Vitor de Souza Reis

Relatorio tecnico

Este relatório descreve a organização interna do projeto “TRAB01-GOGOBOYS-DA-TI”, detalhando a nomenclatura de cada diretório e arquivo, sua responsabilidade funcional e a motivação por trás de cada escolha de nome. A padronização facilita a navegação, a manutenção e a escalabilidade do sistema de análise de perfis Steam.

Estrutura de Pastas

data/steam

Armazena todos os CSVs brutos do dataset Steam (players.csv, games.csv etc.). O subdiretório “steam” explicita a origem dos dados e isola-os de outros datasets futuros.

docs/

Contém artefatos de documentação, como o relatório técnico e manuais. Separar documentação do código evita confusão entre artefatos de uso e de implementação.

include/

Concentra todos os headers (.h) que definem interfaces e estruturas de dados.

– data/SteamDataLoader.h: declara o carregador genérico de CSVs, abstraindo o acesso a arquivos.

– entities/: define as entidades do domínio (Player.h, Game.h, Achievement.h, History.h, Price.h, PurchasedGame.h). Cada arquivo possui o nome da entidade que representa, tornando claro o modelo de dados do projeto.

– queries/QueryEngine.h: expõe a camada de consulta (API de busca e cruzamento de dados). A pasta “queries” delimita o escopo de lógica de negócio, separando-o de armazenamento.

– structures/: inclui as estruturas de índice (HashTable.h, BTree.h), isolando a implementação de algoritmos de acesso de dados do restante do sistema.

src/

Concentra todas as implementações (.cpp) correspondentes aos headers em include/

– data/SteamDataLoader.cpp: implementa a lógica de leitura/parsing dos arquivos CSV.

– queries/QueryEngine.cpp: contém as operações de filtro, junção e agregação de dados.

– structures/BTree.cpp, HashTable.cpp: implementam as estruturas de dados selecionadas para índices, garantindo desempenho nas consultas.

tests/

Agrupa casos de teste automatizados e CMakeLists.txt específico para validar cada módulo (test\_data\_loader.cpp, test\_hash\_table.cpp, test\_btree.cpp, test\_queries.cpp). Facilita executar e manter suites de testes isoladas do código de produção.

build/ (não versionado)

Destinado à saída dos artefatos gerados pelo CMake/compilação. Separar a build mantém o repositório limpo e evita conflitos de arquivos binários em controle de versão.