

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSOS DE CIÊNCIA E ENGENHARIA DA COMPUTAÇÃO**

**DISCIPLINA DE CLASSIFICAÇÃO E PESQUISA DE DADOS  
PROF. LEANDRO KRUG WIVES**

**TRABALHO FINAL – ETAPA II**

**Alunos:**

Ian Kersz Amaral - 00338368

Marcos Luiz Kurth Reckers - 00315653

---

**DESCRIÇÃO DO PROBLEMA:**

Para a realização dessa atividade optamos por buscar uma base de dados de jogos da Steam, visto que ambos os integrantes se interessam bastante pelo tema. Assim sendo, pensamos em desenvolver um programa que consiste em uma interface gráfica, feita em Python, e um programa feito em C++, o qual realiza um conjunto de ações em uma base de dados, possibilitando ao usuário salvar as informações de jogos do seu interesse, bem como comparar dois jogos e também possibilite a aplicação de vários filtros para achar jogos similares;

**FUNCIONALIDADES PREVISTAS:**

1. Extrair dados de CSV e Criar a base de dados em Binário (.bin);
2. Possibilidade de adição de novos valores a base de dados: Sendo um único valor ou a adição de uma nova base de dados a atual, não devendo duplicar os valores já existentes na base original; Em caso de conflitos tais como: Mesmo id do jogo mas valores de avaliações diferentes, deve mostrar ao usuário para que ele escolha qual valor deve manter(indicando qual valor é o mais recente e qual é o original);
3. Pesquisar na base de dados: Pesquisar por qualquer elemento dos jogos, sendo ela parcial ou completa, podendo ter máscara, Ex:  
Nome: Shad, Década de 80: mostra vários jogos com o nome Shadow e Shaders somente da década de 80.
4. Reorganizar a forma que os dados são mostrados em tela: A interface em python pode reverter a lista de elementos que está em tela, além de ter diversas páginas para que o usuário possa ler todos os dados.
5. Ter um botão para recriar a base de dados inteira com um novo arquivo CSV, além de que se o programa já detectar uma base de dados existente, deve carregar ela automaticamente.

6. Dar possibilidade do usuário criar uma lista de favoritos para salvar alguns jogos se desejar;

### **CRONOGRAMA DE ATIVIDADES (ETAPAS, RESPONSÁVEIS E PRAZOS):**

#### **Atividades a serem realizadas exclusivamente em dupla:**

1. Criação do repositório no GitHub (09/03/2023) Já concluído
2. Reuniões semanais: 16/03, 23/03, 30/03;
3. Reunião de revisão final (01/04);

#### **Atividades que podem ser realizadas em conjunto ou individualmente:**

4. Criação da função de conversão de CSV para .bin; (Semana 13-19) [Ian]
5. Criação da classe de Árvore B+; (Semana 13-19) [Ian]
6. Criação da classe de Árvore Patricia; (Semana 13-19) [Marcos]
7. Criação da classe de Tabela Hash; (Semana 13-19) [Ian]
8. Criação da classe de Tabela Fixa; (Semana 13-19) [Marcos]
9. Criar funções de busca; (Semana 20-26) [Marcos]
10. Criar funções de ordenamento; (Semana 20-26) [Marcos]
11. Invocação das operações por linha de comando; (Semana 20-26) [Ambos]
12. Finalização da aplicação em C++; (Semana 20-26) [Ambos]
13. Construção da interface gráfica; (Semana 27-31) [Marcos]
14. União do aplicativo em C++ com a interface em Python; (Semana 27-31) [Ian]
15. Entrega do trabalho;(03/04/2023) [Ambos]

OBS: Dados descritos acima servem apenas para referência, podendo sofrer alterações ao longo do processo;

## FERRAMENTAS E BIBLIOTECAS A SEREM UTILIZADAS:

### Linguagens:

C++: Bibliotecas padrões.

Python: PySimpleGUI, e Bibliotecas padrões.

### Ferramentas:

VSCode: Editor de códigos a ser utilizado, principalmente com a extensão Live Share;

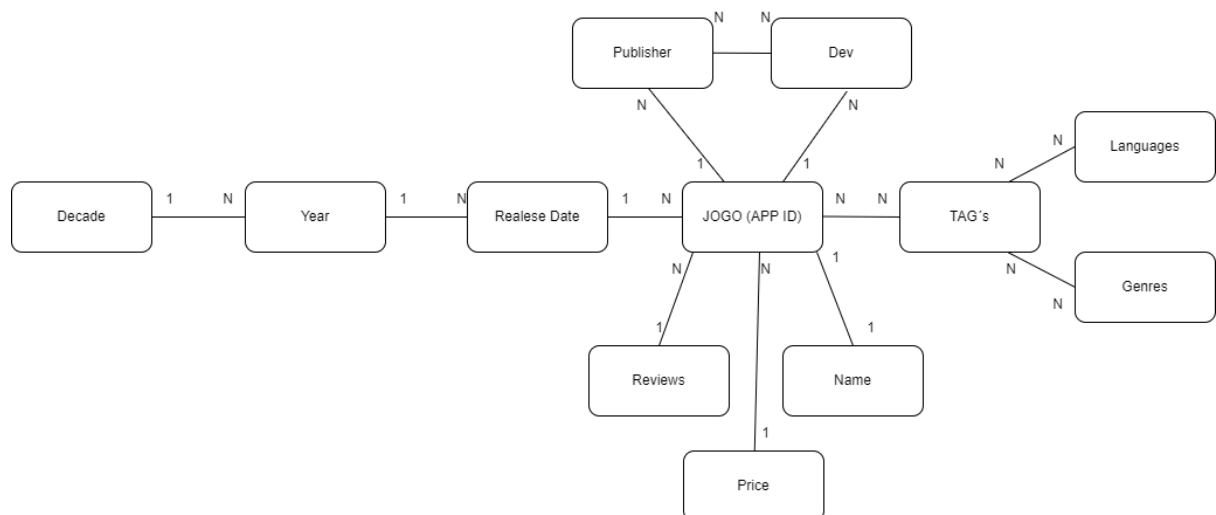
Git e Github: Gestão de versões e armazenamento do código para trabalho em conjunto, utilizando diversas Branches para progredir o trabalho de forma rápida; [GitHub do Projeto](#)

Live Share: Ferramenta que possibilita duas ou mais pessoas editarem o mesmo arquivo simultaneamente;

Trello: Organização das tarefas em Kanban para melhor divisão de trabalho; [Trello do Projeto](#)

Discord: Para comunicação e pair programming.

## PROJETO DE ARQUIVOS (DIAGRAMA ER OU DIAGRAMA DE CLASSES):



Árvores B+: Duas, uma com K: AppID e V: Todos os dados, e outra com K: Preço e V: AppID.

Árvores Patrícia: Key seria a substring do nome, valor seria o AppID

Tabelas Fixas: Década, Ano, Reviews

Tabelas Hash: Tags, Publishers, Developers