**UNIVERSIDADE DO OESTE DE SANTA CATARINA**

**CAMPUS DE SÃO MIGUEL DO OESTE**

**BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO**

LUCAS BARON SPIER

JOÃO GABRIEL DE ABREU

WILLIAN PABLO COLOMBO

**BANCO DE DADOS PARA PROVEDORES DE INTERNET**

São Miguel Do Oeste/SC

2022

LUCAS BARON SPIER

JOÃO GABRIEL DE ABREU

WILLIAN PABLO COLOMBO

BANCO DE DADOS PARA PROVEDORES DE INTERNET

Trabalho Acadêmico do componente de Banco de Dados I do curso de Ciências da Computação da Universidade do Oeste de Santa Catarina no campus de São Miguel do Oeste.

Professor: Roberson Junior Fernandes Alves

São Miguel Do Oeste/SC

2022

**LISTA DE ILUSTRAÇÕES**

**FIGURAS**

[Figura 1: Diagrama lógico relacional 7](https://d.docs.live.net/bd01e062b1d6fde4/Documents/TrabBDI/trabalho%20banco.docx#_Toc107164247)

[Figura 2: Exemplo de criação da tabela atendimento 8](#_Toc107164248)

[Figura 3: Exemplo de inserção da tabela contrato 9](#_Toc107164249)

**SUMÁRIO**

[1 INTRODUÇÃO 5](#_Toc107165136)

[2 DESENVOLVIMENTO 6](#_Toc107165137)

[2.1 PRIMEIROS PASSOS DO PROJETO 6](#_Toc107165138)

[2.2 DIAGRAMA RELACIONAL 6](#_Toc107165139)

[2.3 SCRIPTS 7](#_Toc107165140)

[2.3.1 Script de geração das tabelas 7](#_Toc107165141)

[2.3.2 Script de geração das constraints 8](#_Toc107165142)

[2.3.3 Script de inserção das tabelas 8](#_Toc107165143)

[3 CONCLUSÃO 10](#_Toc107165144)

# 1 INTRODUÇÃO

# 2 DESENVOLVIMENTO

## 2.1 PRIMEIROS PASSOS DO PROJETO

Quando é pensado no desenvolvimento de um banco de dados, sendo este para um provedor de internet, a exemplo do seguinte projeto batizado InsideProvider, é preciso refletir primeiramente nas suas necessidades, para depois partir para o desenvolvimento. Iniciando deste princípio, devemos seguir o processo de criação que foi apresentado durante o decorrer da disciplina, na qual se parte da observação de uma necessidade, neste caso o gerenciamento de um provedor de internet.

## 2.2 DIAGRAMA RELACIONAL

Seguindo para a análise e levantamento de requisitos, foi possível identificar, a partir da experiência de alguns trabalhadores da área de provedor. Com este levantamento, os principais elementos necessários para a etapa seguinte, que foi a elaboração do modelo relacional, foram alcançados, sendo possível definir as entidades e atributos, elencando também as chaves primárias e estrangeiras. Assim como o tipo de dado e suas especificações, como o *not null* (não nulo)e o *unique* (único).

Pode-se também ressaltar a importância de, o diagrama relacional estar normalizado, evitando futuros problemas como inconsistência de dados ou dados repetidos. Seguindo boas práticas através das Formas Normais, como define Moreira(2013):

”Primeira Forma Normal (1NF) Esta regra consiste em verificar se todos os atributos que compõem a tabela são atômicos, ou seja, são atributos indivisíveis.[…],[...]Segunda Forma Normal (2FN) Diz-se que uma tabela está na segunda forma normal se estiver na primeira forma normal e TODOS os atributos dependerem da chave por completo.[…],[...]Terceira Forma Normal (3FN) Esta forma normal introduz o conceito de Dependência Transitiva que consiste em um atributo depender de outro atributo, e este depender diretamente da chave primária.[…]”.

Pensando nas possibilidades futuras de manutenção, implementação e correções que podem ocorrer no banco de dados verifica-se uma grande importância no modelo relacional. Este Será utilizado por diversas pessoas, como por exemplo, programadores ou pessoas que prestam suporte, através da fácil identificação dos componentes, e posterior consulta em banco de dados, migrações e implementação do banco em sistemas mais complexos.

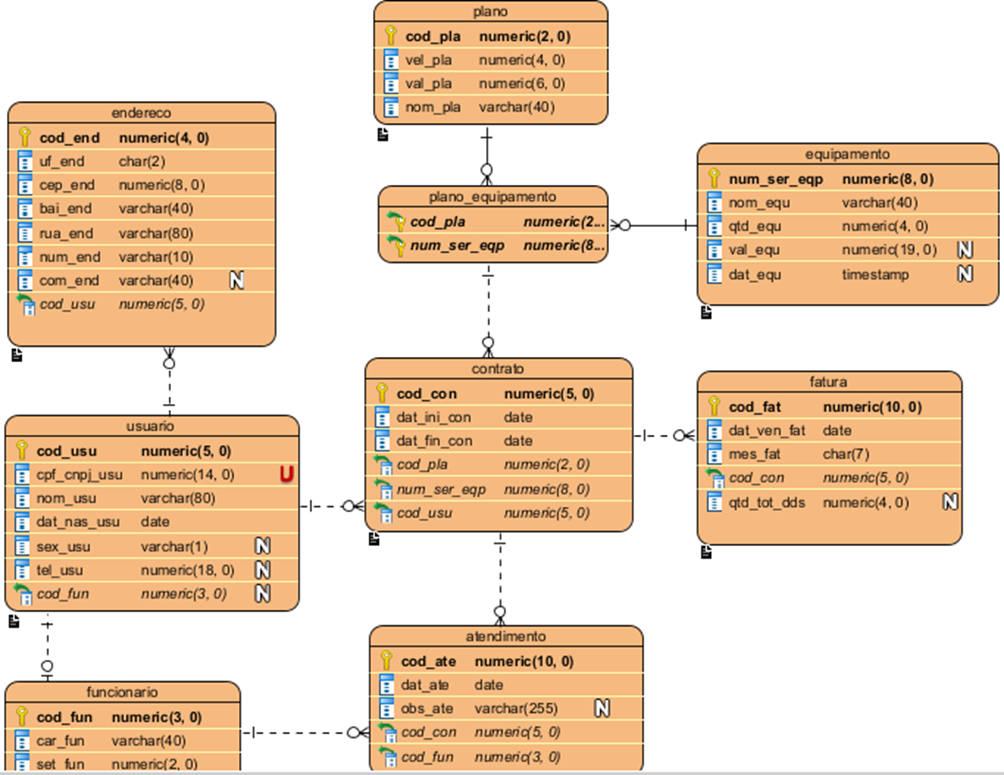
****O modelo lógico pode ser visto na Figura 1.

Figura 1: Diagrama lógico relacional

Fonte: Os autores (2022)

## 2.3 SCRIPTS

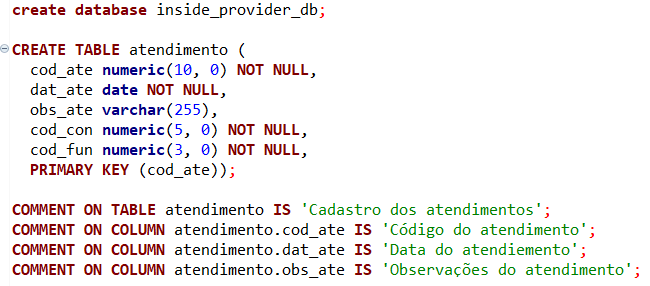
O próximo passo do desenvolvimento, após a conceitualização do projeto no diagrama relacional, é construir a base de dados propriamente dita. Com ajuda de uma função da ferramenta Visual Paradigm (VPP), foi possível criar todo o SQL a partir modelo lógico projetado.

O processo para gerar o *script* no VPP, é feito de maneira simples. Foi selecionado todas as tabelas já terminadas, contendo todos os atributos e relacionamentos necessários para o funcionamento. Com o botão direito do mouse, com as tabelas em destaque, basta selecionar a opção “Generate SQL”. O programa gera todo o SQL de criação das entidades, assim como o usuário especifica, disponibilizando também, um *script* para deletar estes.

### 2.3.1 Script de geração das tabelas

Dentro de todo o código gerado, foi separado em duas partes: tabelas e *constraints*. O *script* com a criação do próprio banco de dados, e as tabelas, foi divido em um arquivo, que pode ser encontrado no repositório do Git com o nome de “create\_database\_tables.sql”. Na figura X, pode ser visto os códigos de criação do *database* e da tabela atendimento, como exemplo, seguidos da adição dos comentários necessários. A criação de todas as tabelas seguiu o mesmo padrão apresentado no exemplo da Figura 2.

Figura 2: Exemplo de criação da tabela atendimento



Fonte: Os autores (2022)

### 2.3.2 Script de geração das constraints

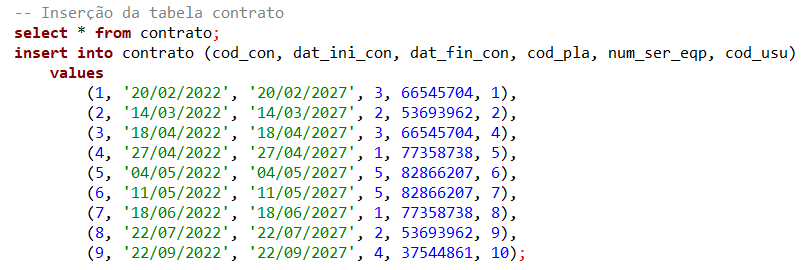
Todas as restrições do tipo chave primária, foram adicionadas junto a criação da tabela. Porém, as regras de integridade, referentes a chaves estrangeiras, foram geradas fora da tabela pelo VPP. Posteriormente, foram separados do *script* principal, para que houvesse uma melhor organização, na qual pode ser encontrado no repositório com o nome “create\_constraints\_FK.sql”.

### 2.3.3 Script de inserção das tabelas

Com a estrutura do banco de dados pronta, o próximo passo seria realizar as consultas solicitadas, para analisar se o *database* continha tudo que era necessário. Mas, para que as consultas pudessem ser feitas, primeiro, as tabelas precisavam ter dados para certificar o êxito dos relatórios.

A inserção ocorreu de forma manual, utilizando-se informações fictícias, geradas de forma aleatória pelo site 4devs. A quantidade de *tuplas* para cada tabela, foi estimado um valor levando em consideração o grau de importância dos dados para as consultas. Na Figura 3, pode ser visto um exemplo de inserção, nessa ocasião, da entidade contrato. O *script* com todos os *insert* das tabelas pode ser encontrado no arquivo “insert\_tables.sql”.

Figura 3: Exemplo de inserção da tabela contrato



Fonte: Os autores (2022)

# 3 CONCLUSÃO

**REFERÊNCIAS**

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS**. NBR 14724: Informação e documentação – Trabalhos acadêmicos** **– Apresentação**. Rio de Janeiro, 2011.

MOREIRA, Flávio Ferry de Oliveira. **Fundamentos de banco de dados**. 2013. 102p. Universidade Federal do Piauí – UFPI, 2013.

7GRAUS. **Ferramentas Online Grátis**. 2022. Disponível em: https://www.4devs.com.br/. Acesso em: 25 jun. 2022.