

# **Relatório do trabalho da cadeira Fundamentos de Banco de Dados Plataforma de Speedrun de Jogos**

**Nícolas Ferreira Leite - 552425, Kauan Pablo de Sousa Silva - 556027**

Universidade Federal do Ceará (UFC) - Campus Quixadá

Ciência da Computação

Fundamentos de Banco de Dados ministrada por Livia Almada

## **1. Introdução**

No mundo dos jogos eletrônicos, a prática conhecida como speedrun, que consiste em completar um jogo o mais rapidamente possível, tornou-se cada vez mais popular ao longo dos anos. À medida que esta tendência crescia, precisávamos de uma plataforma que não só reunisse entusiastas da prática, mas também fornecesse uma forma organizada e eficiente de documentar e compartilhar os resultados alcançados pelos speedrunners (termo designado aos jogadores que praticam speedrun). É neste contexto que surge o projeto de desenvolvimento de um site de registo de speedruns

O objetivo principal da plataforma é fornecer uma central online para a comunidade speedrunner onde eles podem registrar seus tempos, compartilhar estratégias, interagir com outros jogadores e acompanhar o progresso de suas próprias corridas e de outros.

Neste relatório, detalharemos a arquitetura e as funcionalidades do desenvolvimento dessa plataforma de banco de dados dedicada ao universo do speedrun.

## **2. Requisitos Funcionais**

REQ 001 - Cadastro de Usuário

Descrição: Consiste em cadastrar uma nova conta no sistema. As informações necessárias para a criação da conta são um nickname, nome, email, senha, país de origem e uma foto de perfil.

Restrições:

O email e o nickname devem ser únicos, ou seja, não podem ser usados por mais de uma conta.

A senha precisa ter no mínimo 8 dígitos, incluindo números, letras e caracteres especiais.

REQ 002 - Login de Usuário

Descrição: Consiste em realizar o login de um usuário no sistema usando email e senha previamente cadastrados.

Restrições:

O login só é permitido se o email e a senha corresponderem a uma conta existente no sistema.

#### REQ 003 - Cadastro de Jogo

Descrição: Consiste em adicionar um jogo na plataforma. As informações necessárias são o nome do jogo, capa do jogo, plataformas em que foi lançado, categorias de speedrun, data lançamento e um resumo do jogo.

Restrição: O jogo deve ser único na plataforma.

#### REQ 004 - Cadastro de Run (corrida, ou seja, tentativa de finalizar o jogo)

Descrição: O usuário pode solicitar a adição de uma run na plataforma, onde um moderador deve aceitá-la. As informações necessárias são o usuário que solicitou, o jogo, o tempo da run, o vídeo para comprovar, a data da run, a versão do jogo, a plataforma que ele jogou e uma descrição da run.

Restrição: A data da run deve ser posterior a do lançamento do jogo.

#### REQ 005 - Comentários

Descrição: O usuário pode comentar na página do jogo, ou em alguma run. As informações necessárias são o usuário que comentou, data do comentário, número de likes, respostas.

Restrição: A curtida em um comentário é única por usuário.

#### REQ 006 - Seguir

Descrição: É possível seguir um jogo ou usuário. As informações necessárias são o usuário que seguiu.

Restrição: Sem restrições significativas.

#### REQ 007 - Ranking de Classificação

Descrição: Uma página que organiza todas as runs de um jogo em suas respectivas categorias. É necessário ter as runs que foram adicionadas no jogo.

Restrição: Sem restrições significativas.

### **3. Descrição Textual do Modelo ER/EER**

#### **Entidades:**

O modelo inclui 6 entidades: "Usuário", que pode ser um "Jogador" ou "Moderador"; "Jogo", onde uma "Run" é registrada; "Categoria", associada às runs; e "Comentário", feito em um jogo ou em uma run.

#### **Atributos:**

- Usuário:
  - id, nome, senha, nickname, quantidade\_de\_seguidores (DERIVADO), foto\_de\_perfil (OBRIGATÓRIO), pais\_de\_origem.
- Moderador:
  - nível.
- Comentário:
  - data/hora.
- Jogo:
  - id, nome, plataforma, quantidade\_de\_seguidores (DERIVADO), capa, ano\_de\_lancamento, resumo\_do\_jogo.
- Run:
  - id, plataforma, video, tempo, data, versao.
- Categoria:
  - id, nome, descricao.

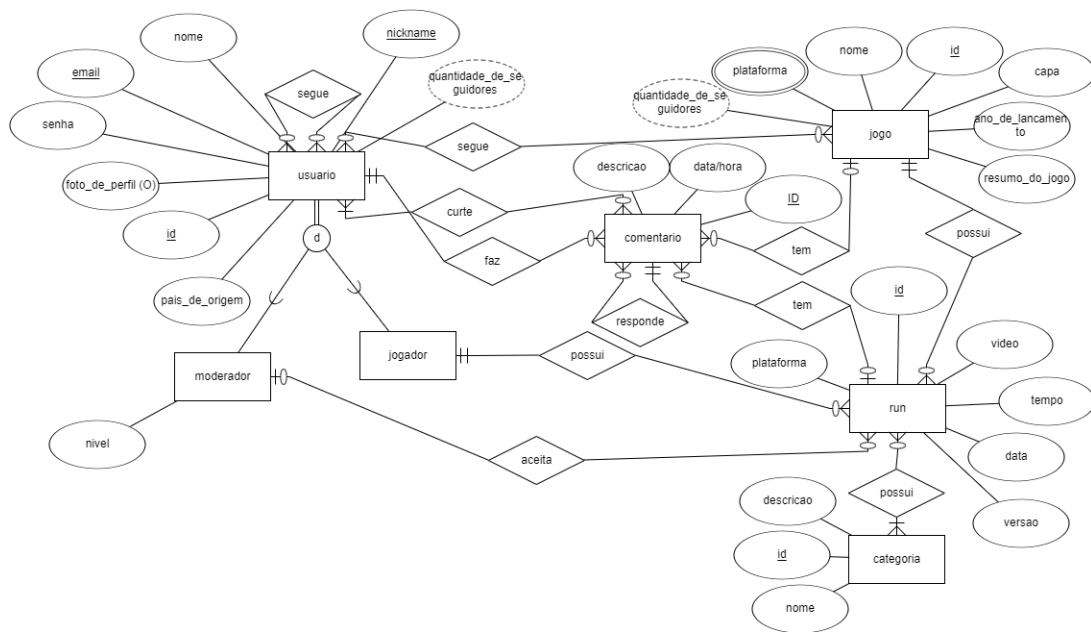
#### **Relacionamentos:**

- O Usuário pode seguir nenhum ou muitos jogos.
- O Usuário pode seguir muitos ou nenhum usuário.
- O Usuário pode fazer muitos ou nenhum comentário, o comentário obrigatoriamente deve estar associado a um único usuário.
- O Usuário pode curtir muitos ou nenhum comentário
- O Jogador possui nenhuma ou muitas runs, mas uma run deve ser obrigatoriamente possuída por um usuário.
- O Moderador pode aceitar nenhuma ou muitas runs, e a run deve ser aceita por um único moderador, sendo essa aceitação não necessária.
- Um jogo/run tem muitos ou nenhum comentário, um comentário só pode estar associado a apenas um jogo ou apenas a uma run.

- Um jogo possui muitas ou nenhuma run, uma run deve ser exclusiva de um jogo.
- Uma run possui obrigatoriamente uma ou muitas categorias e uma categoria pode estar associada a nenhuma ou muitas runs.
- Um comentário pode ter muitas ou nenhuma resposta, e uma resposta deve ser feita obrigatoriamente a um único comentário.

#### 4. Diagrama ER/EER

Aqui temos o diagrama do projeto, feito com a ferramenta erdplus, ele nos mostra uma representação completa de como queremos que os dados do projeto sejam relacionados.



#### 5. Conclusão da parte 1

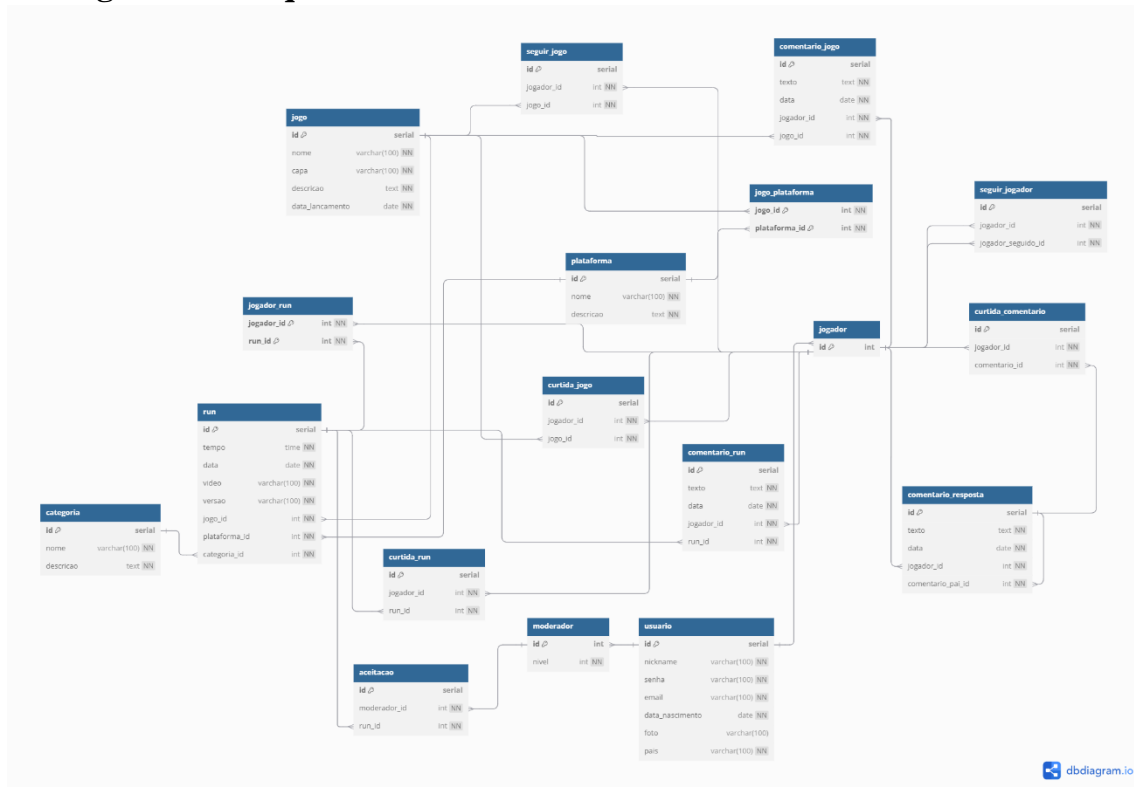
Nesta primeira fase do trabalho o foco foi o levantamento de requisitos para a criação do diagrama EER, que será utilizado como base para modelar o banco de dados nas próximas partes do trabalho. Esta parte nos permitiu compreender de melhor forma os diferentes aspectos que devem ser considerados na criação de uma aplicação funcional e eficiente.

Durante esse processo foram adquiridas habilidades importantes que tem relação à coleta e análise de requisitos, assim aprendemos como coletar requisitos de um projeto do zero tirando disso especificações claras do que precisamos para o desenvolvimento do sistema.

#### 6. Ajustes da Etapa 1

Adicionado o campo ID e descrição, chaves que faltavam no comentário, e sobre os requisitos, foram separados o requisito login/cadastro de usuário e o requisito de relatório é Ranking de Classificação.

## 7. Diagrama do esquema relacional



## 8. Especificação das chaves primárias e candidatas

Tabela: jogo

- **Chave Primária:** id
- **Chaves Candidatas:** id, nome
- **Atributos:** nome, capa, descricao, data\_lancamento

Tabela: plataforma

- **Chave Primária:** id
- **Chaves Candidatas:** id, nome
- **Atributos:** nome, descricao

Tabela: jogo\_plataforma

- **Chave Primária:** (jogo\_id, plataforma\_id)
- **Chaves Candidatas:** (jogo\_id, plataforma\_id)
- **Atributos:** jogo\_id, plataforma\_id

Tabela: categoria

- **Chave Primária:** id

- **Chaves Candidatas:** id, nome
- **Atributos:** nome, descricao

Tabela: run

- **Chave Primária:** id
- **Chaves Candidatas:** id
- **Atributos:** tempo, data, video, versao, jogo\_id, plataforma\_id, categoria\_id

Tabela: usuario

- **Chave Primária:** id
- **Chaves Candidatas:** id, nickname, email
- **Atributos:** nickname, senha, email, data\_nascimento, foto, pais

Tabela: moderador

- **Chave Primária:** id
- **Chaves Candidatas:** id
- **Atributos:** nivel

Tabela: jogador

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: jogador\_run

- **Chave Primária:** (jogador\_id, run\_id)
- **Chaves Candidatas:** (jogador\_id, run\_id)

Tabela: aceitacao

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: comentario\_jogo

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: comentario\_run

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: comentario\_resposta

- **Chave Primária:** id

- **Chaves Candidatas:** id

Tabela: curtida\_jogo

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: curtida\_run

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: curtida\_comentario

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: seguir\_jogo

- **Chave Primária:** id
- **Chaves Candidatas:** id

Tabela: seguir\_jogador

- **Chave Primária:** id
- **Chaves Candidatas:** id

## 9. Implementação dos atributos multivalorados

Os atributos multivalorados foram implementados utilizando tabelas auxiliares com chaves primárias compostas. Por exemplo, a relação entre **jogo** e **plataforma** foi representada pela tabela **jogo\_plataforma**, que permite associar múltiplas plataformas a um jogo.

## 10. Implementação da Hierarquia de Herança

A hierarquia de herança foi implementada utilizando a estratégia de tabela por subclasse. Por exemplo, as tabelas **usuario**, **moderador** e **jogador** compartilham o atributo id, onde **moderador** e **jogador** estendem **usuario**.

## 11. Conclusão Parte II

Na segunda fase do nosso trabalho, avançamos bastante com o desenvolvimento ao implementar o modelo relacional, seguindo os requisitos que havíamos levantado anteriormente. Fizemos alguns ajustes na modelagem inicial e definimos as chaves primárias e candidatas. Também tratamos dos atributos multivalorados e aplicamos conceitos de herança.

Além disso, populamos o banco de dados, o que nos permitiu fazer uma primeira verificação prática da estrutura que criamos. Esse processo nos ajudou a entender melhor como transformar um diagrama EER em um banco de dados relacional funcional e eficiente, mostrando o quão crucial é prestar atenção aos detalhes durante a transição entre as etapas do desenvolvimento.

## 12. Ajustes da etapa 2

Separação da tabela comentários para diferentes tipos de comentário além da remoção de numcurtidas, atualização do modelo ER/ERR e do diagrama do esquema relacional.

## 13. Protótipo da aplicação.

O protótipo da aplicação foi desenvolvido em Jupyter Notebook e pode ser acessado pelo seguinte link: [Repositório do GitHub](#), além do vídeo de apresentação: [Vídeo no drive](#).

A aplicação permite o login e cadastro de usuários, além de pesquisas por ranking de classificação e a adição de novas runs.

A interface é dividida em quatro telas principais: Login, Cadastro, Ranking e Adicionar Run(sendo as últimas duas na mesma tela). Na tela de Login, os usuários podem inserir seu nickname e senha para autenticação, e se o login for bem sucedido ele entra em uma tela com o ranking de classificação e a opção de cadastrar run, inserindo as informações referentes a run. (Não precisa especificar id ou nome de usuário porque o login já guarda o id em uma variável global)

Na tela de Cadastro, um formulário solicita informações como nickname, senha, e-mail, data de nascimento, país e um link ou caminho para a foto do usuário, a função de cadastrar garante que não vai ter campos vazios e que o nickname e email são únicos.

Por fim, a tela de Ranking permite que os usuários busquem o ranking de runs específicas, filtrando por nome do jogo e categoria da run. Uma consulta SQL retorna os resultados, que são apresentados em uma tabela por ordem de classificação (tempo), com as informações do jogador que fez a run e as informações da run.



## 14. Tarefas realizadas pelos Integrantes

Todas as etapas do projeto foram feitas em conjunto em reuniões que tínhamos em algumas noites.

Atividade	Participantes
Escolha do tema	Nicolas e Kauan
Levantamento de Requisitos	Nicolas e Kauan
Criação do Modelo EER	Nicolas e Kauan
Criação do Relatório da Parte I	Nicolas e Kauan
Ajustes da Etapa I	Nicolas e Kauan
Diagrama do Esquema Relacional	Nicolas e Kauan
Implementação do Esquema Relacional	Nicolas e Kauan
Povoamento do Banco de Dados	Nicolas e Kauan
Criação do Relatório da Parte II	Nicolas e Kauan
Ajustes da Etapa II	Nicolas e Kauan
Tela de login/ranking de classificação	Nicolas
Tela de cadastro usuário e cadastro de runs	Kauan
Criação do Relatório Parte III	Nicolas e Kauan

## **15. Conclusão Parte II**

Desenvolver este protótipo da aplicação de speedrun em Jupyter Notebook foi uma experiência valiosa para nós. Aprendemos a usar ferramentas e tecnologias que expandiram nosso conhecimento em programação e desenvolvimento de aplicativos. O Jupyter Notebook se mostrou perfeito para prototipagem rápida, permitindo que testássemos e ajustássemos funcionalidades de maneira ágil. Além disso, trabalhar com SQLAlchemy para interagir com o banco de dados PostgreSQL nos ajudou a entender melhor como gerenciar dados e construir consultas eficazes.