

PROJETO: UMA LOJA DE JOGOS

Abraão F. de Medeiros, Givanilson G. Fernandes, Mickeias A. Lucena, Pedro V. Dantas,

Taynná M. Cardoso

Informática para Internet - Instituto Federal do Rio Grande do Norte (IFRN)

Rodovia RN 288, s/n, Nova Caicó - Caicó-RN – CEP: 59300-000

Programação de Sistemas para Internet (PSI)

abraao.m@escolar.ifrn.edu.br, givanilson.fernandes@escolar.ifrn.edu.br,
mickeias.a@escolar.ifrn.edu.br, azevedo.v@escolar.ifrn.edu.br, taynna.montel@escolar.ifrn.edu.br

Abstract: This project presents the development of a virtual library of external games for managing and organizing information. The application allows the registration, editing, deletion and listing of games through a web interface. To implement the system, modern technologies such as FastAPI, Next.js, SQLite, SQLAlchemy and Tailwind CSS were used, with the aim of offering a dynamic, intuitive and efficient environment to the user.

Resumo: Este projeto apresenta o desenvolvimento de uma biblioteca virtual de jogos voltada para o gerenciamento e a organização de informações. A aplicação permite o cadastro, edição, exclusão e listagem de jogos por meio de uma interface web. Para a implementação do sistema, foram utilizadas tecnologias modernas como FastAPI, Next.js, SQLite, SQLAlchemy e Tailwind CSS, com o objetivo de oferecer um ambiente dinâmico, intuitivo e eficiente ao usuário.

Palavras-chave: Jogos; Sistema Web; Biblioteca Virtual; FastAPI; Tecnologias Web.

1. Introdução

A produção de sistemas web tem se intensificado nos últimos anos, contribuindo para que o acesso a informações em diversas áreas seja mais fácil de ser alcançado. No âmbito da tecnologia e do entretenimento digital, plataformas voltadas para a organização de jogos tornam-se interessantes ao oferecer soluções práticas e eficientes para o controle de dados.

Diante desse cenário, este projeto propõe o desenvolvimento de uma plataforma **DiGames**, que é basicamente uma biblioteca virtual de jogos voltada para o gerenciamento e a organização de informações. A aplicação disponibiliza funcionalidades como cadastro, edição, exclusão e listagem de jogos, promovendo maior organização e facilidade no acesso às informações.

Além disso, o sistema apresenta um ambiente web dinâmico e intuitivo, buscando aliar simplicidade de uso às funcionalidades essenciais para o gerenciamento eficiente de dados.

2. Metodologia

O desenvolvimento do projeto foi realizado com base nos conteúdos da matéria de Programação de Sistemas para Internet (PSI), com foco na construção e no consumo de APIs REST, visando garantir uma implementação eficiente e organizada do sistema.

O **backend** da aplicação foi desenvolvido utilizando o framework FastAPI. Para a validação dos dados enviados à API, foi utilizado o Pydantic, responsável pela definição de modelos de dados em formato JSON, garantindo que as informações recebidas atendam aos critérios estabelecidos antes de serem processadas. Entre as validações aplicadas, destacam-se restrições como o tamanho mínimo de caracteres para campos textuais e a limitação de valores permitidos para determinados atributos.

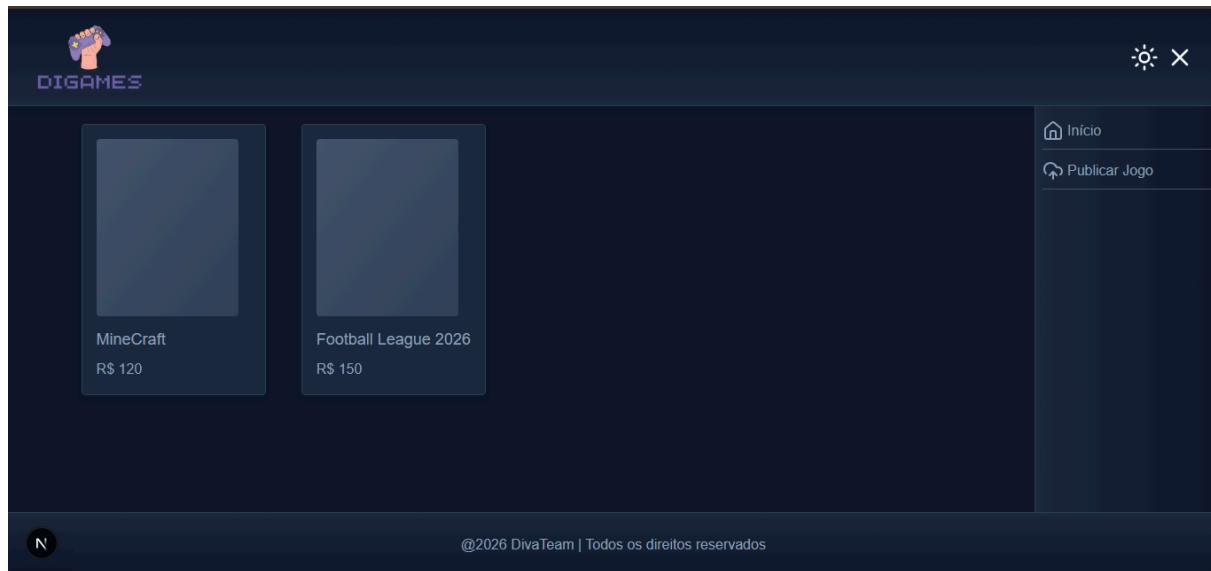
A configuração do CORS (Cross-Origin Resource Sharing) foi realizada com o objetivo de permitir a comunicação entre o frontend e a API desenvolvida em FastAPI. Essa configuração evita o bloqueio de requisições pelo navegador e assegura a integração adequada entre a aplicação cliente e o servidor.

O **frontend** foi desenvolvido utilizando JavaScript, com o auxílio do framework Next.js para a construção da interface. A estilização visual da aplicação foi realizada por meio do Tailwind CSS, visando proporcionar uma interface moderna, responsiva e de fácil utilização.

3. Resultados

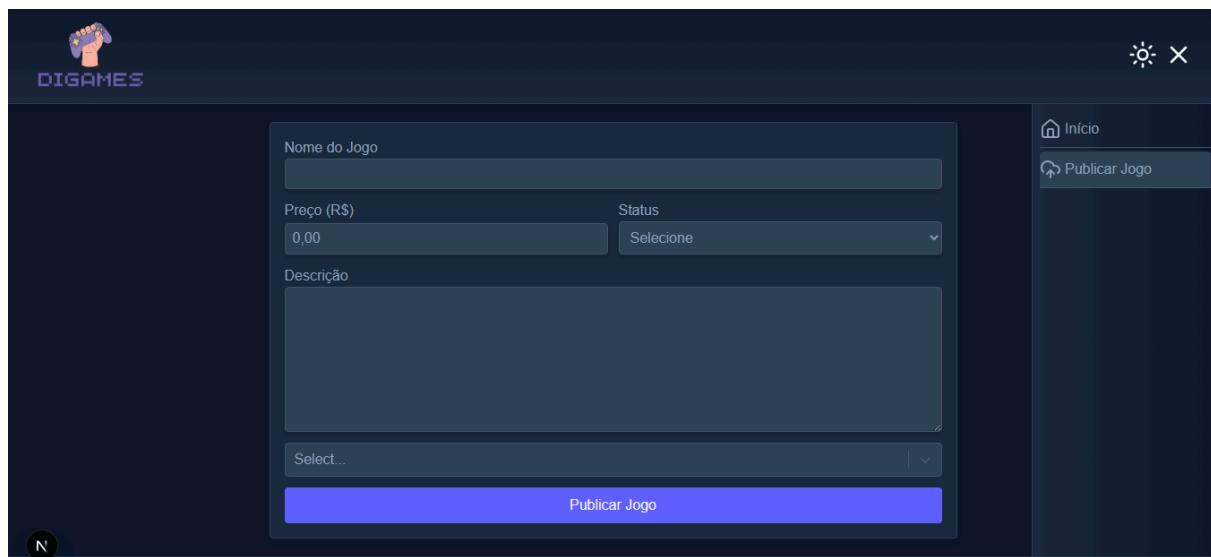
Neste ponto, são apresentados os resultados alcançados com a conclusão do projeto, por meio de registros visuais da interface da aplicação e de exemplos de dados armazenados no banco SQLite. Os elementos apresentados evidenciam o correto funcionamento das funcionalidades implementadas, incluindo cadastro, edição, exclusão e listagem de jogos.

Página Inicial com a listagem dos jogos



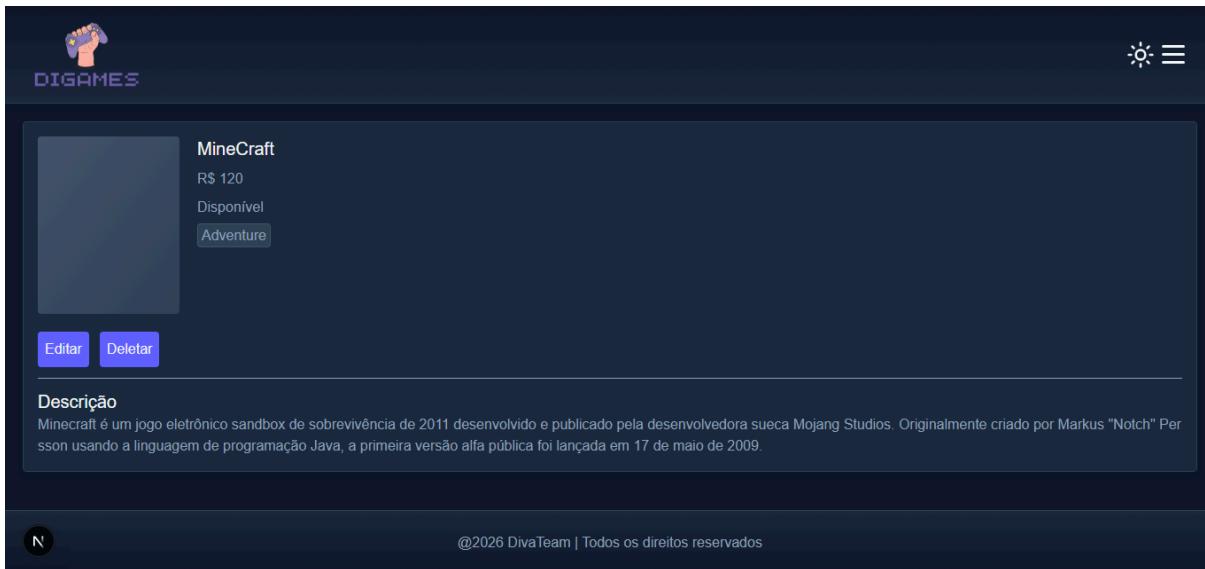
The screenshot shows the DIGAMES homepage with a dark blue header featuring the logo 'DIGAMES' and a search bar. Below the header, there are two game cards. The first card for 'MineCraft' shows a placeholder image and the price 'R\$ 120'. The second card for 'Football League 2026' also shows a placeholder image and the price 'R\$ 150'. To the right of the cards is a sidebar with links for 'Inicio' and 'Publicar Jogo'. At the bottom left is a 'Novo' button, and at the bottom center is a copyright notice: '@2026 DivaTeam | Todos os direitos reservados'.

Página de cadastro ou edição dos jogos



The screenshot shows a modal window for creating or editing a game. The form includes fields for 'Nome do Jogo' (Game Name), 'Preço (R\$)' (Price), 'Status' (Status), 'Descrição' (Description), and a file selection field 'Select...'. A large blue button at the bottom right is labeled 'Publicar Jogo' (Publish Game). The background of the page shows the same dark blue header and sidebar as the previous screenshot.

Página de Detalhes dos jogos com as opções de editar/excluir



Exemplo do uso de JSON no projeto (validação dos campos que virão do frontend)

```
STATUS_PERMITIDOS = ["Disponivel", "Indisponivel", "Pré-venda"]

class JsonJogoAdicionar(BaseModel):
    nome: str
    status: str
    generos: list[int]
    preco: float
    descricao: str

    @field_validator('nome', mode='before')
    @classmethod
    def validar_nome(cls, v):
        if not v or len(str(v).strip()) < 3:
            raise ValueError('Titulo deve ter no minímo 3 caracteres')
        return str(v).strip()

    @field_validator('status')
    @classmethod
    def validar_status(cls, v):
        if v and v not in STATUS_PERMITIDOS:
            raise ValueError(f'Status inválido. Permitidos: {" ".join(STATUS_PERMITIDOS)}')
        return v
```

4. Conclusão

O desenvolvimento da plataforma **DiGames** possibilitou a aplicação prática dos conceitos relacionados à Programação de Sistemas para Internet, especialmente no que se refere ao consumo de APIs REST. O sistema desenvolvido atendeu ao objetivo proposto, ao disponibilizar funcionalidades essenciais para o gerenciamento de uma biblioteca virtual de jogos, como cadastro, edição, exclusão e listagem de informações.

Os resultados obtidos demonstram que a aplicação apresenta funcionamento adequado, com integração eficiente entre frontend e backend, além de uma interface intuitiva e de fácil utilização. Dessa forma, o projeto contribuiu para o aprimoramento dos conhecimentos técnicos dos desenvolvedores e para a compreensão dos processos envolvidos no desenvolvimento de sistemas web.

Algumas limitações foram identificadas ao longo do desenvolvimento e são apontadas como possibilidades para trabalhos futuros, entre elas a implementação de um sistema de autenticação e controle de acesso, bem como a persistência de dados em um banco de dados relacional. Além disso, sugere-se o aprimoramento da interface e a inclusão de novas funcionalidades, visando tornar o sistema mais completo e robusto.

5. Referências

FASTAPI. *FastAPI Documentation*. Disponível em: <https://fastapi.tiangolo.com/>. Acesso em: fev.

2026.

NEXTJS. *Next.js Documentation*. Disponível em: <https://nextjs.org/docs>. Acesso em: fev. 2026.

PYDANTIC. *Pydantic Documentation*. Disponível em: <https://docs.pydantic.dev/>. Acesso em: fev. 2026.

SQLALCHEMY. *SQLAlchemy Documentation*. Disponível em: <https://docs.sqlalchemy.org/>. Acesso em: fev. 2026.

TAILWIND CSS. *Tailwind CSS Documentation*. Disponível em: <https://tailwindcss.com/docs>. Acesso em: fev. 2026.