

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE

WIRLLEY DE OLIVEIRA DELFINO

**SOFTWARE DE AUXÍLIO PEDAGÓGICO PARA ENSINO DE CONSULTAS SIMPLES A
BANCO DE DADOS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2021

WIRLLEY DE OLIVEIRA DELFINO

**SOFTWARE DE AUXÍLIO PEDAGÓGICO PARA ENSINO DE CONSULTAS SIMPLES A
BANCO DE DADOS**

Trabalho de Conclusão de Curso apresentado
ao Curso de Bacharelado em Engenharia de
Software da Universidade Tecnológica Federal
do Paraná, como requisito para a obtenção do
título de Bacharel em Engenharia de Software
Orientador: Prof. Dr. Eduardo Cotrin Teixeira

CORNÉLIO PROCÓPIO

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos.

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Nome da Diretoria
Nome da Coordenação
Nome do Curso



TERMO DE APROVAÇÃO

SOFTWARE DE AUXÍLIO PEDAGÓGICO PARA ENSINO DE CONSULTAS SIMPLES A BANCO DE DADOS

por

Wirley de Oliveira Delfino

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Software” e aprovado em sua forma final pelo Programa de Graduação em Engenharia de Software da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 01/09/2021

Prof. Dr. Eduardo Cotrin Teixeira

Professor Orientador

Prof. Ms. José Antônio Gonçalves

Professor membro da banca

Prof. Dr. Eduardo Filgueiras Damasceno

Professor membro da banca

RESUMO

DELFINO, Wirley de O. **Software de Auxílio Pedagógico para ensino de Consultas Simples a Banco de Dados**. 2021. 42 f. Trabalho de Conclusão de Curso – Curso de Bacharelado em Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

Com a evolução das tecnologias na Área da Tecnologia da Informação, hoje é possível desenvolver aplicações que auxiliam em diversos tipos de processos, como, por exemplo, o processo de ensino/aprendizagem. Neste caso, estas novas tecnologias permitem a criação de softwares que geram ambientes, que embora sejam virtuais, são geralmente interativos, e podem propiciar um maior aproveitamento neste processo. Por este fato este trabalho trata da elaboração de um software para o auxílio no aprendizado de consultas a banco de dados. Por meio dele é possível realizar consultas utilizando a linguagem SQL e, em seguida, mostrar os passos que são realizados para a obtenção do resultado final. Para a implementação da aplicação, foi utilizado o ambiente de desenvolvimento Visual Studio Code e o Sistema Gerenciador de Banco de Dados PostgreSQL. Para a gestão do projeto, utilizou-se a Metodologia de desenvolvimento *Feature Driven Development*.

Palavras-chave: Consulta SQL. Software Educacional. Seleção. Aprendizado.

ABSTRACT

DELFINO, Wirley. **Pedagogical Aid Software for Teaching Simple Queries to Database.** 2021. 42 f. Trabalho de Conclusão de Curso – Curso de Bacharelado em Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

With the advancement of technologies, it is now possible to develop tools that help in various processes, such as the teaching process, creating tools that help learning. Thus, this work deals with the elaboration of a software to aid in the learning of database queries, being a software capable of executing an SQL query and then showing the processes that are carried out to obtain the final result. For the implementation, the Visual Studio Code development environment and the PostgreSQL database were used. For project management, the Feature Driven Development methodology was used.

Keywords: SQL Query. Educational Software. Selection. Learning.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Tipos de softwares educacionais. | 4 |
| Figura 2 - Exemplo de Modelo Entidade Relacional feito a partir do brModelo | 5 |
| Figura 3 – Plano de Execução. | 6 |
| Figura 4 – Exemplo de uma tabela. | 7 |
| Figura 5 – Estrutura do comando SELECT. | 8 |
| Figura 6 – Processamento lógico do comando SELECT. | 8 |
| Figura 7 – Processos da Metodologia FDD. | 10 |
| Figura 8 – Alterando ordem dos atributos para inserção. | 13 |
| Figura 9 - Exemplo do formato a ser utilizado pela tabela Pessoa. | 13 |
| Figura 10 – Tela de criação da tabela. | 15 |
| Figura 11 – Construção da consulta. | 16 |
| Figura 12 – Seleção da tabela. | 16 |
| Figura 13 – Aplicação do filtro. | 17 |
| Figura 14 – Resultado da consulta. | 17 |
| Figura 15 – Caso de uso do usuário. | 18 |
| Figura 16 – Mapeamento de Dependências. | 19 |
| Figura 17 – Caso de uso da criação de tabelas. | 19 |
| Figura 18 – Caso de uso de inserção de registros. | 20 |
| Figura 19 – Caso de uso de consultas. | 21 |
| Figura 20 – Diagrama de Entidade Relacional. | 21 |
| Figura 21 – Diagrama de Atividade. | 22 |
| Figura 22 – Diagrama de Classe. | 23 |
| Figura 23 – Tela inicial da aplicação. | 24 |
| Figura 24 – Tela de cadastro após a inclusão de um campo. | 25 |
| Figura 25 – Tela de cadastro após a inclusão de todos os campos. | 25 |
| Figura 26 – Tela de inserção de registros. | 26 |
| Figura 27 – Exemplo a ser seguido para utilizar arquivo externo. | 26 |
| Figura 28 – Prévia da instrução INSERT a ser realizada. | 27 |
| Figura 29 – Utilização de um arquivo externo para adicionar registros. | 27 |
| Figura 30 – Tentativa de utilizar arquivo com formato inválido. | 28 |
| Figura 31 – Tela de consulta. | 28 |
| Figura 32 – Demonstração dos possíveis campos para exibição em Pessoa. | 29 |
| Figura 33 – Exemplo da consulta que será executada. | 29 |
| Figura 34 – Etapa onde é obtida a tabela. | 30 |
| Figura 35 – Etapa onde é realizado o primeiro filtro na tabela. | 30 |
| Figura 36 – Finalizando a consulta retornando apenas o campo selecionado. | 31 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1 - Requisitos funcionais da Aplicação. | 12 |
| Quadro 2 - Requisitos não funcionais da Aplicação. | 14 |

SUMÁRIO

| | |
|--|----|
| 1 INTRODUÇÃO | 1 |
| 1.1 OBJETIVOS GERAIS | 1 |
| 1.2 OBJETIVOS ESPECÍFICOS | 2 |
| 1.3 ORGANIZAÇÃO DO TRABALHO | 2 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 3 |
| 2.1 SOFTWARES EDUCACIONAIS | 3 |
| 2.2 TRABALHOS RELACIONADOS | 5 |
| 2.3 CONCEITOS DE BANCO DE DADOS RELACIONAL | 7 |
| 3 TECNOLOGIAS E FERRAMENTAS | 9 |
| 4 METODOLOGIA | 10 |
| 5 DESENVOLVIMENTO DA APLICAÇÃO | 12 |
| 5.1 VISÃO GERAL DA APLICAÇÃO | 12 |
| 5.1.1 REQUISITOS FUNCIONAIS | 12 |
| 5.1.2 REQUISITOS NÃO FUNCIONAIS | 14 |
| 5.2 PROTÓTIPOS | 14 |
| 5.3 DIAGRAMAS DE CASOS DE USO | 18 |
| 5.4 DIAGRAMA DE ENTIDADE RELACIONAL | 21 |
| 5.5 DIAGRAMA DE ATIVIDADE | 22 |
| 5.6 DIAGRAMA DE CLASSE | 23 |
| 5.7 EXEMPLO DE USO DA FERRAMENTA | 24 |
| 6 CONSIDERAÇÕES FINAIS | 32 |
| REFERÊNCIAS | 33 |

1 INTRODUÇÃO

Os avanços tecnológicos têm grande influência em todos os setores da Sociedade, inclusive no setor da Educação, no qual pode-se utilizar os recursos adquiridos com estes avanços no auxílio do processo de ensino/aprendizagem. Como exemplo desta influência pode-se citar o desenvolvimento de softwares que realizam simulações de diversos tipos, o que pode facilitar a compreensão de um determinado tema totalmente abstrato.

Pacheco Barros (2013) afirma que, na tarefa de ensinar do professor, há uma abstração de todo o processo que ocorre para alcançar o resultado final, o que pode tornar esta tarefa mais complexa e demorada. Para facilitar a compreensão de tais conceitos, o uso de ferramentas pode auxiliar na demonstração de como uma ação chegou em um determinado resultado (PACHECO, BARROS, 2013).

Nas disciplinas que se aplicam ao ensino sobre banco de dados, professores precisam estabelecer, de forma abstrata, como é o comportamento de uma base de dados. Portanto, neste trabalho, consoante a Pacheco Barros (2013), considera-se importante buscar novos meios que facilitem o processo de ensino/aprendizado e que minimizem as dificuldades que os alunos têm em compreender conceitos por vezes muito abstratos. A escassa quantidade de softwares que auxiliem neste processo motivou o desenvolvimento do presente projeto, que busca facilitar o ensino/aprendizagem nestas disciplinas.

De acordo com os levantamentos bibliográficos feitos neste trabalho, vê-se a necessidade de ferramentas simples, que possam auxiliar e incentivar o aluno, de maneira autônoma, a buscar, a praticar e rever os conceitos aprendidos. Acredita-se que, para o domínio e popularização desta ferramenta, não se deve despende um tempo muito prolongado, pois considera-se o custo de tempo em dominar o uso de ferramentas muito complexas com grandes quantidades de funções pode impactar de forma negativa e desestimular seu aprendizado e seu uso.

1.1 OBJETIVOS GERAIS

O objetivo deste trabalho é a construção de um software capaz de realizar uma simulação demonstrativa de como um Sistema Gerenciador de Bancos de Dados (SGBD) trabalha para projetar o resultado final de uma consulta em um banco de dados. Este apresenta aplicação de filtros, por meio dos quais percebe-se a eliminação de tuplas não requisitadas a realização de agrupamentos e de ordenações até atingir o resultado final esperado.

1.2 OBJETIVOS ESPECÍFICOS

Criar uma aplicação capaz de executar uma simulação demonstrativa, passo a passo de como funciona uma seleção em um banco de dados, uma vez que o processo de construção do resultado não é visível.

Para facilitar a compreensão deste processo, esta ferramenta possibilitará a criação de tabelas com a quantidade de campos desejados, a inserção de valores nas tabelas criadas e por fim, nas seleções, a adição de filtros. O software então realiza a exibição etapa por etapa de como a consulta é executada.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em seis seções, no qual a primeira seção fez uma breve abordagem sobre o tema apresentado no trabalho. A segunda seção apresenta os benefícios de softwares educacionais na educação e conceitos relacionados à banco de dados. A terceira e a quarta seção apresentam as Tecnologias utilizadas para o desenvolvimento da aplicação proposta e a Metodologia de desenvolvimento utilizada. Em seguida, na quinta seção, é apresentada a aplicação desenvolvida. O trabalho é concluído com a seção de Considerações Finais.

2 FUNDAMENTAÇÃO TEÓRICA

A primeira seção apresenta os softwares educacionais, discutindo sua aplicação no ensino. Na seção 2.2, são apresentados os conceitos necessários para o entendimento sobre banco de dados.

2.1 SOFTWARES EDUCACIONAIS

A capacidade de aprender está presente nos indivíduos desde muito cedo. Com o avanço em pesquisas destinadas ao aprendizado, as metodologias de ensino foram aprimoradas, evoluindo de metodologias que se baseavam apenas em memorização para metodologias em que o estudante passe a refletir e a criar novos problemas e soluções (PERRAUDEAU, 2009, p. 13).

Para Sayegh (SAYEGH, 2006) o processo de aprendizagem e de ensino possui um relacionamento bem próximo. A forma que um professor trabalha, inova, cria e estimula um estudante em um determinado assunto, pode aumentar a compreensão e facilita o domínio de um novo conceito.

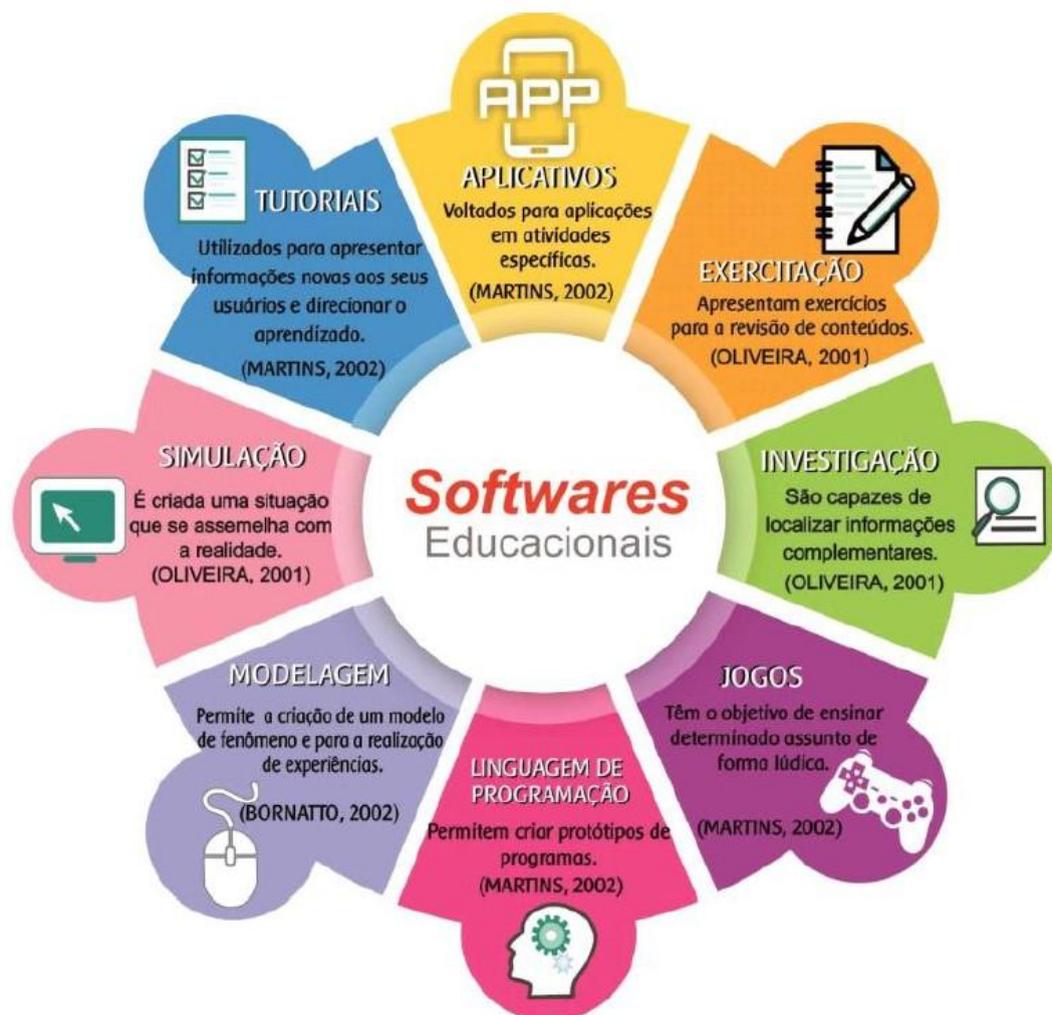
A Tecnologia é muito importante em vários aspectos da vida moderna, no que tange ao processo de ensino/aprendizagem, pode ser identificada por meio das ferramentas auxiliares neste processo, como os softwares voltados para a educação. Uma parcela significativa destes softwares são capazes de proporcionar simulações interativas, o que pode torna o processo de ensinar mais ameno e robusto.

Os softwares educacionais tem sido uma das maneiras de inovar muito adotadas por professores, uma vez que estes softwares são capazes de proporcionar uma experiência que muitas vezes não é possível por meio de livros. Dentre as possíveis experiências pode-se citar as simulações e as animações, que buscam facilitar as explicações.

De acordo com Oliveira (2001), considera-se softwares educacionais todos aqueles que podem ser adequadamente utilizados para ensinar, mesmo que não tenham sido desenvolvidos com esta finalidade.

Existe uma grande variedade de softwares que podem ser utilizados para a educação, como aplicativos de smartphones como o Duolingo, que se trata de um software que auxilia, de forma interativa, no aprendizado de um novo idioma de forma interativa. Há também jogos que são exclusivamente voltados para a Educação, estes procuram tornar o processo de ensinar mais dinâmico. Os tipos de softwares educacionais podem ser observados na Figura 1.

Figura 1 – Tipos de softwares educacionais.



Fonte – (TAVARES, 2017).

Segundo Bellan (BELLAN, 2005) há uma necessidade de apresentar informações através de técnicas de ensino para criar um ambiente adequado para aprendizagem.

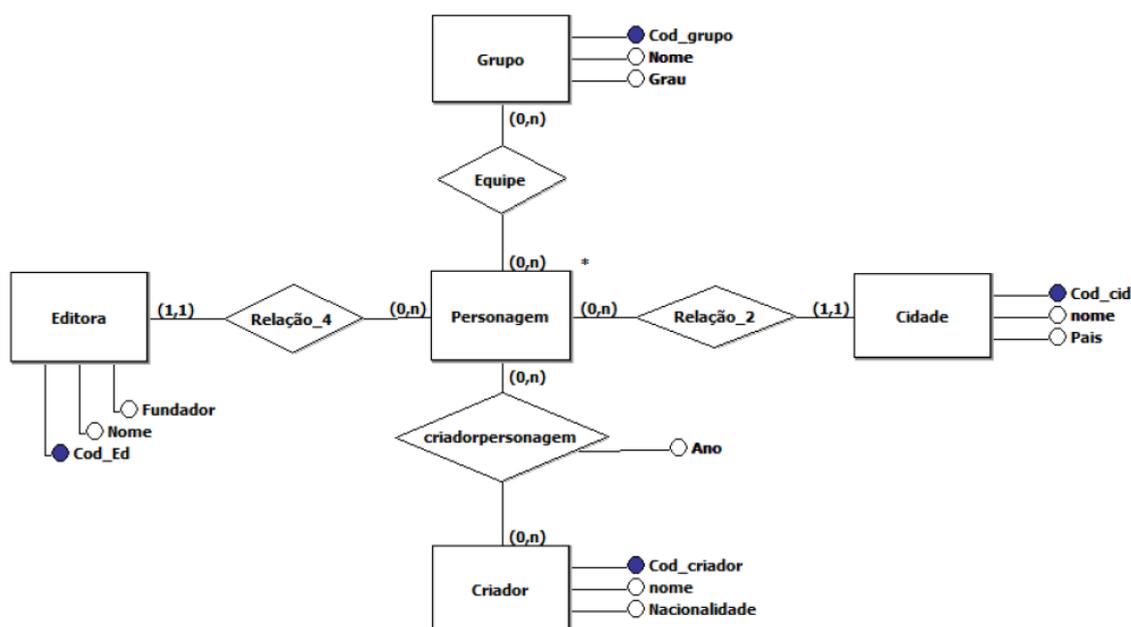
Um dos modelos que facilitam a compreensão de temas mais complexos são os softwares de simulação, que tem como base a abstração de fluxos mais complexos em um fluxo mais simples, que facilita o aprendizado do primeiro contato do aluno com o tema. Esta modalidade de software motivou o desenvolvimento da ferramenta que será apresentada nos próximos seções.

2.2 TRABALHOS RELACIONADOS

Existe uma grande carência de softwares voltados para a didática do ensino do conteúdo sobre banco de dados, de softwares que realizem simulações, demonstrações passo a passo de como ocorre uma operação até chegar no resultado final. Nesta seção se apresentam duas aplicações análogas a que se propõe neste trabalho: brModelo e o Plano de Execução do SGBD SqlServer da empresa Microsoft.

Softwares como brModelo¹ são capazes de apresentar os diagramas denominados Modelos Entidade Relacional, o que pode ajudar a entender os conceitos envolvidos. A Figura 2 apresenta um exemplo deste tipo de diagrama.

Figura 2 - Exemplo de Modelo Entidade Relacional feito a partir do brModelo



* Personagem(Cod_perso, alterego, nm_heroi, apelido, habilidade, arma)

Os desenhos apresentados na Figura 2, descreve notações que deverão ser transpostas para um banco de dados. Nesta transposição deve-se considerar que os retângulos representam as tabelas, ou as relações nos banco de dados e, os losangos as relações entre estas tabelas. A numeração entre cada associação, dependendo de seu formato, pode ser denominada Cardinalidade ou Multiplicidade. Esta numeração descreve a semântica da associação. Em cada retângulo pode-se notar a existência de escritas, estas

¹ <http://www.sis4.com/brModelo/>

representam as colunas da tabela. No Modelo, nota-se que em alguns atributos há pequena circunferência preenchida, isto indica que estas colunas são as chaves primárias e as não preenchidas os demais atributos.

Outra ferramenta capaz de auxiliar no aprendizado de consultas em banco de dados, é a ferramenta de Plano de Execução do SQL Server da empresa Microsoft que é mostrada por meio da Figura 3. Esta ferramenta possibilita visualizar as etapas executadas para uma consulta a fim de propiciar a otimização da consulta. Outro recurso desta ferramenta, é a capacidade de comparar duas consultas, que apresentem o mesmo resultado, mas escritas de maneira diferente, assim permitindo uma análise de recursos que são mais custosos para a execução.

Figura 3 – Plano de Execução.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the SQL query: `INSERT INTO [tblClientes] VALUES (1, 'VITOR DA SILVA E SILVA'), (2, 'JOÃO JOSÉ MARIA DA SILVA'), (3, 'FERNANDO HENRIQUE CARDOSO'); GO` followed by `SELECT codigo_cliente, nome_cliente from dbo.tblClientes;`. The bottom pane shows the Execution Plan for the query. The plan consists of a single step: a **Table Scan** operation. The details for this operation are as follows:

| Property | Value |
|--------------------------------|--|
| Physical Operation | Table Scan |
| Logical Operation | Table Scan |
| Estimated I/O Cost | 0,0032035 |
| Estimated CPU Cost | 0,0000818 |
| Estimated Number of Executions | 1 |
| Estimated Operator Cost | 0,0032853 (100%) |
| Estimated Subtree Cost | 0,0032853 |
| Estimated Number of Rows | 3 |
| Estimated Row Size | 115 B |
| Ordered | False |
| Node ID | 0 |
| Object | [TSQL]::[dbo]::[tblClientes] |
| Output List | [TSQL]::[dbo]::[tblClientes].codigo_cliente; [TSQL]::[dbo]::[tblClientes].nome_cliente |

The status bar at the bottom indicates "Query executed successfully." and "0 rows".

Na Figura 3, é apresentada a imagem da realização de uma consulta que retorna informações sobre uma entidade denominada Cliente. Os dados retornados podem ser observados na descrição da cláusula da “select” SQL: `codigo_cliente, nome_cliente from tblClientes`. Ao realizar a consulta, o Plano de Execução mostra as etapas executadas, onde é um procedimento chamado *Table Scan*, que consiste na leitura completa da tabela. Como resultado do *Table Scan*, pode-se observar dados como número de tuplas solicitadas, tamanho, custo operacional e vários outros.

2.3 CONCEITOS DE BANCO DE DADOS RELACIONAL

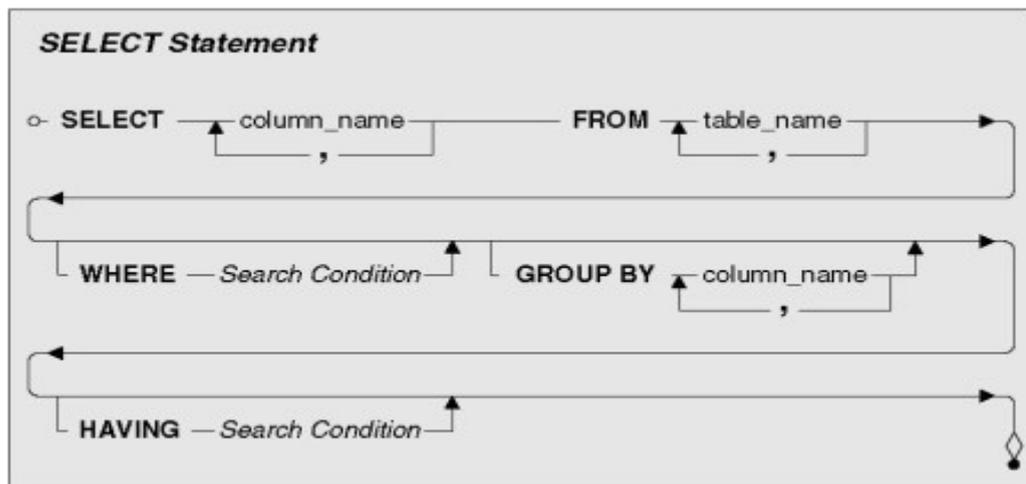
Para uma melhor compreensão do software a ser desenvolvido neste trabalho, é necessário conhecer alguns conceitos relacionados a banco de dados relacional, principalmente quanto a forma como é feita uma seleção em um banco de dados e suas restrições.

Para Korth (KORTH, 2016) um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”. Um banco de dados relacional armazena os dados em tabelas que pode ser observado por meio da Figura 4, formas de se organizar um conjunto de informações relacionadas, por meio de uma linhas e colunas e funciona com um sistema de coordenadas como um plano cartesiano. A linha ou tupla representa um conjunto de valores de todos os atributos de um determinado registro, como por exemplo, uma pessoa, onde temos um nome, sobrenome, data de nascimento entre várias outras características. Já as colunas representam um conjunto de uma mesma informação, como todos os nomes de pessoas que foram cadastradas.

Figura 4 – Exemplo de uma tabela.

| idMidia | idFilme | Idioma | Tipo | Seção | Preço |
|---------|---------|--------|------|-------|-------|
| 1001 | 656565 | dub | DVD | 25 | 3,50 |
| 1002 | 656565 | dub | VHS | 25 | 3,50 |
| 1003 | 656565 | leg | DVD | 25 | 3,50 |
| 1004 | 656565 | leg | VHS | 25 | 3,50 |
| 2550 | 323232 | dub | DVD | 32 | 4,20 |
| 2550 | 323232 | dub | VHS | 32 | 4,20 |
| 2550 | 323232 | leg | DVD | 32 | 4,20 |
| 2550 | 323232 | leg | VHS | 32 | 4,20 |

Em um banco de dados, há comandos capazes de recuperar informações de um banco de dados, ou seja, fazer uma seleção por meio do comando *SELECT*. Ao realizar uma seleção deve-se informar os atributos solicitados, e onde aquela informação está armazenada, ou seja, qual tabela deve ser consultada, o que é informado no *FROM*. Por fim, pode-se definir condições para filtrar os dados a partir do comando *WHERE*. A seguir, na Figura 5 é apresentado a estrutura completa do comando *SELECT*.

Figura 5 – Estrutura do comando *SELECT*.

Fonte: Júnior (2016)

Na Figura 6 é apresentado a execução lógica de execução do comando *SELECT*, onde no primeiro momento, o SGBD encontraria a fonte de dados (tabela) que será necessário para a aplicação do segundo passo. O segundo passo consiste na aplicação de filtros condicionais (*WHERE*) para limitar a quantidade de informações a serem trabalhadas no passo três. No passo quatro, é aplicado os filtros condicionais nos dados processados (*HAVING*). Após todos os filtros aplicados, o SGBD exibe como resultado(*SELECT*), os campos solicitados pelo usuário e ordenados(*ORDER BY*).

Figura 6 – Processamento lógico do comando *SELECT*.

Fonte: Bontempo (2012)

A ferramenta apresentada neste trabalho contempla as etapas: um (*FROM*), dois (*WHERE*) e cinco (*SELECT*).

3 TECNOLOGIAS E FERRAMENTAS

Este capítulo apresenta brevemente as tecnologias e ferramentas que foram utilizadas no decorrer do projeto, sendo que cada uma se adequa a uma categoria.

Gerenciamento de Projeto:

- Google Drive: é um serviço de armazenamento e sincronização de arquivos em nuvem, que abriga várias ferramentas como: o GoogleDocs, Google Slides, Google Sheets e um leque de aplicações.

Prototipação:

- Quant-ux: é um software online para prototipação. Além de possuir a possibilidade de carregar imagens, também é possível construir as interações de uma tela para a outra através do controle de fluxo.

Tecnologias para desenvolvimento:

- ReactJS: biblioteca javascript que otimiza o desenvolvimento *frontend*.
- Electron: framework javascript para desenvolvimento de aplicações desktop.

Ambiente de desenvolvimento:

- Visual Studio Code: é uma ferramenta de edição de texto desenvolvida pela Microsoft.

Repositório:

- GitHub: é um serviço de hospedagem de projetos, servindo como um repositório remoto. É capaz de gerenciar e controlar versões de forma distribuída. Oferece suporte à implantação (deploy), colaboração, gerenciamento de projeto e outros.

Sistema Gerenciador de Banco de Dados (SGBD) e Modelagem:

- brModelo | versão 3.2: ferramenta de código aberto para a diagramação de modelo de banco de dados relacionais, permitindo a visualização em modelo lógico e modelo conceitual.
- PostGreSQL | versão 13.0: é um sistema gerenciador de banco de dados *Open Source* que usa linguagem SQL estendida.
- Astah UML 7.2.0 | versão 37: é uma ferramenta UML para modelagem de diagramas. A ferramenta possui várias funcionalidades, entre elas: comparação de diagramas, modelagem colaborativa, integração com códigos-fonte, modificações no ambiente por plugins e APIs.

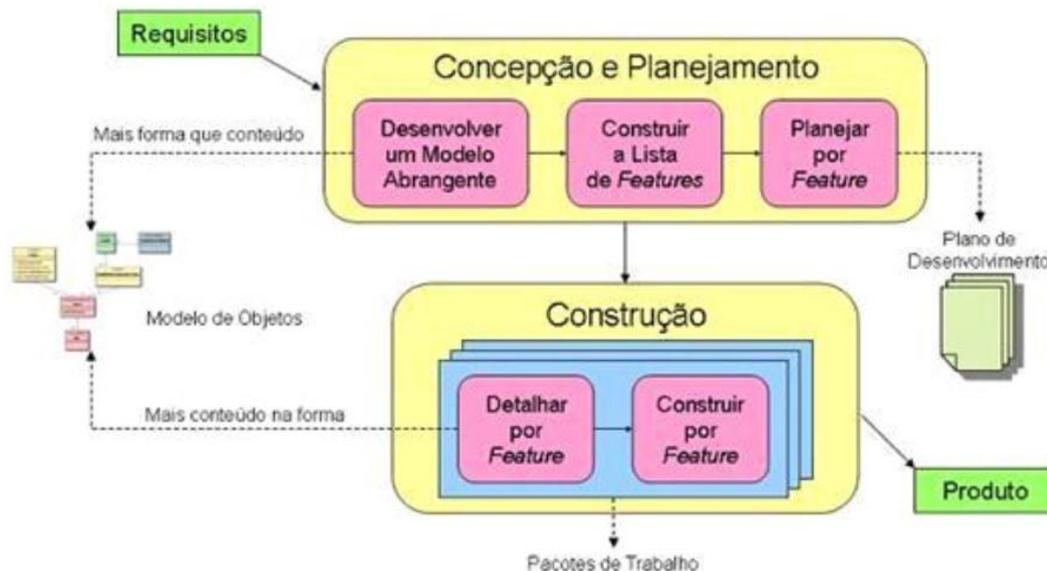
4 METODOLOGIA

A metodologia utilizada para o desenvolvimento do projeto, será a *Feature Driven Development* (Desenvolvimento Baseado em Recursos), também conhecida como FDD. Essa metodologia é adequada para projetos que possuem grande foco em codificação, por ser uma metodologia que atua diretamente com o processo de desenvolvimento, ao contrário do Scrum, por exemplo, que possui um maior foco no gerenciamento do projeto. (ROCHA, 2013).

No FDD, temos o desenvolvimento de requisito a requisito, ou seja, função por função do software a ser desenvolvido, o que torna o gerenciamento de todo o desenvolvimento mais prático e preciso, pois após estabelecer os requisitos funcionais, pode-se analisar de forma mais precisa dados como: requisitos já desenvolvidos, requisitos a serem desenvolvidos, se há ou não há atraso no cronograma.

Na Figura 7 a seguir, pode-se observar que o FDD possui cinco processos básicos, a saber, desenvolver um modelo abrangente, construir a lista de *features* (recursos), planejar por *feature*, detalhar por *feature* e construir por *feature*.

Figura 7 – Processos da Metodologia FDD.



Fonte: Robaski (2014)

Segundo Ambler (AMBLER, 2019), na primeira etapa (“Desenvolver um Modelo Abrangente”, na Figura 3) é feito estudo e apresentação da visão geral do problema e, na etapa “Construir a Lista de Features”, são identificados os requisitos funcionais a partir dessa visão geral.

Na etapa “Planejar por Feature”, são atribuídas uma data de início e uma data de término para cada requisito identificado com base na complexidade e se há ou não dependências, para que na etapa “Detalhar por Feature” sejam construídos todos os diagramas referentes a cada requisito. Por fim, na etapa “Construir por Feature” é realizada a implementação de requisito a requisito.

5 DESENVOLVIMENTO DO PROJETO

Neste capítulo são apresentadas as funcionalidades da aplicação e sua visão geral em conjunto com uma seção de prototipação. Por fim, é apresentada a aplicação e nesta também é apresentado um exemplo de um acesso ao banco de dados. Neste acesso é demonstrado uma tabela, a inserção de dados nesta e a realização de uma consulta a seus dados.

5.1 VISÃO GERAL DA APLICAÇÃO

Esta seção apresenta a elicitação dos requisitos funcionais e dos não funcionais levantados pelo autor para o desenvolvimento da aplicação.

5.1.1 REQUISITOS FUNCIONAIS

Para envolver o aluno com um fluxo comum no uso do banco de dados, como a criação de tabela, a inserção de dados e criação de consultas, as funcionalidades da aplicação foram segmentadas em três áreas distintas, sendo uma área reservada para a criação de tabelas, uma área para inserção de dados nas tabelas criadas e por fim, outra área reservada para a criação de consultas simples (em apenas uma tabela) nas tabelas criadas.

Quadro 1 - Requisitos funcionais da Aplicação.

| ID | Requisito |
|-------|--|
| RFT01 | Deve permitir ao usuário criar tabelas |
| RFI02 | Deve permitir ao usuário inserir registros na tabela via interface |
| RFI03 | Deve permitir ao usuário inserir registros na tabela via arquivo externo |
| RFC04 | Deve permitir ao usuário realizar consultas em nas tabelas cadastradas |
| RFC05 | Deve permitir ao usuário que remova atributos enquanto cria tabela |
| RFC06 | Deve permitir que o usuário navegue entre as etapas geradas |

No processo de criação de novas tabelas, o usuário informa o nome da tabela e adiciona os atributos, sendo disponibilizados alguns tipos primitivos de atributos (integer, decimal, char, varchar entre outros). Enquanto o usuário cria a tabela, ele pode visualizar um exemplo da sintaxe SQL que será executada para a criação da tabela. Antes de

executar a *query* (consulta) gerada, o usuário pode remover atributos e selecionar um dos atributos cadastrados como *primary key* (chave primária).

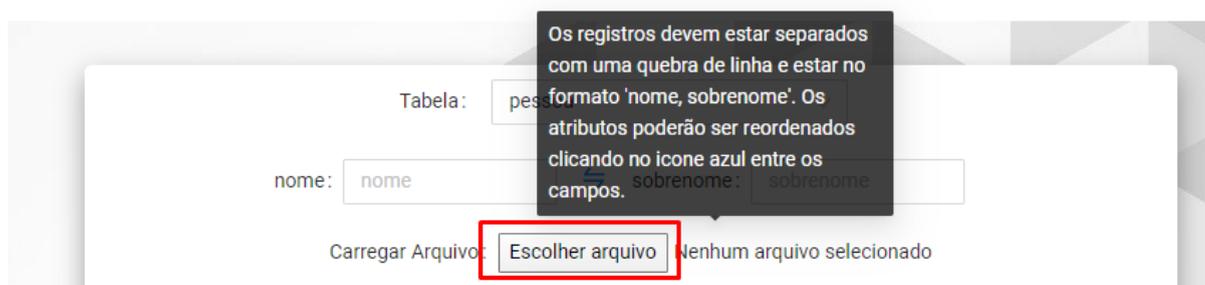
A opção de inserção de registros fica disponível assim que o usuário realiza o cadastro de ao menos uma tabela. Nesta área, o usuário pode selecionar a tabela na qual deseja inserir registros. A inserção de registros pode ser feita de duas maneiras: via interface e por um arquivo com extensão “.txt”. Caso o usuário queira inserir registros via interface pode ser definida a ordem dos atributos como mostrado na Figura 8 abaixo, assim informando os valores respectivamente.

Figura 8 – Alterando ordem dos atributos para inserção.

```
INSERT INTO PESSOA(nome, sobrenome)
VALUES ("NOME_EXEMPLO", "SOBRENOME_EXEMPLO")

INSERT INTO PESSOA(sobrenome, nome)
VALUES ("SOBRENOME_EXEMPLO", "NOME_EXEMPLO")
```

Figura 9 - Exemplo do formato a ser utilizado pela tabela Pessoa.



Se o usuário optar por usar um arquivo, a aplicação apresenta o formato para os valores como apresentado na Figura 9. O formato dos dados no arquivo *txt* deve seguir o formato que o usuário definiu via interface gráfica da aplicação. A aplicação em todo momento, mostrará as *queries* que serão executadas no banco.

Na seção de consultas, o usuário pode criar consultas via interface com operadores condicionais na cláusula *Where*, lembrando que essa seção só é disponibilizada caso o usuário já tenha cadastrado ao menos uma tabela, assim como acontece na seção de inserção. Enquanto o usuário cria uma consulta, é exibida a sintaxe *SQL* que será

executada no banco de dados. Ao executar a consulta, o resultado é apresentado ao usuário simulando um passo a passo da lógica que o SGBD usou para chegar no resultado final.

5.1.2 REQUISITOS NÃO FUNCIONAIS

A fim de melhorar a usabilidade da aplicação foram desenvolvidas algumas funcionalidades que, embora não sejam inerentes à proposta, acabam por tornar a aplicação mais dinâmica. Uma destas funcionalidades trata-se da validação de uma fonte externa de dados, nesta situação, quando o usuário optar por utilizar um arquivo externo como fonte de dados *txt*, será previamente verificando se o formato está correto. Todo campo definido como chave primária terá a propriedade *auto increment* (auto incremento) para garantir a unicidade de cada registro a fim de se evitar que ocorram erros de conflito com a mesma chave. A seguir, podemos, na Tabela 2, são apresentados os requisitos não funcionais.

Quadro 2 - Requisitos não funcionais da Aplicação.

| ID | Requisito |
|-------|---|
| RNF01 | Deve criar atributos do tipo <i>primary key</i> com auto incremento. |
| RNF02 | Deve aceitar arquivos com extensão <i>txt</i> para inserção de dados. |
| RNF03 | Deve validar arquivos <i>txt</i> e informar linhas com erros. |
| RNF04 | Deve exibir um modelo do arquivo <i>txt</i> a ser enviado pelo usuário. |
| RNF05 | Deve manter os registros inseridos via interface pelo usuário ao realizar o envio de arquivo para inserção. |
| RNF06 | Deve exibir um exemplo da sintaxe SQL do comando que será executada |
| RNF08 | Deve exibir um exemplo da sintaxe SQL da consulta que será executada |

5.2 PROTÓTIPOS

Esta seção apresenta os protótipos desenvolvidos para a construção das principais telas do software. Considerou-se aqui o cadastro de uma tabela nomeada como Pessoa e contendo o atributo identificador, "*nome*" e, posteriormente apresenta-se a execução de uma consulta para a exibição do nome referente ao identificador que possui valor igual a um.

A Figura 10 apresenta a tela de cadastro de tabela, onde encontra-se o campo para definir o nome da tabela e outro campo para adicionar atributos com seus respectivos tipos (por exemplo INT ou FLOAT) dentre os tipos primitivos de atributos disponíveis. Para exemplificar esta operação, foi criada uma tabela nomeada como “Pessoa”, contendo dois atributos: um atributo identificador do tipo INT, e um atributo do tipo VARCHAR chamado “Nome”. O atributo identificador é registrado como chave primária.

Figura 10 – Tela de criação da tabela.

The screenshot shows a web interface for creating a table. At the top, there is a navigation bar with four tabs: "Menu", "Tabela", "Select", and "Resultado". The "Tabela" tab is active. Below the navigation bar, there are two main sections. The first section is for defining the table name, with a label "TABELA", an input field containing "Enter a value", and an "Add" button. The second section is for defining attributes, with a label "ATRIBUTO", an input field containing "Enter a value", a "TIPO" dropdown menu currently showing "Option 1", and an "Add" button. Below these sections is an "Executar" button. At the bottom, there is a preview of the SQL code generated: "CREATE TABLE Pessoa" (with a red 'x' icon), "id int" (with a red 'x' icon), "nome varchar(50)" (with a red 'x' icon), and "PRIMARY KEY id" (with a dropdown menu showing "id").

A Figura 11 representa a tela por meio do qual é possível realizar a construção de operações de *SELECT*. Nesta pode-se especificar em qual tabela a operação deve ser realizada. Uma vez que a tabela seja selecionada, todos os atributos referentes à tabela são listados e passíveis de serem escolhidos. Por fim, pode ser realizada a adição de filtros.

Figura 11 – Construção da consulta.

TABELA

ATRIBUTOS

CONDIÇÃO

```
SELECT nome
FROM Pessoa
WHERE id = 1
```

Após a execução da operação *SELECT*, o usuário é direcionado a uma tela onde lhe são apresentadas as etapas lógicas que foram executadas até a obtenção do resultado final da consulta. Como mostrado na Figura 12, as etapas do processo são apresentadas em forma progressiva com o uso do sistema de movimentação de imagens chamado genericamente de Carrossel. Neste exemplo citado, a imagem representa a primeira etapa do processo, onde o SGBD identifica a tabela desejada.

Figura 12 – Seleção da tabela.

SELECT NOME FROM PESSOA WHERE id = 1;

Pessoa

| ID | NOME |
|----|---------|
| 1 | João |
| 2 | Maria |
| 3 | Helôisa |
| 4 | Alex |
| 5 | Amélie |
| 6 | Gabriel |
| 7 | Pedro |
| 8 | César |
| 9 | Luiz |
| 10 | Eduardo |

A Figura 13 mostra a imagem seguinte da que foi mostrada na Figura 12, no sistema de Carrossel o qual permite que o usuário transite livremente entre todas as etapas e, nesta aplicação, cada etapa mostrada no sistema Carrossel tem em seu topo a instrução SQL do processo que foi executado. Na Figura 13 também observa-se a aplicação do filtro, o qual eliminou todas as tuplas que possuem identificadores apresentados na cláusula *where* aplicado na consulta.

Figura 13 – Aplicação do filtro.



Na Figura 14, encontra-se a apresentação do resultado final. O usuário pode transitar livremente entre todas as etapas, e cada etapa contém em seu topo o processo que foi executado.

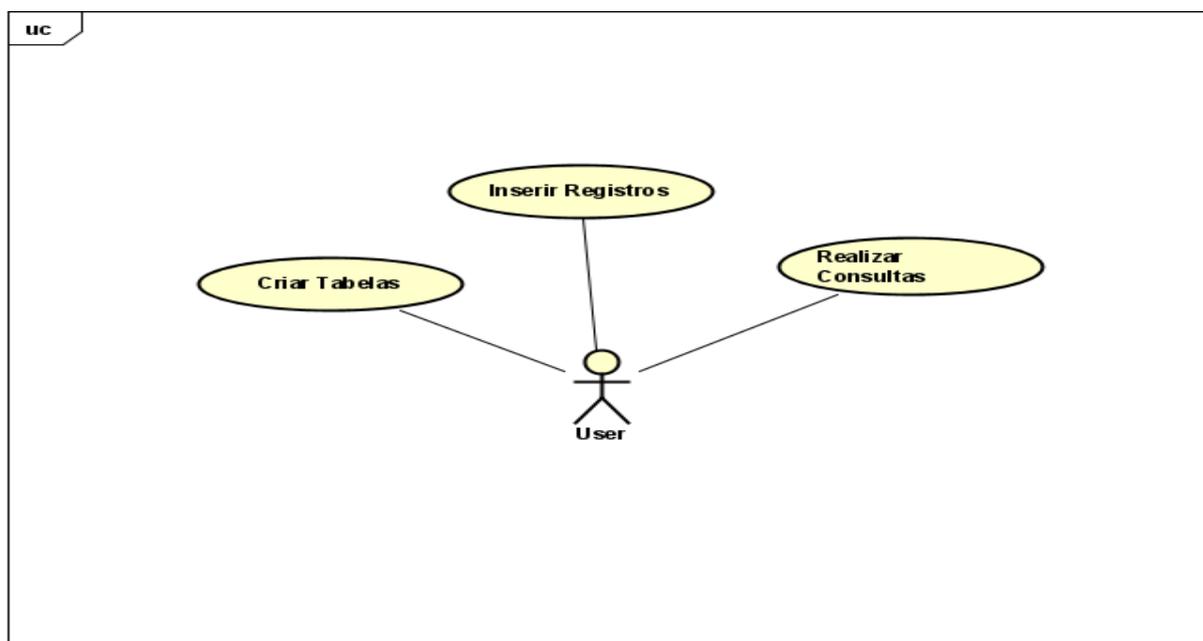
Figura 14 – Resultado da consulta.



5.3 DIAGRAMAS DE CASOS DE USO

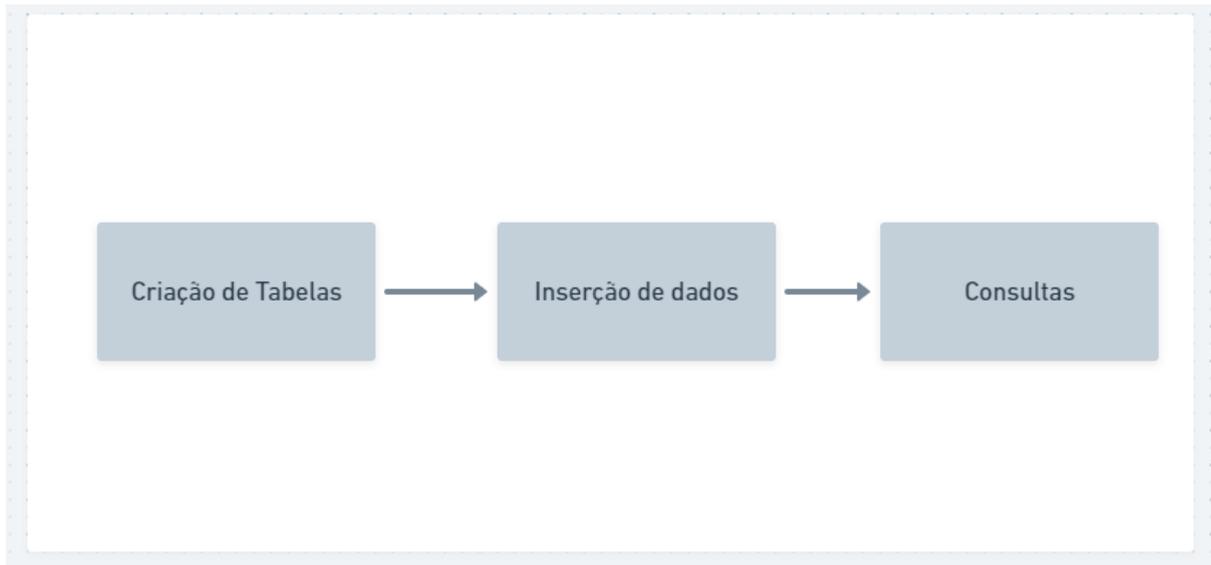
Nesta seção é apresentado os diagramas dos casos de usos utilizados para a implementação da ferramenta seguindo a Metodologia *Feature Driven Development* (Desenvolvimento Baseado em Recursos). Pode ser observado na Figura 15, a apresentação do modelo geral da aplicação, sendo separado em três módulos, onde o usuário pode realizar cadastros de tabelas, inserção de registros nas tabelas e criação de consultas.

Figura 15 – Caso de uso do usuário.



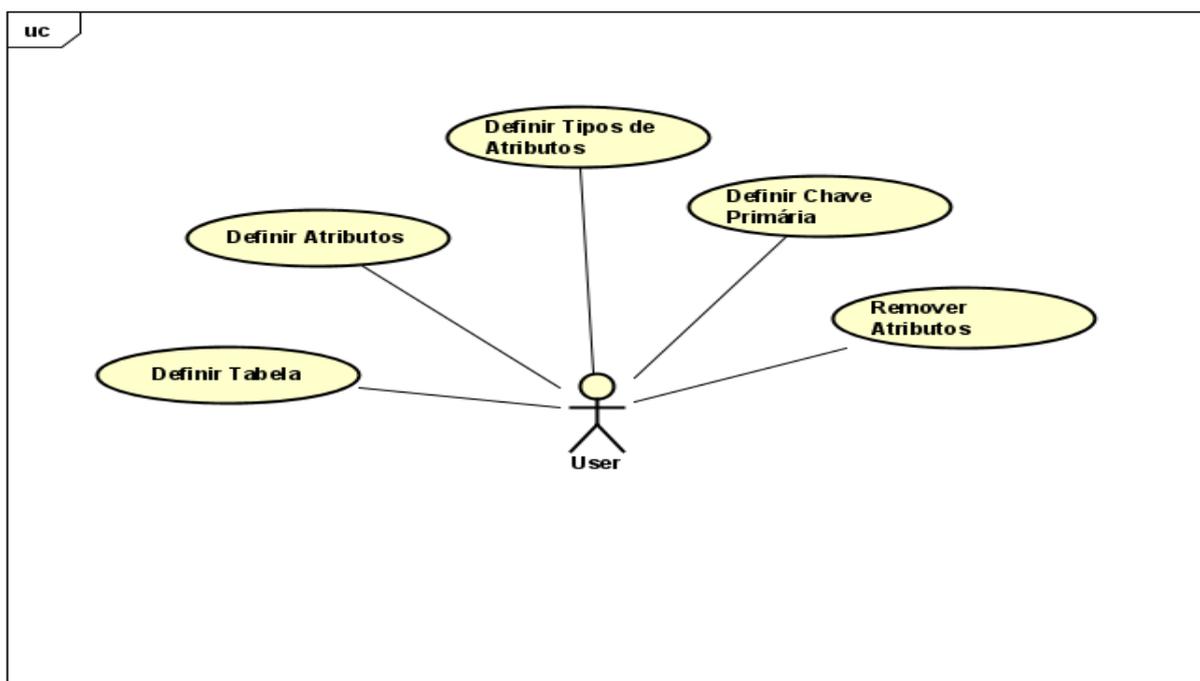
Nos módulos desenvolvidos, têm-se a dependência entre eles, onde é necessário implementar o módulo de Cadastro de Tabelas para posteriormente iniciar a implementação do módulo de Inserção de Registros. O módulo de Consultas possui a maior complexidade entre os três módulos, pois possui dependência do módulo de Inserção que já possui dependência do módulo de criação de tabelas. Com as dependências e complexidades mapeadas, foi definida a ordem de implementação como apresentada na Figura 16.

Figura 16 – Mapeamento de Dependências.



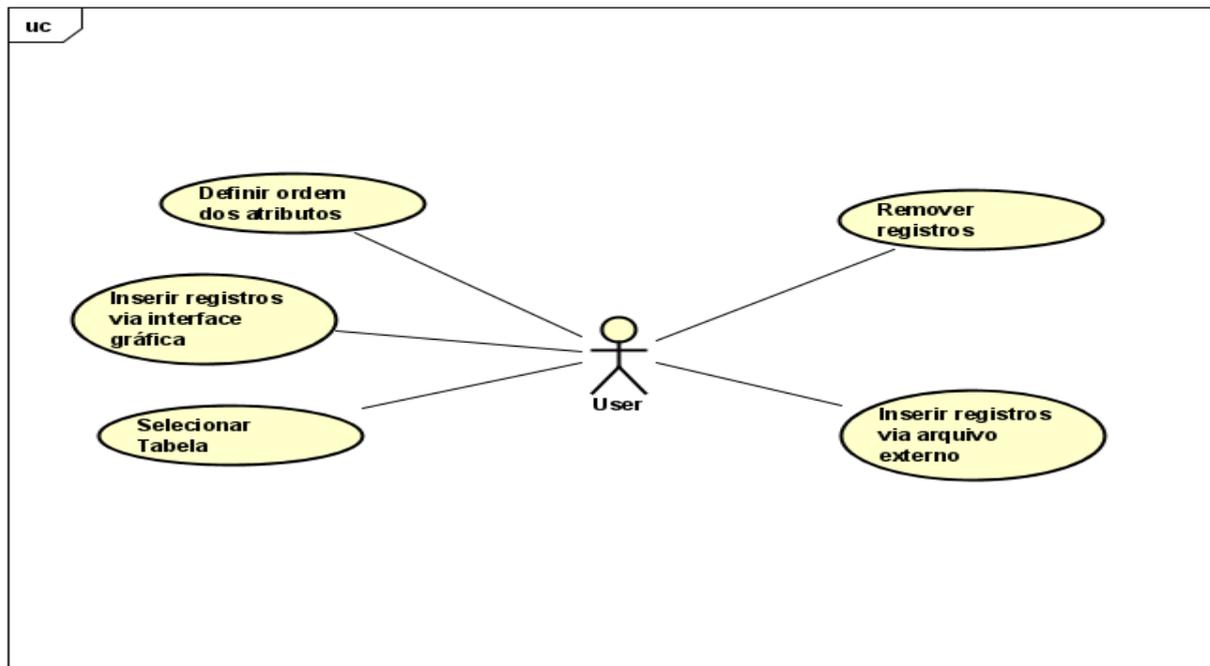
Na Figura 17 a seguir são apresentadas as funcionalidades do módulo de criação de tabelas. Neste módulo é possível que o usuário defina o nome da tabela a ser criada e não permita que o usuário informe um nome de uma tabela que já foi utilizada. A criação de atributos é dinâmica, onde o usuário poderá informar atributos e seus tipos e, caso queira, remover atributos. Por fim, antes da execução do comando para criar a tabela, o usuário poderá informar qual campo será utilizado como chave primária.

Figura 17 – Caso de uso da criação de tabelas.



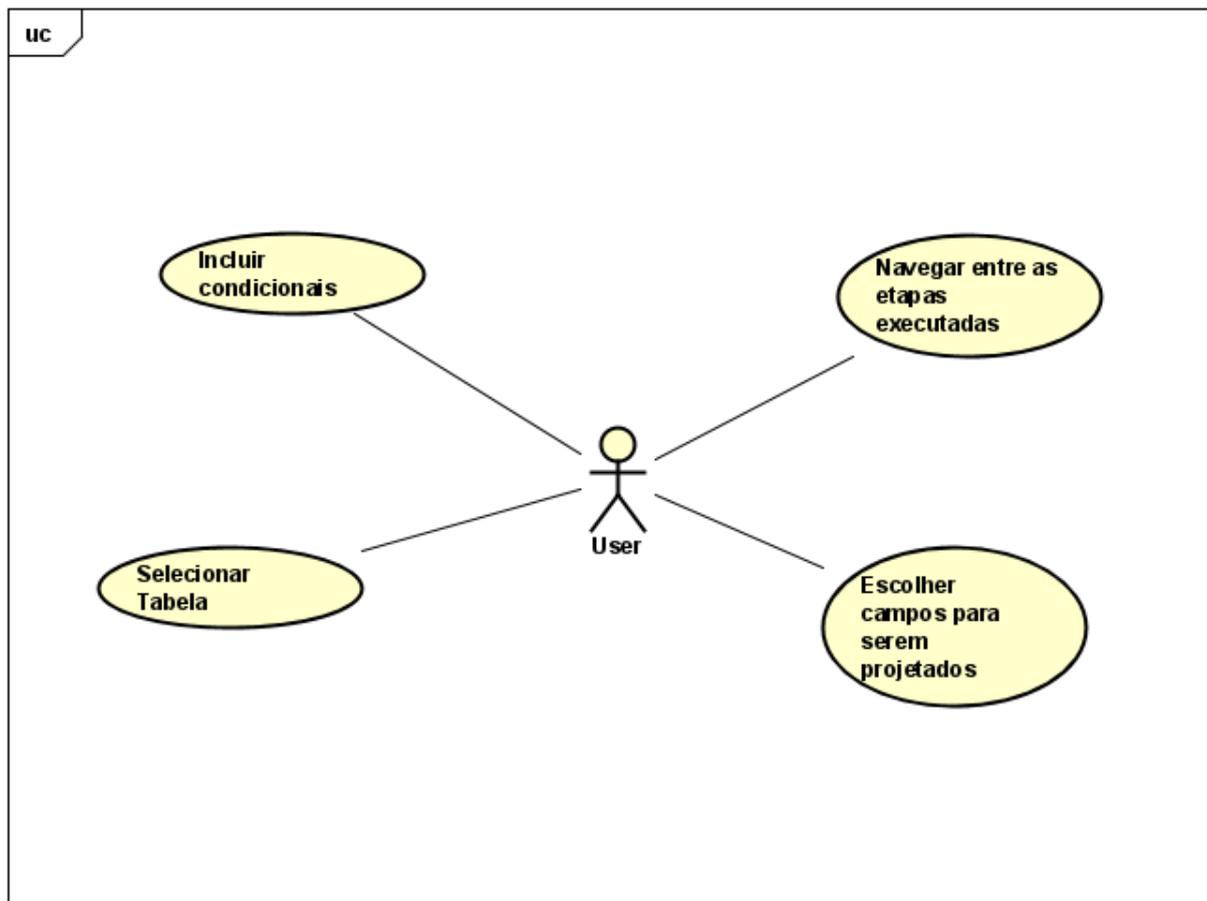
No módulo de inserção de registros (Figura 18), após a escolha da tabela, é possível realizar as inserções de duas maneiras: via interface gráfica e arquivo externo. Para essas opções é possível ordenar os campos conforme desejado. É possível que o usuário remova comandos (*INSERT*) antes da execução na base de dados.

Figura 18 – Caso de uso de inserção de registros.



A Figura 19 apresenta o módulo de Consultas, onde o usuário é capaz de selecionar as tabelas que foram criadas e realizar consultas simples (em apenas uma tabela). Neste módulo, é apresentado ao usuário o modelo de execução lógica do SGBD, onde após ser definida a tabela, os campos a serem apresentados e os condicionais, o usuário poderá navegar entre as etapas geradas.

Figura 19 – Caso de uso de consultas.



5.4 DIAGRAMA DE ENTIDADE RELACIONAL

Esta ferramenta cria tabelas conforme os usuários a utilizam, por este motivo o Diagrama de Entidade Relacional (Figura 20) possui inicialmente duas tabelas base, nomeadas *Tables* e *Attributes*. A entidade *Tables* é responsável por guardar todas as tabelas já criadas para serem exibidas ao usuário nos módulos de Inserção e Consultas em conjunto com seus atributos. As tabelas criadas podem possuir vários atributos, gerando assim um relacionamento de um para muitos com a entidade *Attributes*.

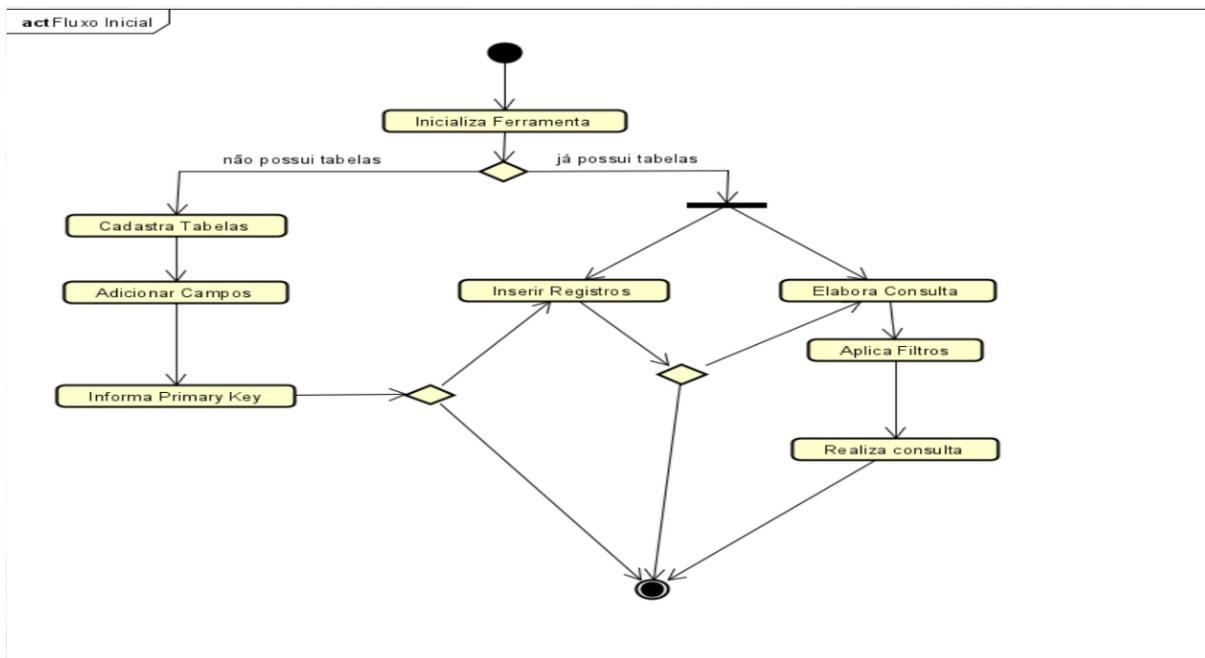
Figura 20 – Diagrama de Entidade Relacional.



5.5 DIAGRAMA DE ATIVIDADE

O Diagrama de Atividade apresentado a seguir demonstra o fluxo de uso da ferramenta, que ao inicializar, pode-se tomar três ações, caso já haja tabelas cadastradas, sendo: criar tabelas, inserir dados e elaborar consultas. Caso não haja nenhuma tabela cadastrada, será necessário a criação de, pelo menos, uma tabela. Caso haja tabelas cadastradas, pode-se inserir registros e realizar consultas. Na realização de consultas, pode-se selecionar os campos a serem buscados, incluir filtros e realizar a consulta.

Figura 21 – Diagrama de Atividade.

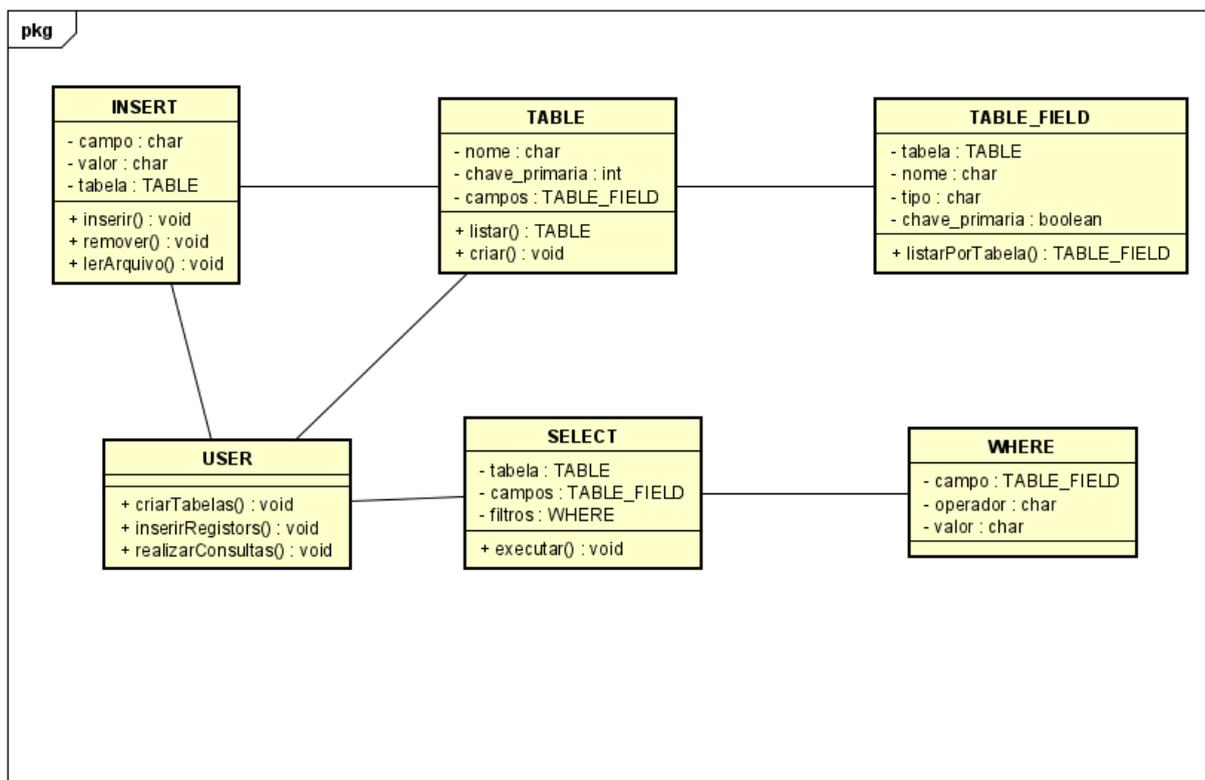


5.6 DIAGRAMA DE CLASSE

Um diagrama de classes é uma representação estrutural e relacional das classes que servem de modelo para objetos, ou seja, é um diagrama que tem o intuito de mostrar estruturas e relacionamentos entre representações de objetos do mundo real, ligados em um contexto como o de um projeto de software, sendo considerada uma modelagem muito útil para desenvolvimento de sistemas, pois define todas as classes que o sistema necessita possuir e, mais especificamente, sua relação com as outras.

A perspectiva usada neste diagrama de classes é destinada ao usuário, que não necessariamente precisa saber quais métodos e atributos serão utilizados na construção do projeto, conforme a Figura 22 que representa os conceitos do domínio em estudo, ou seja, como vai funcionar o projeto e como as classes estão se relacionando.

Figura 22 – Diagrama de Classe.

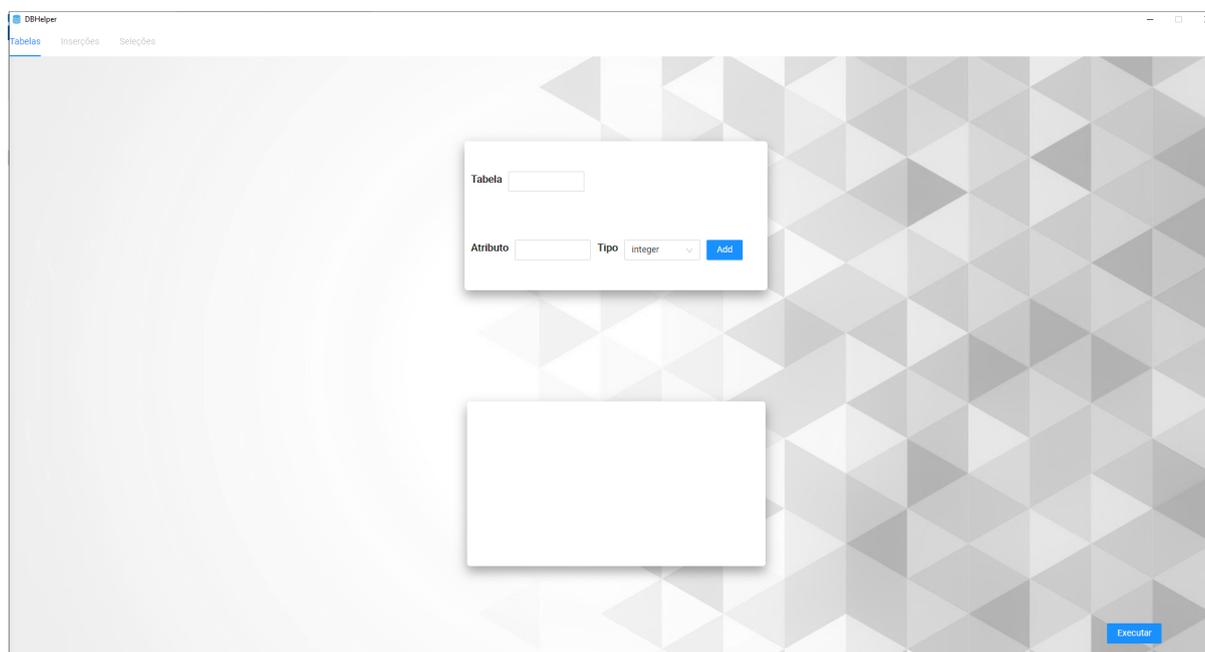


5.7 EXEMPLO DE USO DA FERRAMENTA

Esta seção apresenta um exemplo do funcionamento da aplicação, com a criação da tabela *Pessoa* com dois atributos, *id* e *nome*. O campo *id* é *primary key* do tipo *int*, já o campo *nome* é do tipo *char* com tamanho máximo de cinquenta caracteres.

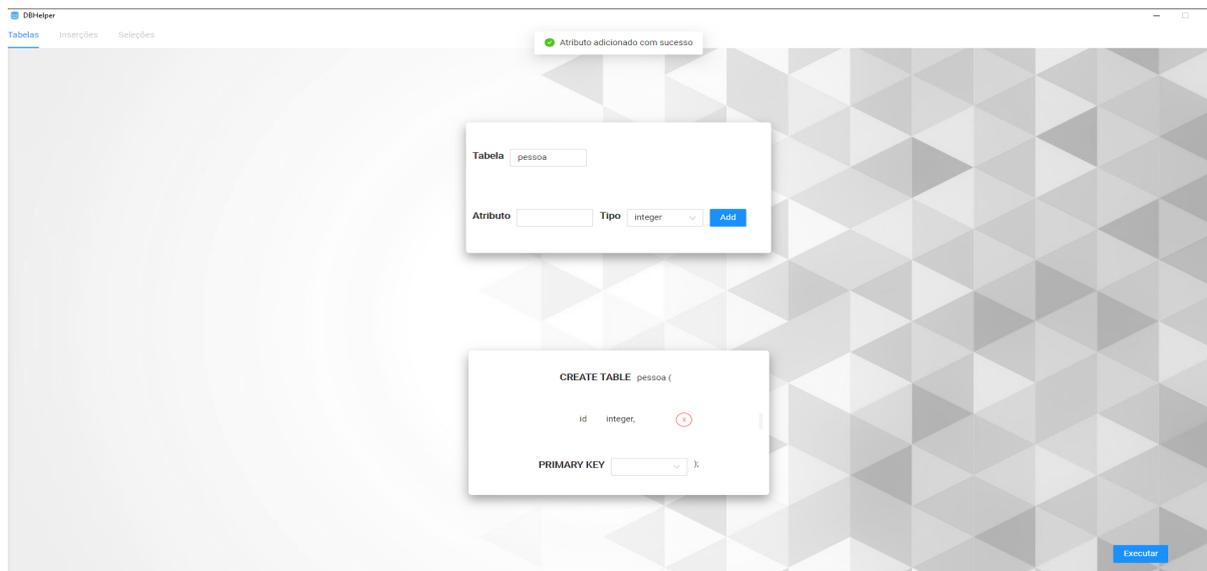
Ao inicializar a ferramenta pela primeira vez, as abas superiores reservadas para inserção de dados e consulta estão desabilitadas pois não há nenhuma tabela cadastrada (Figura 23).

Figura 23 – Tela inicial da aplicação.



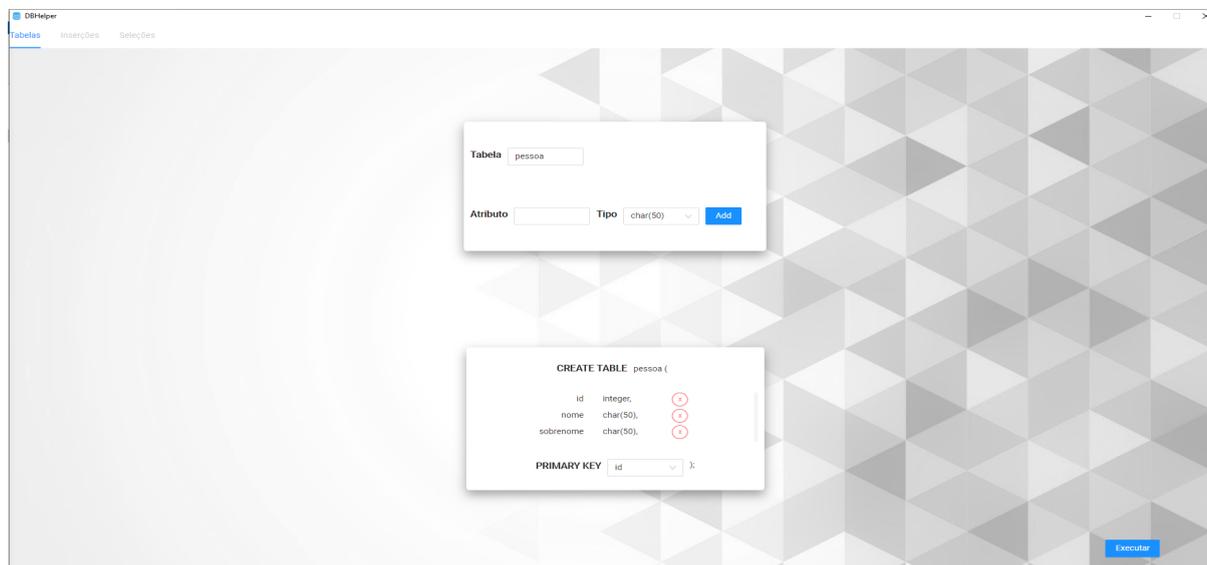
Após o preenchimento do nome da tabela e escolha de um dos atributos, é exibido no quadro abaixo uma prévia do comando na sintaxe *SQL* que será executado no banco relacional (Figura 24). Também é disponibilizado um campo de seleção onde o usuário pode informar a chave primária da tabela, selecionando qualquer campo que tenha sido cadastrado.

Figura 24 – Tela de cadastro após a inclusão de um campo.



No quadro da prévia SQL (Figura 25) é possível remover quaisquer campos e, após a conclusão do preenchimento de todos os campos, pode ser clicado o botão de “Executar” localizado no canto inferior direito para criar a tabela.

Figura 25 – Tela de cadastro após a inclusão de todos os campos.



Ao realizar a criação da primeira tabela, as seções de inserção e consultas são habilitadas. Com isso, na seção de inserção o usuário poderá ordenar os campos, sendo exibida a prévia do *INSERT* a ser realizado (Figura 26). Esta ordem também define o padrão que o arquivo deverá seguir para que possa ser utilizado para popular a tabela conforme apresentado na Figura 27.

Figura 26 – Tela de inserção de registros.

DBHelper

Tabelas Inserções Seleções

Tabela:

nome:

Carregar Arquivo: Nenhum arquivo selecionado

INSERT INTO pessoa(nome, sobrenome) VALUES (...)

Figura 27 – Exemplo a ser seguido para utilizar arquivo externo.

DBHelper

Tabelas Inserções Seleções

Tabela:

sobrenome:

Carregar Arquivo: Nenhum arquivo selecionado

Os registros devem estar separados com uma quebra de linha e estar no formato 'sobrenome, nome'. Os atributos poderão ser reordenados clicando no ícone azul entre os campos.

INSERT INTO pessoa(sobrenome, nome) VALUES (...)

Enquanto o usuário informa os valores para os campos, é exibido no bloco abaixo a instrução completa que será executada (Figura 28). O usuário poderá incluir vários registros para serem adicionados a tabela, inclusive adicionar registros manualmente em conjunto com a adição via arquivo externo. Todos os campos serão exibidos no último bloco com a opção de removê-los. Para adicionar os registros ao banco de dados, o usuário deverá pressionar o botão de executar localizado no canto inferior direito da aplicação.

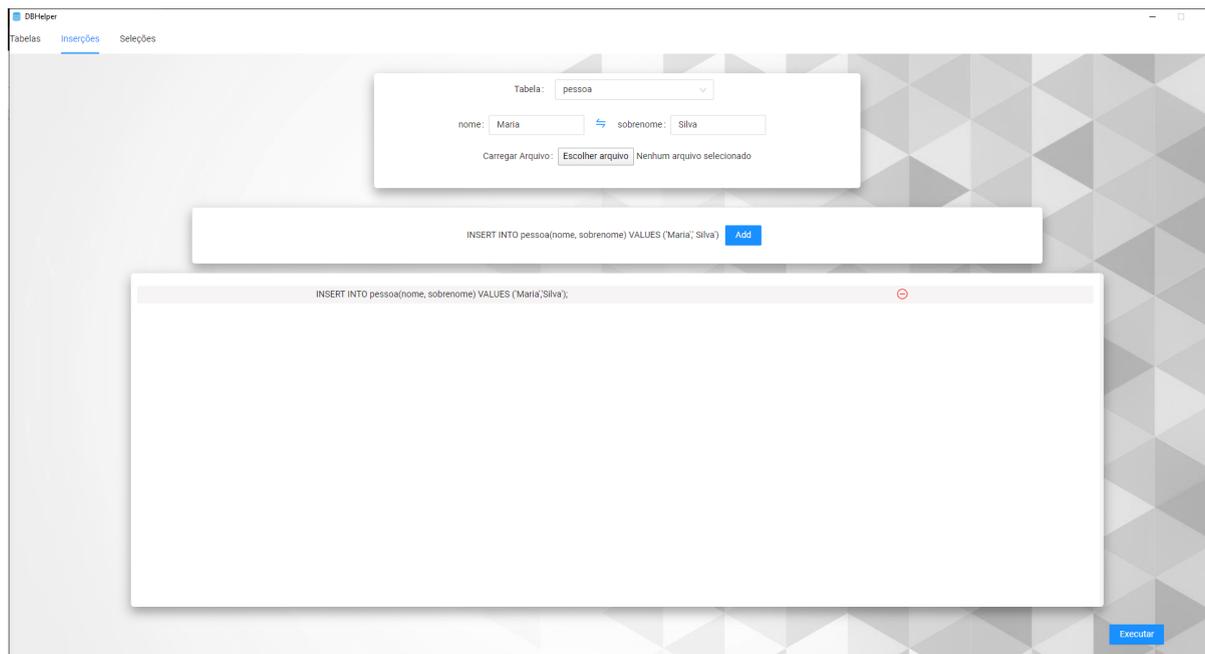
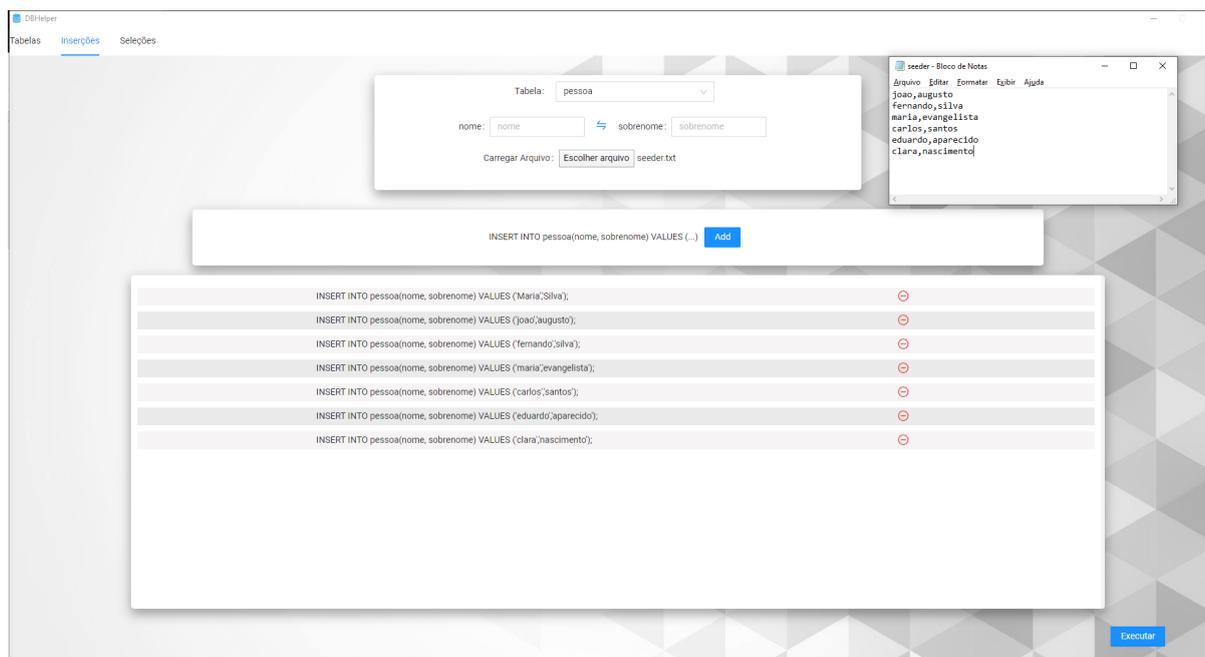
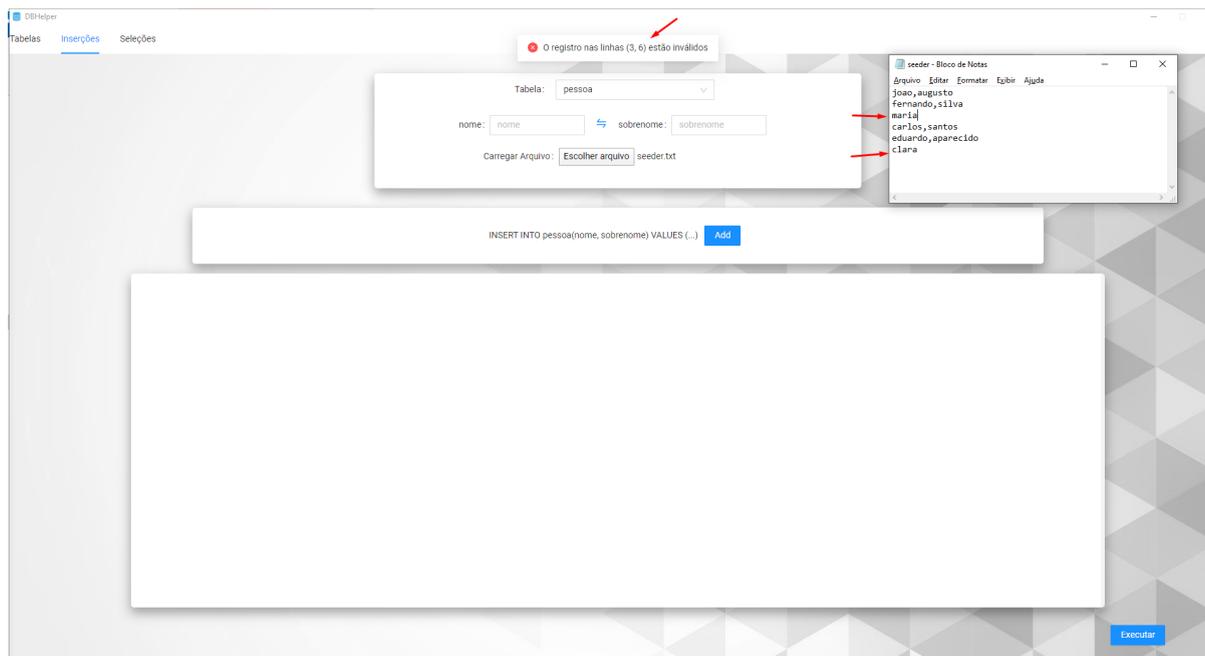
Figura 28 – Prévia da instrução *INSERT* a ser realizada.

Figura 29 – Utilização de um arquivo externo para adicionar registros.



A Figura 29 apresenta a utilização de um arquivo com um formato de dados válido para a inserção dos valores no banco e, caso o usuário utilize um arquivo externo com um formato de dados inválidos, será exibido as linhas que são incompatíveis para serem inseridas na tabela selecionada (Figura 30).

Figura 30 – Tentativa de utilizar arquivo com formato inválido.



Finalmente, a seção de consultas (ilustrada na Figura 31) apresenta todas as tabelas cadastradas pelo usuário para realização de consultas simples, onde no primeiro quadro é possível selecionar quaisquer tabelas que tenham sido cadastradas, os campos a serem exibidos (Figura 32) e os condicionais para a realização da consulta.

Figura 31 – Tela de consulta.

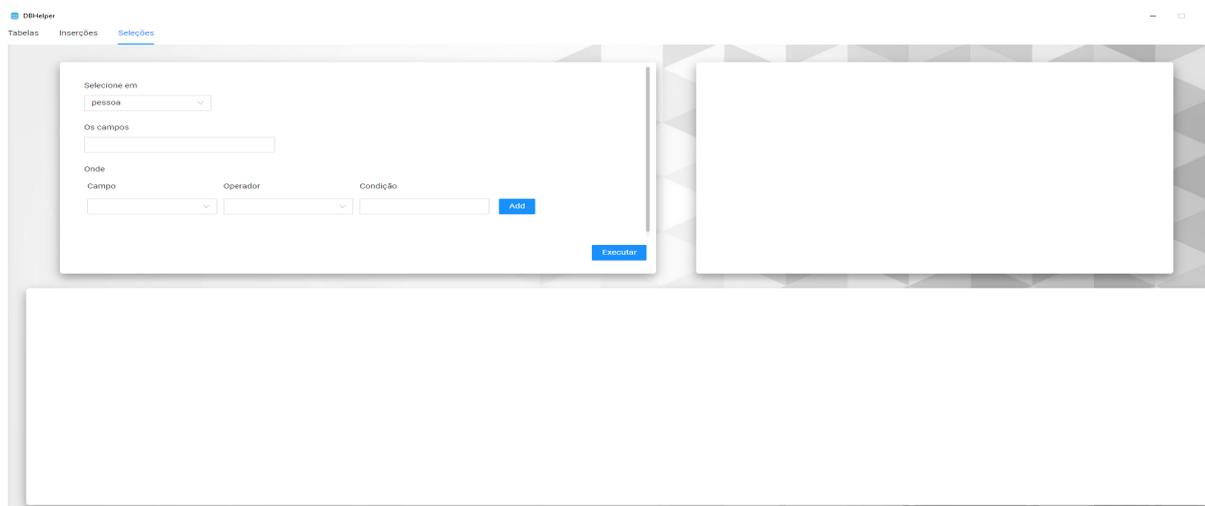
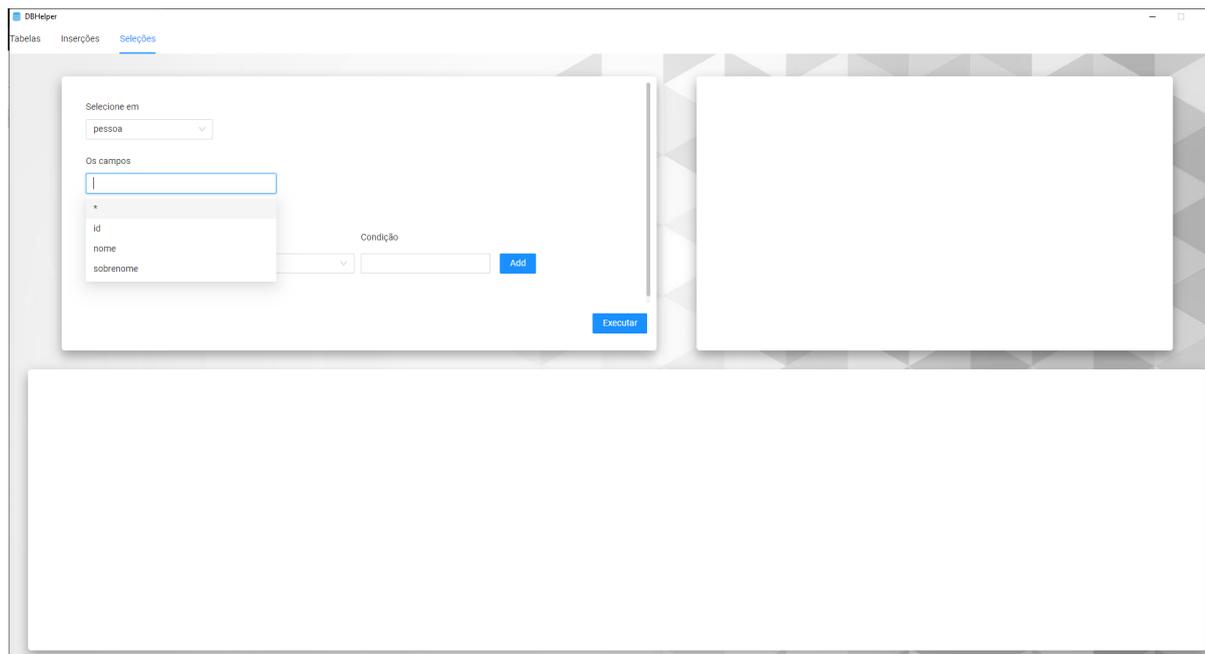
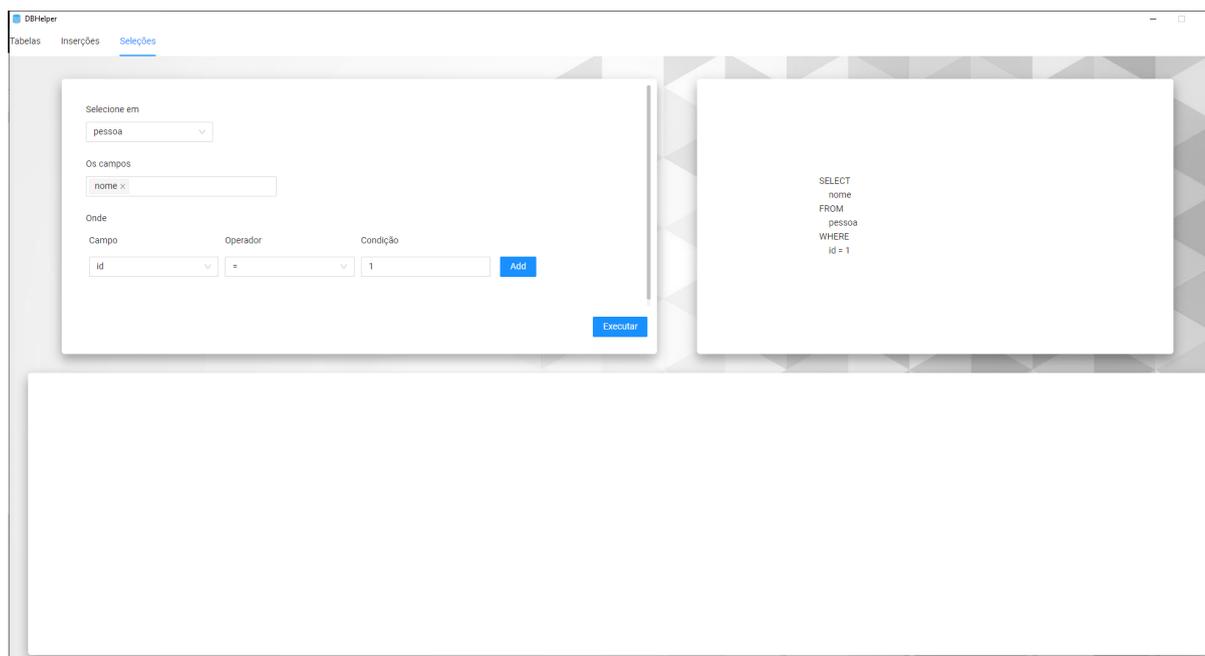


Figura 32 – Demonstração dos possíveis campos para exibição em *Pessoa*.

Após o preenchimento do primeiro quadro com campos e condicionais, o quadro à direita exibe a consulta na sintaxe *SQL* que será executada no banco de dados para a obtenção do resultado (Figura 33). De maneira semelhante a seção de inserção, o usuário também pode remover e adicionar filtros quantos forem necessários.

Figura 33 – Exemplo da consulta que será executada.



Assim que a consulta é executada, o resultado é exibido com divisão de etapas de acordo com a quantidade de cláusulas condicionais utilizadas pelo usuário possibilitando a navegação entre todas as etapas. Com isso é possível revisitar todas as etapas para entender a sequência de instruções executadas e o resultado final da consulta.

A seguir, nas Figuras 34, 35 e 36 é demonstrada a execução de um comando de seleção na tabela Pessoa: `SELECT nome FROM PESSOA WHERE id = 1`.

Figura 34 – Etapa onde é obtida a tabela.

The screenshot shows the DBHelper interface with the 'Seleção' (Selection) tab active. The 'Onde' (Where) clause is configured with 'id' as the field, '=' as the operator, and '1' as the condition. The 'Executar' (Execute) button is visible. The SQL query displayed is:

```
SELECT
nome
FROM
pessoa
WHERE
id = 1
```

Below the query, the results are shown in a table with columns 'id', 'nome', and 'sobrenome'.

| id | nome | sobrenome |
|----|-------------|-----------|
| 1 | augusto | joao |
| 2 | silva | fernando |
| 3 | santos | carlos |
| 4 | evangelista | maria |
| 5 | aparecido | eduardo |

Figura 35 – Etapa onde é realizado o primeiro filtro na tabela.

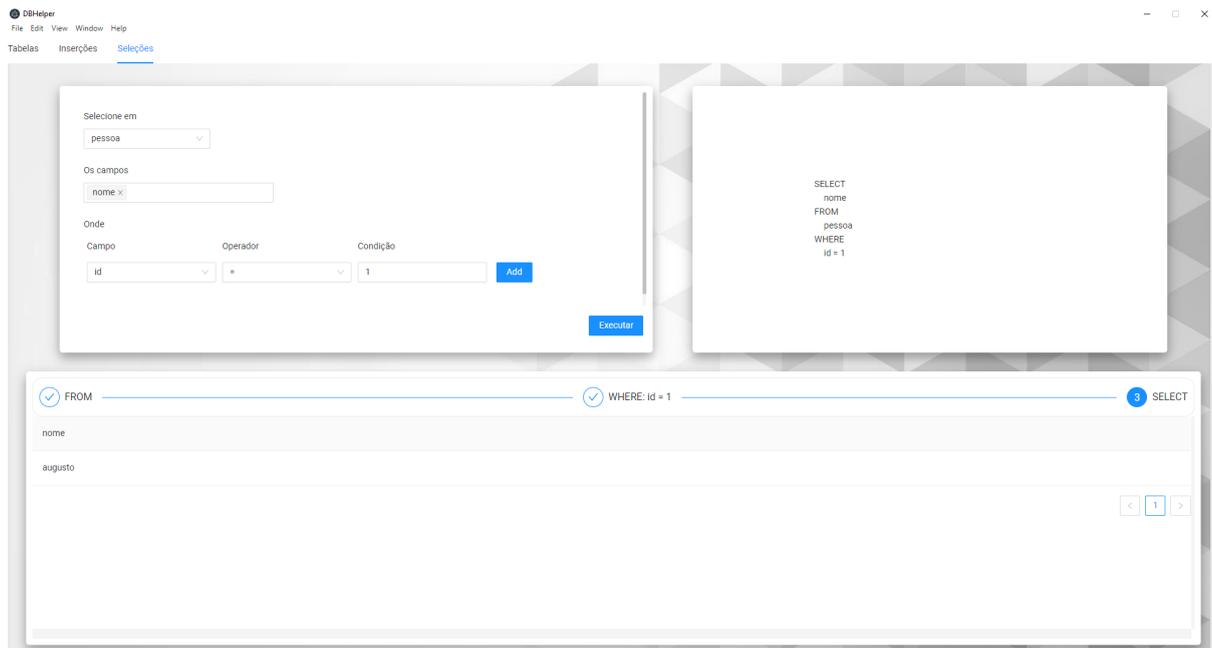
The screenshot shows the DBHelper interface with the 'Seleção' (Selection) tab active. The 'Onde' (Where) clause is configured with 'id' as the field, '=' as the operator, and '1' as the condition. The 'Executar' (Execute) button is visible. The SQL query displayed is:

```
SELECT
nome
FROM
pessoa
WHERE
id = 1
```

Below the query, the results are shown in a table with columns 'id', 'nome', and 'sobrenome'. Only the first row is visible, indicating a filter has been applied.

| id | nome | sobrenome |
|----|---------|-----------|
| 1 | augusto | joao |

Figura 36 – Finalizando a consulta retornando apenas o campo selecionado.



A ferramenta está hospedada em um repositório² do Github. Este repositório é um repositório público, aberto para receber sugestões e melhorias providas pela comunidade. Também possui uma área reservada para dúvidas e problemas encontrados na ferramenta, assim sendo possível realizar manutenções quando necessário.

² <https://github.com/wirleyd/dbhelper-tcc>

6 CONSIDERAÇÕES FINAIS

Como já dizia Sayegh (SAYEGH, 2006), a forma com que um professor trabalha, buscando sempre novos métodos para melhor estimular os alunos, os incentivando a pensar, raciocinar, imaginar, indo além do aprender por repetição, torna o processo de ensinar mais eficiente, estimulando o raciocínio próprio dos alunos.

Assim, com o avanço do mercado tecnológico, tornando possível a disseminação da tecnologia, viu-se uma grande porta para melhorar a forma de ensino, através de softwares educacionais. Estes softwares são capazes de proporcionar uma experiência que antes não existia, possibilitando alunos a interagirem de forma dinâmica.

O uso de softwares educacionais é um método que tem mostrado sua eficiência cada vez mais, desde seu uso com crianças, através de softwares que possui uma maior interação, como jogos para um ensino de forma lúdica, aplicativos que acabam prendendo a atenção por mais tempo até softwares que podem ser voltado para o auxílio de ensino superior, como softwares que realizam simulação que facilita o entendimento de conceitos mais complexos.

Deste modo, há a necessidade em melhorar áreas que carecem de softwares que possam ser utilizados para o ensino, como a de banco de dados. Com a criação deste software, alunos poderão realizar experimentos, criando tabelas, aplicando filtros, adicionando novas tuplas, realizando agrupamentos e assim poder acompanhar a sequência lógica até obter o resultado final da consulta.

REFERÊNCIAS

AMBLER, Scott W.. **Feature Driven Development (FDD) and Agile Modeling**. Disponível em: <<http://agilemodeling.com/essays/fdd.htm>>. Acesso em: 18 de maio de 2019.

BELLAN ZEZINA SOARES. **Andragogia em ação: como ensinar adultos sem se tornar maçante**. Santa Barbara d'Oeste, SP: SOCEP Editora, 2005.

BOMTEMPO, Paulo. **Ordem Lógica de Execução**. Disponível em <<http://phbontempo.blogspot.com/2012/08/ordem-logica-de-execucao.html>>. Acesso em 24 de agosto de 2021

Faculdade de Engenharia da Universidade do Porto. **Metodologia Ágil: Feature Driven Development**. Disponível em: <https://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_22.pdf>. Acesso em: 18 de maio de 2019.

JÚNIOR, Benedito Silva. **SQL - Cláusula SELECT**. Disponível em <https://www.criandobits.com.br/sql/sql_cbasicos-select>. Acesso em 24 de agosto de 2021.

OLIVEIRA, Noé. **Uma proposta de avaliação de Softwares educacionais**. Dissertação (Mestrado em Engenharia de Produção). Florianópolis, Universidade Federal de Santa Catarina, 2001. Disponível em: <<https://core.ac.uk/download/pdf/30362580.pdf>>.

PACHECO, José Adson D.; BARROS, Janaina V. **O uso de softwares educativos no ensino de matemática**. Revista de Estudos Culturais e da Contemporaneidade, Garanhuns, n. 8, p. 5-13, 2013.

PERRAUDEAU, Michel. **Estratégias de aprendizagem: como acompanhar os alunos na aquisição dos saberes**. Artmed Editora, 2009.

ROBASKI, José Ricardo. **FDD (Feature Driven Development)**. Disponível em <<http://medium.com/@jrobaski/fdd-feature-driven-development-7d08c5c24c8f>>. Acesso em 20 de maio de 2019

ROCHA, Fábio Gomes. **Introdução ao FDD: Feature Driven Development**. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971>>. Acesso em: 18 de maio de 2019.

SAYEGH, Flávia. **As relações entre desenvolvimento e aprendizagem para Piaget e Vygotsky**. Recuperado de: [http://www. profala.com/artpsico60. htm](http://www.profala.com/artpsico60.htm), 2006.