema foi utilizada a linguagem C# com o auxílio do framework .NET, utilizando o ASPNET CORE MVC, persistindo os dados no banco relacional MySQL e para a codificação da parte visual do sistema, foi utilizado HMTL e CSS, com o auxílio dos frameworks Bootstrap e VUE.js.

Palavras-chave: RAK GAMES, Desenvolvimento, Sistema, Tecnologia.

CURVELO, Israel Andrelino; NAKA, Lucio Kenji; SANTOS, André Luis Cosme. RAKGAMESHOP: Development of a system for digital media sales. Mogi das Cruzes, 2022. 26 pages. Technical-Scientific Report - Graduation Work, Faculty of Technology of Mogi das Cruzes.

ABSTRACT

According to recent research data, many companies are migrating to the online sales model, not unlike that the company RAK GAMES decided to create a digital product sales platform, using popular tools in the technology market, with a simplistic and intuitive design. For the development of this platform, it was necessary to survey the requirements and business rules, definition of the scope, elaboration of the project vision document, elaboration of design and creation of prototype, coding of validations and persistence and of the visual part of the system. Having raised the requirements, the technologies to be used were defined, Figma software was used for the prototype, for the development of the system's logical module, the C# language was used with the help of the .NET framework, using the ASPNET CORE MVC, persisting the data in the MySQL relational database and for the coding of the visual part of the system, HMTL and CSS were used, with the help of Bootstrap and VUE.js frameworks.

Keywords: RAK GAMES, Development, System, Technology.

SUMÁRIO

INTRODUÇÃO	7
1.1. APRESENTAÇÃO DO PROBLEMA EM ESTUDO	7
1.2. OBJETIVOS.....	8
1.2.1. Objetivos Gerais	8
1.2.2. Objetivos Específicos	8
2. FUNDAMENTAÇÃO TEÓRICA.....	9
2.1. HTML e CSS	9
2.2. HTTP E SISTEMAS EM AMBIENTE WEB.....	9
2.3. PADRÃO DE PROJETO COM ARQUITETURA REST API.....	9
2.4. BANCO DE DADOS RELACIONAL	11
3. METODOLOGIA.....	12
4. DESENVOLVIMENTO DO TEMA PROPOSTO	13
4.1. TECNOLOGIA UTILIZADA	13
4.1.1. .NET-CORE 2.1.....	13
4.1.2. ASP.NET CORE MVC.....	13
4.1.3. VUE JS	14
4.1.4. MySQL.....	14
4.2. ARQUITETURA	14
4.2.1. Representação Arquitetural.....	14
4.2.2. Restrições Arquiteturais	15
4.2.3. Objetivos e Restrições Arquiteturais	16
5. RESULTADOS.....	17
CONCLUSÕES	18

REFERÊNCIAS.....	19
ANEXOS	20

INTRODUÇÃO

Este documento tem como objetivo expor a necessidade de negócios e justificar o projeto, dando entendimento das necessidades do cliente e descrição de como o **RAK Game Shop** irá satisfazer esses requisitos.

Tendo como objetivo alinhar as expectativas dos interessados para formalizar o início do projeto, apresentando uma visão arquitetural de um módulo que atenda todas as necessidades de uma loja de jogos digitais abrangendo diversas plataformas, sendo este módulo uma parte de um sistema maior.

1.1. APRESENTAÇÃO DO PROBLEMA EM ESTUDO

A empresa RAK GAMES é uma empresa especializada em venda de jogos, porém com todo o cenário atual da COVID-19 e pandemia, ela resolveu se adaptar ao modelo virtual de negócio, vendendo atualmente apenas mídias digitais, evitando assim o contato físico com o cliente e a propagação do vírus.

Sendo assim, surgiu a necessidade de criar uma plataforma para disponibilizar seus jogos, solicitando o desenvolvimento de um sistema informatizado para venda de jogos digitais onde é necessário para que a empresa consiga ter total controle de todos os jogos que ela possui. Com o sistema, também é possível controlar todas as vendas que são efetuadas e a partir dos dados armazenados referentes as operações realizadas pela loja gere relatórios que serão utilizados para o controle interno de rentabilidade e faturamento.

1.2. OBJETIVOS

1.2.1. Objetivos Gerais

Conceber e construir um MVP (Mínimo produto viável) para a empresa RAK Games possibilitando adequar-se as formas do cenário atual de vendas com menor contato físico, maximizando o lucro e mitigando falência de pequenos negócios.

1.2.2. Objetivos Específicos

Desenvolver um sistema online de vendas de mídias digitais que atenda às necessidades da empresa RAK GAMES, visando a sustentabilidade do negócio dado a grande queda de vendas de mídias físicas de jogos.

2. FUNDAMENTAÇÃO TEÓRICA

Um sistema informatizado para venda de jogos digitais é necessário para que a instituição que utilizar esta solução consiga ter total controle de todos os jogos que ela possui. Com o sistema, também é possível controlar todas as vendas que são efetuadas e a partir dos dados armazenados referentes as operações realizadas pela loja, consiga gerar relatórios que serão utilizados para o controle interno de rentabilidade e faturamento. Dado o cenário, foram levantados os modelos de desenvolvimento a serem adotados no projeto, sendo eles:

2.1. HTML e CSS

HTML é a sigla em inglês para HyperText Markup Language, que, em português, significa linguagem para marcação de hipertexto, a qual se resume a todo conteúdo inserido em um documento web, que tem como principal função interligar outros documentos web, esses hipertextos são construídos com os links, que estão presentes na maioria das páginas que costumamos utilizar na internet. Já o CSS é a sigla para Cascading Style Sheet (Folhas de Estilo em Cascata), que se trata de um mecanismo para desenvolver de uma forma simples a definição de estilos, como: fonte, cores, tamanhos, espaçamentos, etc. Conforme **Maurício** (p. 25, 2008).

2.2. HTTP E SISTEMAS EM AMBIENTE WEB

HTTP do inglês Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto), é um protocolo que move massas de dados de forma rápida, conveniente e confiável de servidores da Web em todo o mundo para navegadores da Web na área de trabalho das pessoas. Conforme **Gourley** (p. 3, 2002).

2.3. PADRÃO DE PROJETO COM ARQUITETURA REST API

API REST (do inglês, API: Application Programming Interface) é uma interface de programação de aplicações em conformidade com as restrições da arquitetura REST (do inglês, REST: Representational State Transfer),

essa que por sua vez é um conjunto de restrições de arquitetura. Para que a API seja considerada do tipo REST, ela deve estar em conformidade com os seguintes critérios:

- Ter uma arquitetura cliente/servidor formada por clientes, servidores e recursos, com solicitações gerenciadas por HTTP.
- Estabelecer uma comunicação [stateless](#) entre cliente e servidor. Isso significa que nenhuma informação do cliente é armazenada entre solicitações GET e toda as solicitações são separadas e desconectadas.
- Armazenar dados em cache para otimizar as interações entre cliente e servidor.
- Ter uma interface uniforme entre os componentes para que as informações sejam transferidas em um formato padronizado. Para tanto, é necessário que:
 - os recursos solicitados sejam identificáveis e estejam separados das representações enviadas ao cliente;
 - os recursos possam ser manipulados pelo cliente por meio da representação recebida com informações suficientes para tais ações;
 - as mensagens autodescritivas retornadas ao cliente contenham informações suficientes para descrever como processá-las;
 - hipertexto e hipermídia estão disponíveis. Isso significa que após acessar um recurso, o cliente pode usar hiperlinks para encontrar as demais ações disponíveis para ele no momento.
- Ter um sistema em camadas que organiza os tipos de servidores (responsáveis pela segurança, pelo carregamento de carga e assim por diante) envolvidos na recuperação das informações solicitadas em hierarquias que o cliente não pode ver.
- Possibilitar código sob demanda (opcional): a capacidade de enviar um código executável do servidor para o cliente quando solicitado para ampliar a funcionalidade disponível ao cliente.

Conforme site do **RedHat** (<<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>)

2.4. BANCO DE DADOS RELACIONAL

Um banco de dados relacional é um tipo de banco de dados que armazena e fornece acesso a pontos de dados relacionados entre si. Bancos de dados relacionais são baseados no modelo relacional, uma maneira intuitiva e direta de representar dados em tabelas. Em um banco de dados relacional, cada linha na tabela é um registro com uma ID exclusiva chamada chave. As colunas da tabela contêm atributos dos dados e cada registro geralmente tem um valor para cada atributo, facilitando o estabelecimento das relações entre os pontos de dados. Conforme site da **Oracle** (<<https://www.oracle.com/br/database/what-is-a-relational-database/>>)

3. METODOLOGIA

Para o desenvolvimento desse sistema foi adotado o modelo de desenvolvimento cascata. Para o protótipo foi utilizado o software Figma, para a codificação foram utilizados o editor de texto Visual Studio Code e a IDE Visual Studio, ao final do desenvolvimento foi realizado o teste end-to-end da aplicação.

O modelo em cascata leva as atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução e as representa como separadas fases do processo, como especificação de requisitos, projeto de software, implementação, teste e assim por diante.

4. DESENVOLVIMENTO DO TEMA PROPOSTO

4.1. TECNOLOGIA UTILIZADA

4.1.1. .NET-CORE 2.1

Um framework open-source criado pela Microsoft para desenvolvimento web, composto por uma biblioteca padrão, um compilador e uma máquina virtual. Criado no intuito de auxiliar o dia a dia da rotina de desenvolvimento, ele oferece agilidade e alto desempenho nas aplicações.

Em relação à linguagem de programação e IDEs (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado), o .NET é bastante completo. Sendo assim, confira abaixo as linguagens e os IDEs mais usados em conjunto.

Linguagens

- C# (C Sharp): ela vem da família das linguagens C, é orientada a objetos e fortemente tipada.
- F#: é uma linguagem de programação que abrange programação funcional, orientada a objetos e imperativa.
- Visual Basic: tem sintaxe de fácil aprendizado, é estável e dá suporte a aplicativos web.

Conforme site da **Microsoft** (<<https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>>)

4.1.2. ASP.NET CORE MVC

Uma extensão do .NET, o ASP.NET CORE é uma plataforma com ferramentas específicas para o auxílio em desenvolvimento de aplicações web. Uma dessas ferramentas é o ASP.NET CORE MVC, que traz consigo o padrão de arquitetura MVC (Model-View-Controller), em um único pacote de projeto. Conforme site da **Microsoft** (<<https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>>)

4.1.3. VUE JS

Um framework para desenvolvimento de interfaces, criado a partir do HTML, CSS e JavaScript. Proporciona um desenvolvimento eficiente em programação baseada em componentes seja ele simples ou complexo.

Os requisitos para sua utilização são conhecimentos básicos em HTML e JavaScript, o tornando um framework bastante acessível e de fácil manuseio.

Algumas formas de como o VUE pode ser implementado são as seguintes:

- Componentes de Arquivo único
- Integração direta ao JavaScript

Conforme site da **VUE.JS** (<<https://vuejs.org/guide/introduction.html#single-file-components>>)

4.1.4. MySQL

Um SGBD (Sistema Gerenciador de Banco de Dados), desenvolvido pela Oracle. Consiste em um sistema simples de gerenciamento de banco de dados relacional de fácil manuseio utilizando a linguagem SQL (Structured Query Language) muito utilizado para aplicações web.

4.2. ARQUITETURA

4.2.1. Representação Arquitetural

Os sistemas serão desenvolvidos tendo como base a arquitetura ilustrada na Figura 1.

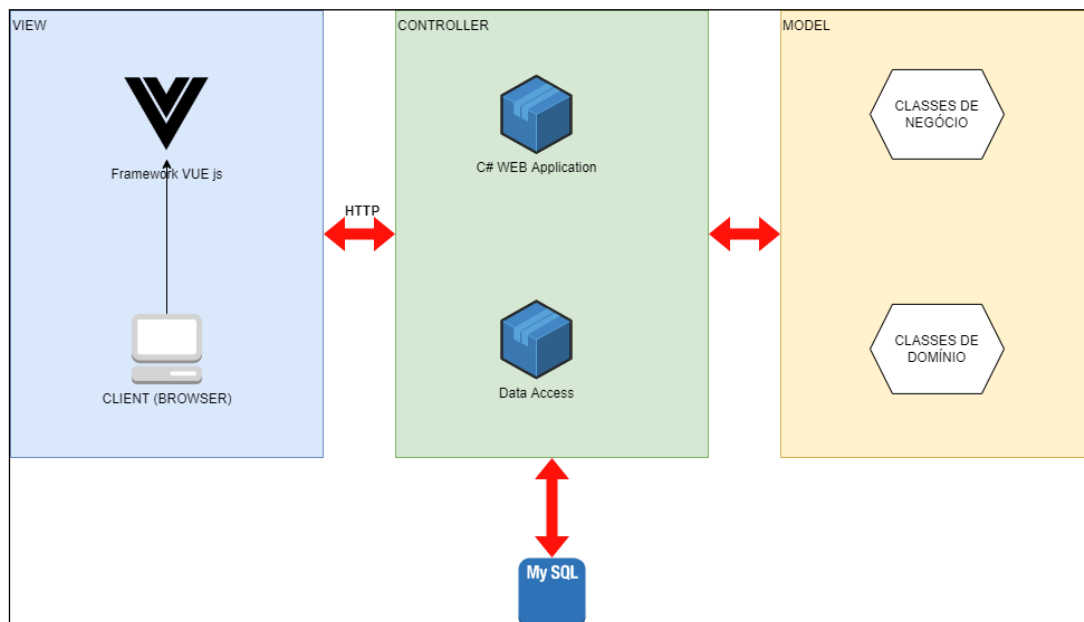


Figura 1 - Modelo Arquitetural

O Container VIEW abrigará os componentes da arquitetura responsáveis pela camada de apresentação. Será utilizado para esta camada os frameworks VUE e Bootstrap na implementação do modelo MVC.

As Classes de Domínio são as classes que representam os Value Object, contendo somente os atributos e os métodos getters/setters.

As Classes de Negócio representam as classes responsáveis por aplicar as regras de negócio do sistema como, por exemplo, Cadastrar Jogo. Constarão também nas classes de negócio o relacionamento com os DALs responsáveis por persistir e recuperar os objetos no banco de dados.

No C# WEB Application constarão os componentes de software principais. Para esta arquitetura de software, será utilizado o padrão de projeto DAL (Data Access Layer), criando uma camada de persistência na aplicação, assim tornando a aplicação desacoplada ao Banco de Dados, facilitando uma possível alteração do banco e sua manutenção.

4.2.2. Restrições Arquiteturais

Orientações pertinentes ao desenvolvimento deste subsistema:

- Utilização do C#;
- Utilização do banco de dados MySQL;

- Utilização do software figma para desenvolvimento do protótipo;
- Utilização dos frameworks VUE e Bootstrap para desenvolvimento da camada de apresentação;
- Utilização do framework ASP.NET core 2.1 ou superior para desenvolvimento da camada de negócio e controle de transações;

4.2.3. Objetivos e Restrições Arquiteturais

Alguns requisitos registrados que impactam diretamente a arquitetura do sistema RAK Game Shop são:

- A consideração de persistência das licenças dos jogos nas bases de dados MySQL, considerando uma criptografia das chaves para maior segurança;
- Considerando premissas definidas para o sistema RAK GameShop podese citar as seguintes restrições:
- Utilização da Linguagem C#
- Considerar a utilização de software Livre, quando possível
- Necessidade de criptografia das chaves de licenças dos jogos

5. RESULTADOS

A partir dos requisitos levantados e através da elaboração do sistema, obtivemos os seguintes resultados:

- Detalhamento do escopo e dos requisitos;
- Definição de design, logotipo da marca, paleta de cores e layout das telas;
- Elaboração do protótipo de telas, contendo as funcionalidades do sistema a ser desenvolvido;
- Codificação do frontend do sistema, respeitando o layout e funcionalidades apresentados no protótipo;
- Codificação do backend do sistema, contendo as regras de negócio e requisitos solicitados;

CONCLUSÕES

Com base nos dados coletados durante o desenvolvimento do sistema, concluímos que com o escopo e implementação atual, obtivemos um produto embrionário que pode nos trazer ideias de um sistema completo contendo implementações de novos módulos e funcionalidades, para que tenhamos uma solução que atenda às necessidades de negócio de uma loja virtual de jogos eletrônicos em todos os seus setores, por exemplo, financeiro, tecnologia, relacionamento com o cliente, marketing, administração, entre outros.

REFERÊNCIAS

ABNT. **Estrutura do Trabalho com ABNT.** Disponível em: <<https://www.normasabnt.org/>>. Acesso em: 20 mai. 2022.

GOURLEY, D. and TOTTY, B. and SAYER, M. and AGGARWAL, A. and REDDY, S. **HTTP: The Definitive Guide.** California: O'Reilly Media, 2002.

Microsoft. **ASP.NET.** Disponível em: <<https://dotnet.microsoft.com/en-us/learn/aspnet/whatis-aspnet>>. Acesso em: 23 mai. 2022.

MySQL. **MySQL.** Disponível em: <<https://dev.mysql.com/doc/mysql-getting-started/en/>>. Acesso em: 23 mai. 2022

Oracle. **RDBMS.** Disponível em: <<https://www.oracle.com/br/database/what-is-a-relationaldatabase/>>. Acesso em: 13 jun. 2022.

RedHat. **API REST.** Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-restapi>>. Acesso em: 13 jun. 2022.

SILVA, Maurício Samy. **Criando Sites com HTML: Sites de Alta Qualidade com HTML e CSS.** SP: Novatec Editora, 2008.

SOMMERVILLE, Ian. **SOFTWARE ENGINEERING: Ninth Edition.** Massachusetts: Pearson Education, 2011.

VUE. **VUE.JS.** Disponível em: <<https://vuejs.org/guide/introduction.html#single-filecomponents>>. Acesso em: 23 mai. 2022.

ANEXOS

REPOSITÓRIO CODIGO-FONTE

Link do repositório com o código do backend:

https://github.com/IsraelCurvelo/Project_RAK_GAME_SHOP/tree/master/RakGameShopAPI

Link do repositório com o código do frontend:

https://github.com/IsraelCurvelo/Project_RAK_GAME_SHOP/tree/master/rakameshop_client

Link do protótipo desenvolvido no FIGMA:

<https://www.figma.com/file/iQPv7IADIfB8nzuScFQ7Zh/TG?node-id=0%3A1>

REGRAS DE NEGÓCIO

RN0011	Dados obrigatórios para o cadastro de um produto	Para todo produto cadastrado é obrigatório o cadastro dos seguintes dados: Nome, Foto, Descrição, Categoria, Classificação, Tamanho, Data de Lançamento e Desenvolvedora;
RN0012	Associação com categorias	Um produto pode estar associado com mais de uma categoria.
RN0013	Cadastro de cliente	Para todo cliente cadastrado é obrigatório o cadastro dos seguintes dados: Nome, Apelido, Data de Nascimento, CPF, Telefone (deve ser composto pelo tipo, DDD e número), e-mail e senha.
RN0014	Composição do registro de cartões de crédito	Todo cartão de crédito associado a um cliente deverá ser composto pelos seguintes campos: Nº do Cartão, Nome impresso no Cartão, Bandeira do Cartão e Código de Segurança.
RN0015	Cadastro de custo de produto	Para todo produto deve haver um valor de custo.

REQUISITOS FUNCIONAIS

RF0011	CRUD produto	O sistema deve ser capaz de efetuar o cadastro, exclusão, alteração e consulta de jogos do acervo da empresa.
RF0012	CRUD Sacola	O sistema deve ser capaz de inserir, listar e retirar produtos da sacola.
RF0013	Controle de vendas	O sistema deve ser capaz de efetuar o controle de vendas.

REQUISITOS NÃO-FUNCIONAIS

RNF0011	Definição de linguagem	Utilizar linguagem C#
RNF0012	Definição de Banco de Dados	Utilizar o banco de dados MySQL
RNF0013	Definição de Arquitetura	A arquitetura da solução obedecer à arquitetura MVC.
RNF0014	Compatibilidade	O sistema deve rodar em qualquer navegador

DIAGRAMA DE CASO DE USO DE CLIENTE

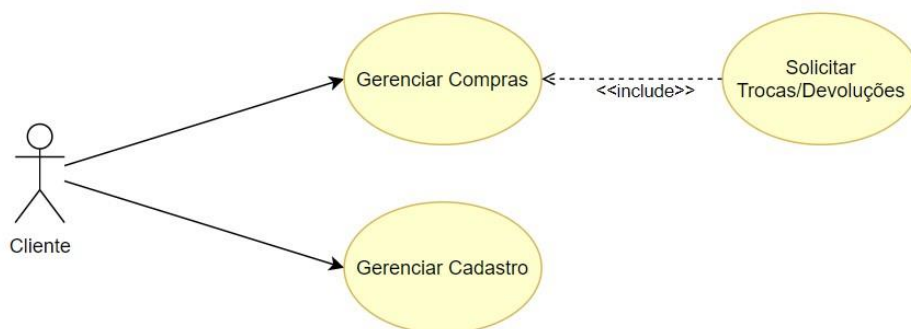
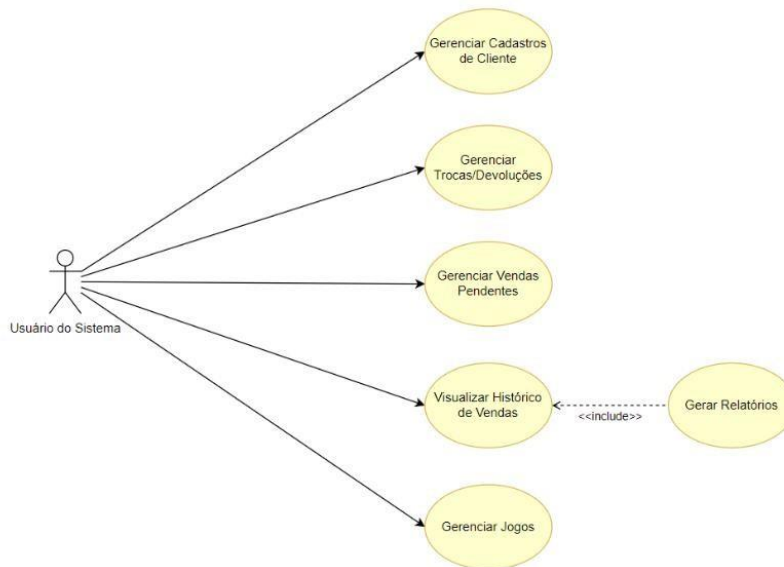


DIAGRAMA DE CASO DE USO DE USUÁRIO DO SISTEMA



TELAS DO SISTEMA

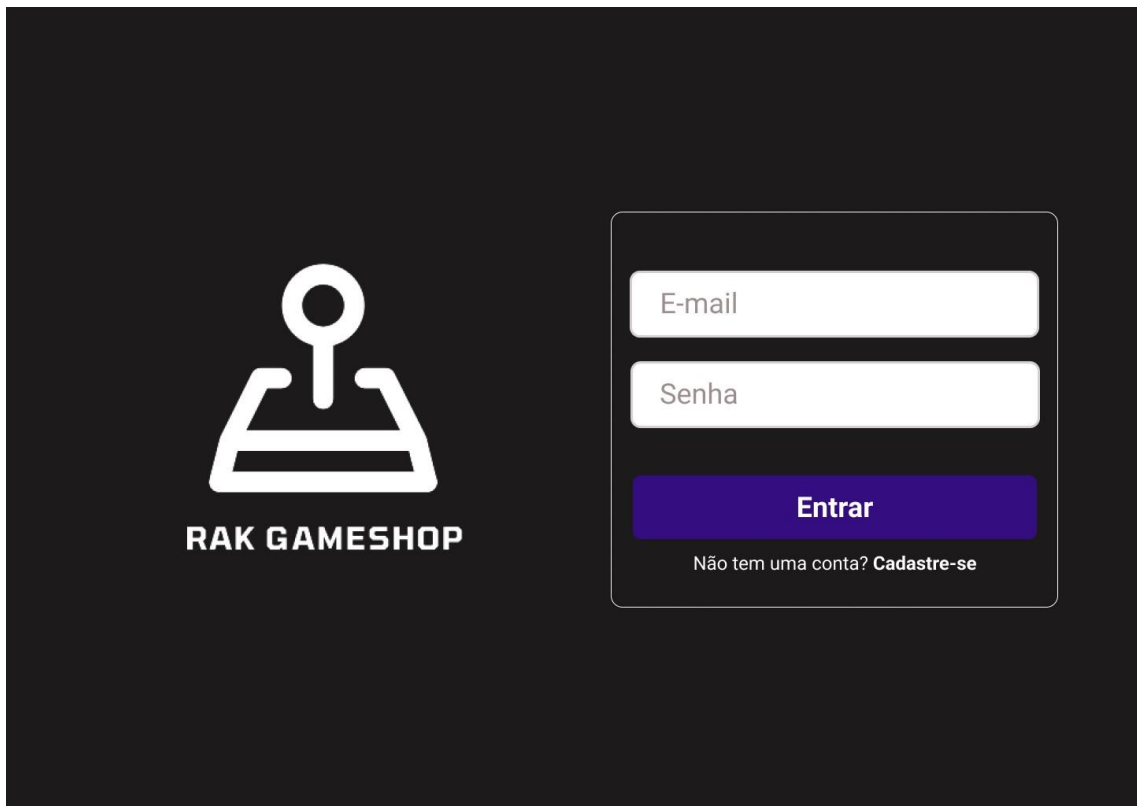




Figura 2. Login

Nome_usuario

Criar conta

Dados Cadastrais

Nome Completo

Apelido



E-mail

Data de Nascimento

CPF

Telefone

Figura 3. Cadastro de Cliente

Nome_usuario

Cadastro de Produto

Dados do Produto

Nome do Jogo

Foto do produto

Selecionar arquivo

Descrição

Categoria

Classificação

Tamanho

Data Lançamento

Desenvolvedora

Figura 4. Cadastro de Produto

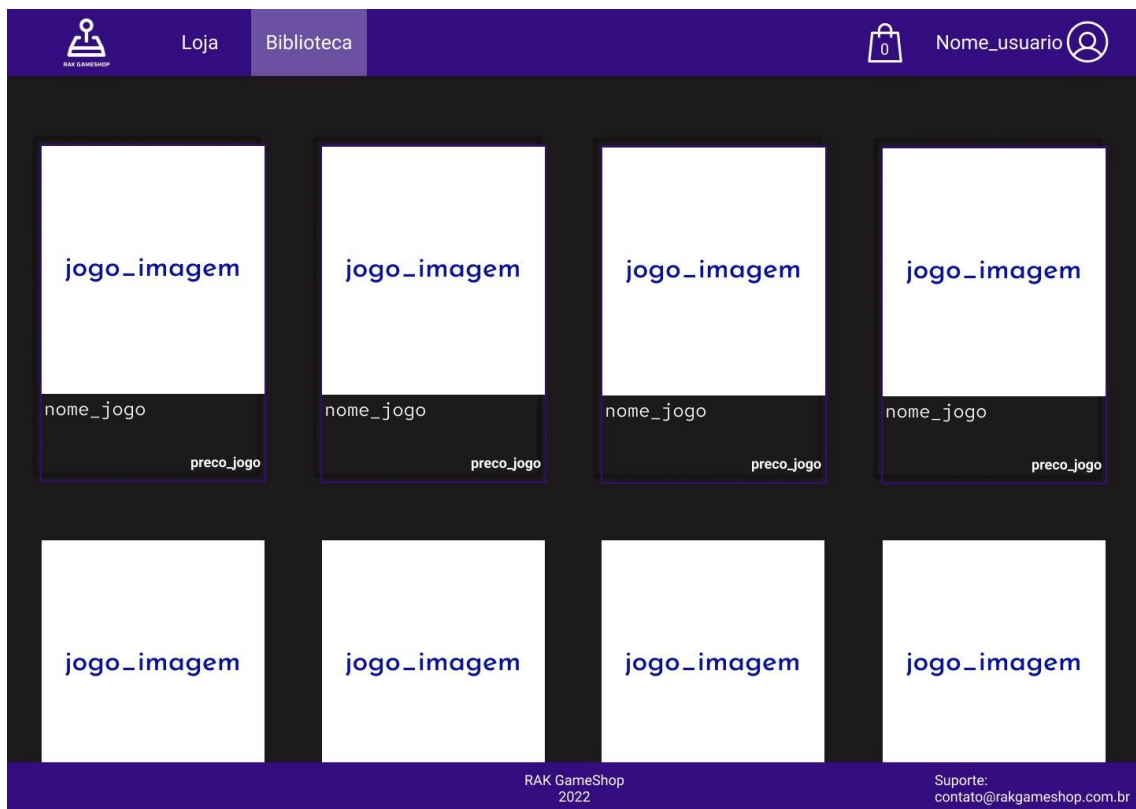


Figura 5. Biblioteca

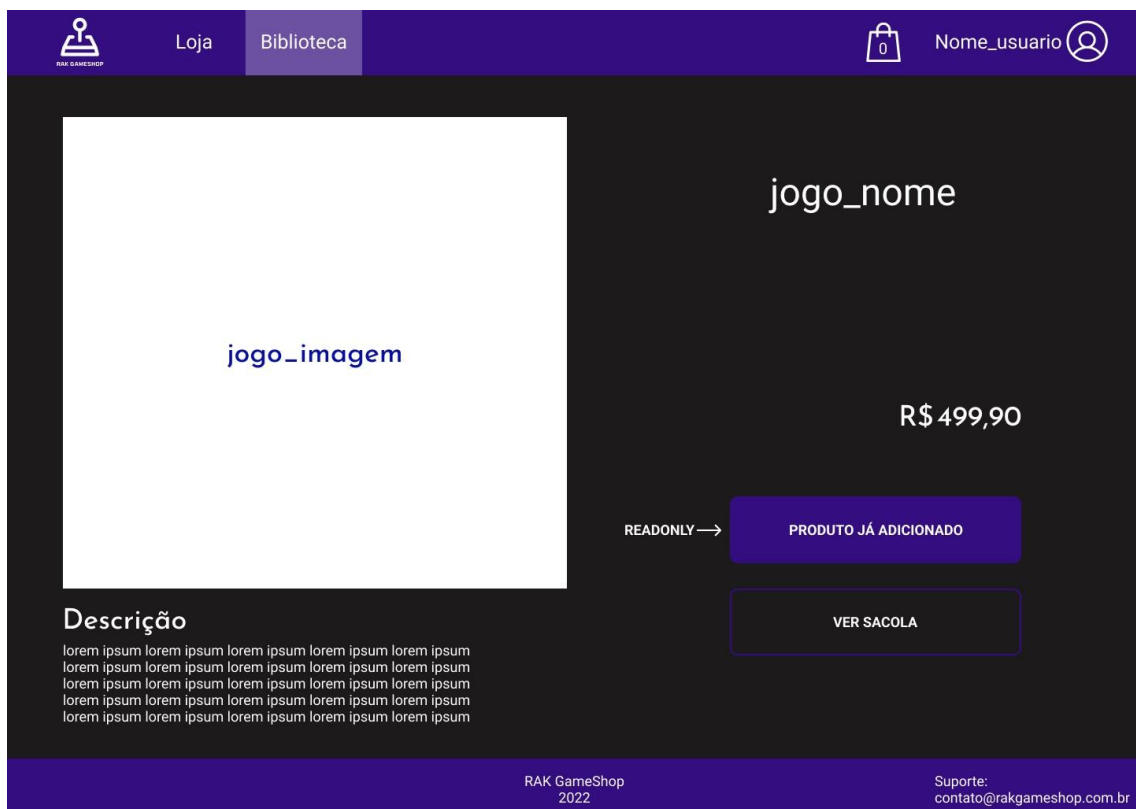





Figura 6. Detalhe Produto





LojaBiblioteca




Nome_usuario 

Finalizar compra

☐  Cartão de crédito

☐  Boleto

☐  PIX

Resumo do pedido

jogo_nome

jogo_valor

jogo_nome

jogo_valor

Total

valor_total

FAZER PEDIDO

RAK GameShop
2022

Suporte:
contato@rakgameshop.com.br

Figura 7. Pagamento

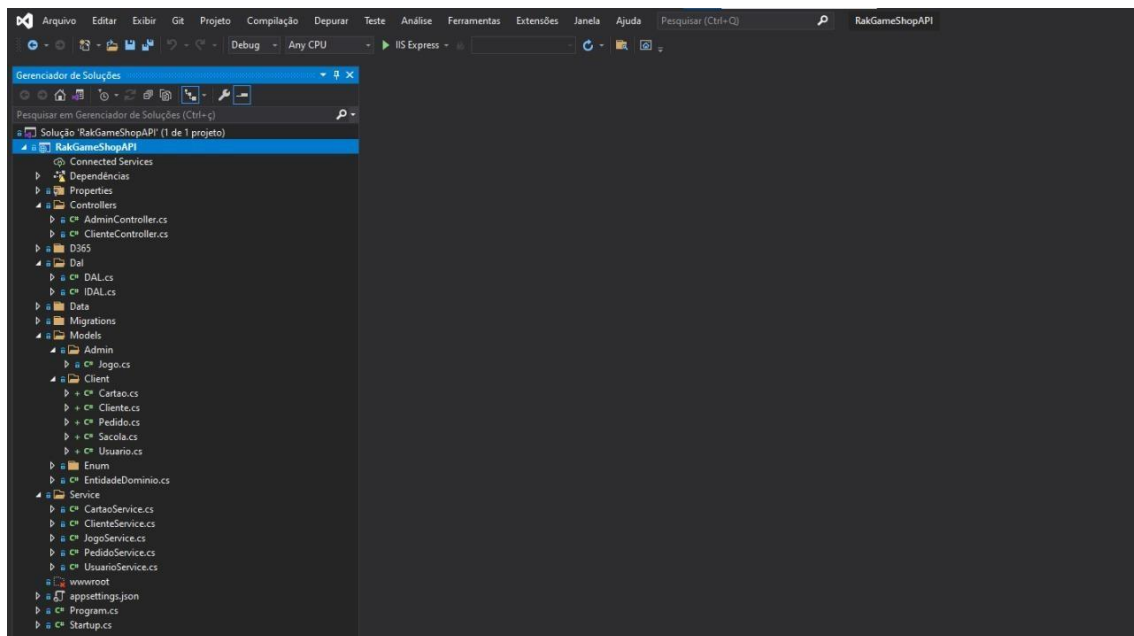


Figura 8. Estrutura de pastas

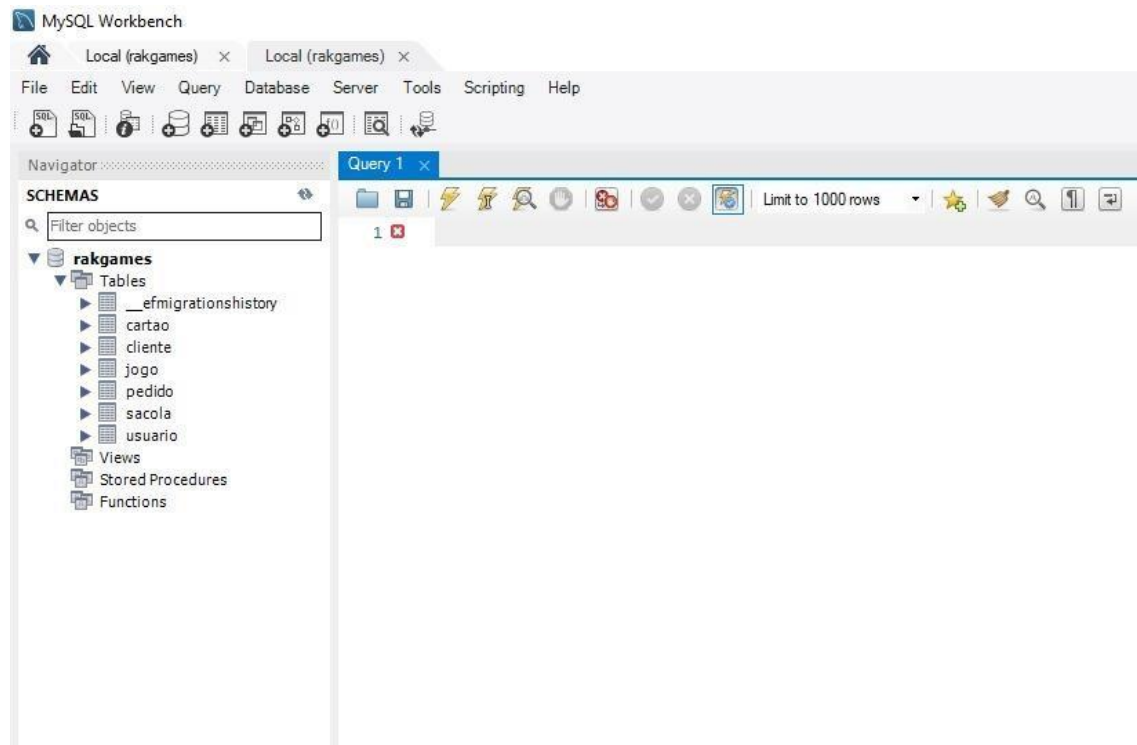


Figura 9. Banco de dados