

**ATIVIDADE BANCO DE DADOS AV PBL**

Turma: 3001

Disciplina: Banco de Dados

Professor(a): Heleno Cardoso da Silva Filho

**EQUIPE**

Victor Fernandes – 202308426044

Sinézio da Silva – 202302375081

Rui Romer – 202302846157

Karine Costa – 202308426079

Rafaela Soares – 202403610876

**Salvador**

**2024**

Turma: 3001

Disciplina: Banco de Dados

Professor(a): Heleno Cardoso da Silva Filho

Trabalho apresentado à UniRuy Wyden  
como requisito parcial de aprovação  
da disciplina Banco de dados  
sob orientação do professor Heleno Cardoso da Silva Filho.

#### **EQUIPE**

Victor Fernandes – 202308426044

Sinézio da Silva – 202302375081

Rui Romer – 202302846157

Karine Costa – 202308426079

Rafaela Soares – 202403610876

**Salvador**

**2024**

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>4</b>
<b>2. MODELO CONCEITUAL.....</b>	<b>4</b>
<b>2.1 DIAGRAMA ENTIDADE-RELACIONAMENTO (ER).....</b>	<b>5</b>
<b>4. MODELO FÍSICO.....</b>	<b>7</b>
<b>5. CONCLUSÃO.....</b>	<b>23</b>
<b>6. REFERÊNCIAS.....</b>	<b>24</b>

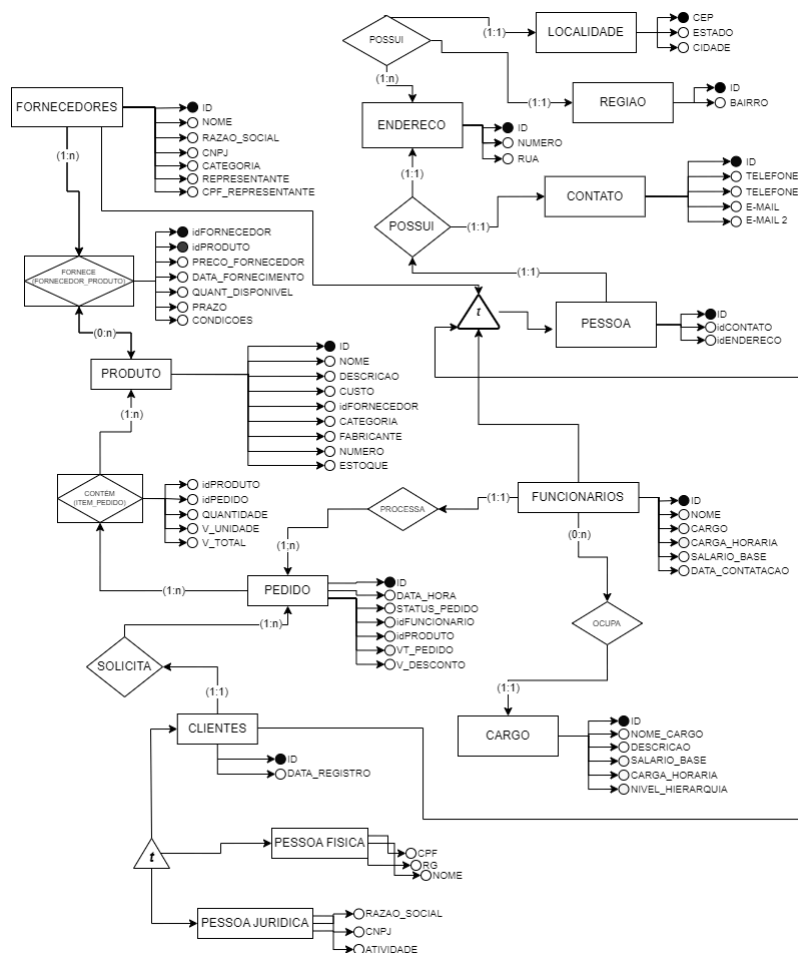
## 1. Introdução

O presente trabalho tem como propósito apresentar um projeto de banco de dados, que será abordado em várias fases da modelagem, tais como o Modelo Conceitual, Diagrama Entidade-Relacionamento, Modelo Lógico, Modelo Físico e Normalização. O entendimento dos conceitos de banco de dados é fundamental para compreender como essa tecnologia é aplicada para armazenar e gerenciar informações cruciais em diversos contextos.

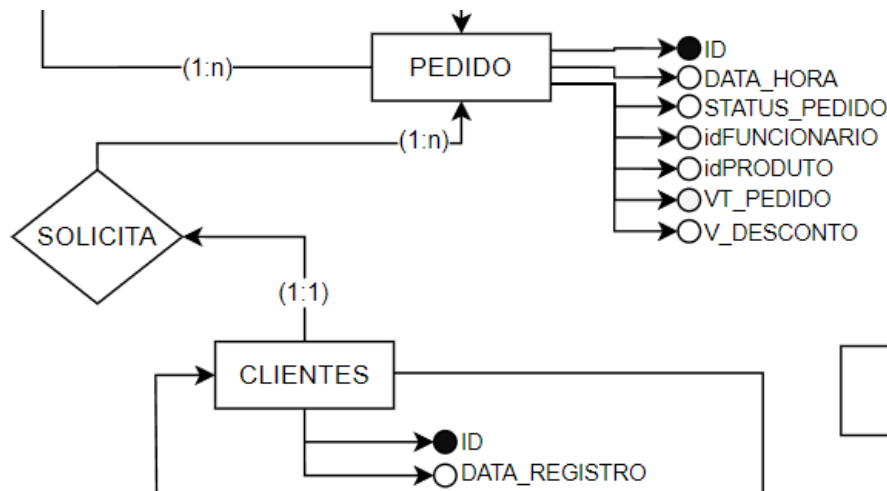
## 2. Modelo Conceitual

O presente modelo conceitual trata-se da representação do banco de dados destinado ao modelo de domínio escolhido, Loja de Peças e Acessórios Automobilísticas. O objetivo deste projeto, visa principalmente o armazenamento integral de todas as informações que compõem seu funcionamento sistêmico, como também organizar e facilitar o acesso a dados críticos para a operação eficiente da empresa, melhorando a gestão no processo de compras, vendas, e recursos humanos.

### 2.1 Diagrama Entidade-Relacionamento (DER)

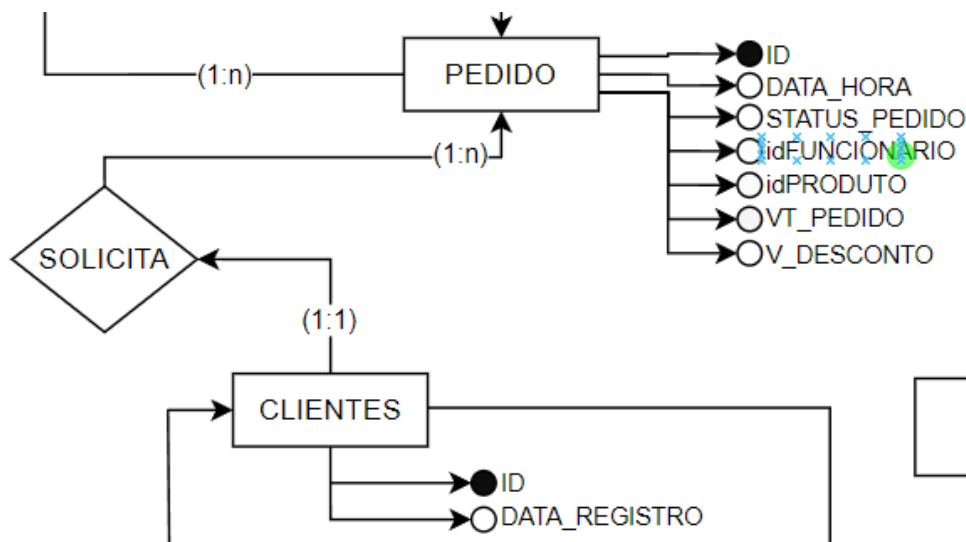


Tendo em vista a **Figura 1**, podemos observar o **Diagrama de Entidade-Relacionamento**, nota-se as seguintes cardinalidades e especializações.



*Figura 2: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.*

Na **Figura 2**, podemos verificar a existência de uma **especialização exclusiva** duas entidades filhas “PESSOA FÍSICA” e “PESSOA JURÍDICA”, com atributos específicos para a entidade pai “CLIENTES”.



*Figura 3: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.*

Na **Figura 3**, verificamos a relação entre as entidades CLIENTES e PEDIDO:

**CLIENTES:** No mínimo 1 cliente pode solicitar 1 ou vários pedidos. (1:n)

**PEDIDO:** No mínimo 1 pedido pode ser solicitado por apenas 1 cliente. (1:1)

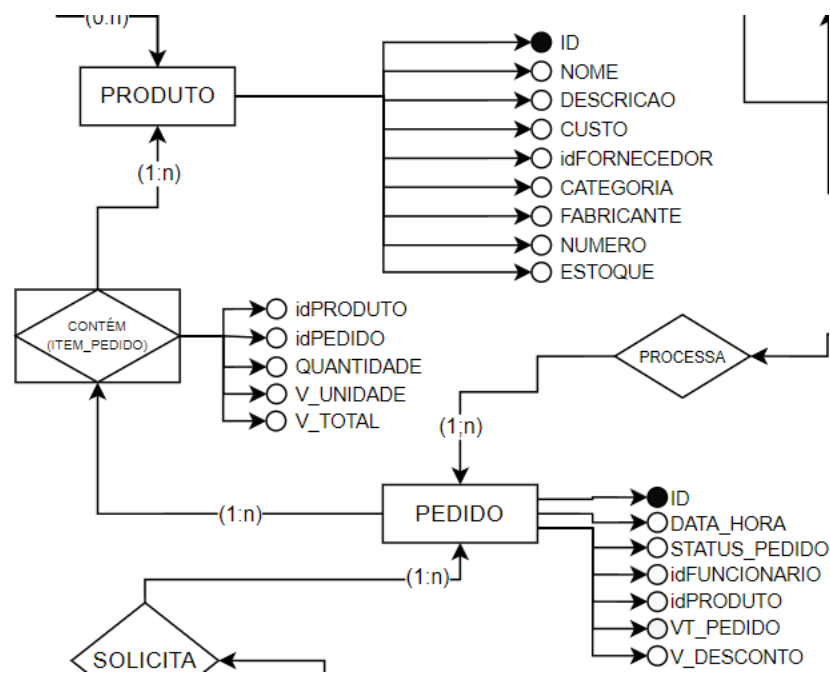


Figura 4: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.

Na **Figura 4**, verificamos a relação entre as entidades PEDIDO e PRODUTO:

**PEDIDO:** No mínimo 1 pedido pode conter 1 ou  $n$  produtos. (1:  $n$ )

**PRODUTO:** No mínimo 1 produto pode estar contido a nenhum ou  $n$  pedidos. (0:  $n$ )

**ITEM\_PEDIDO:** Entidade Associativa gerada na relação “ $n:m$ ”, das entidades “PEDIDO” e “PRODUTO”.

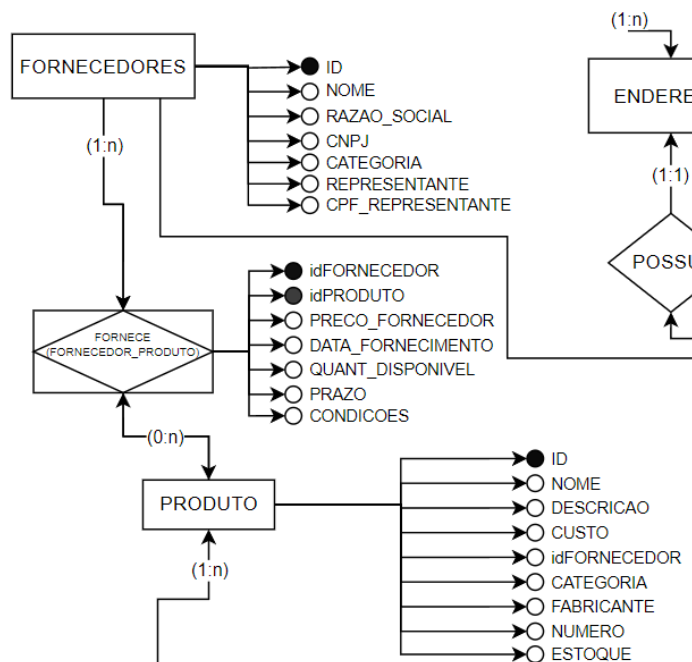


Figura 5: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.

Na **Figura 5**, verificamos a relação entre as entidades PRODUTO e FORNECEDORES:

**PRODUTO:** No mínimo 1 produto é fornecido por 1 ou  $n$  fornecedores. (1: $n$ )

**FORNECEDOR:** No mínimo 1 fornecedor fornece nenhum ou  $n$  produtos. (0: $n$ )

**FORNECEDOR\_PRODUTO:** Entidade Associativa gerada pela relação “ $n:m$ ” das entidades “PRODUTO” e “FORNECEDOR”.

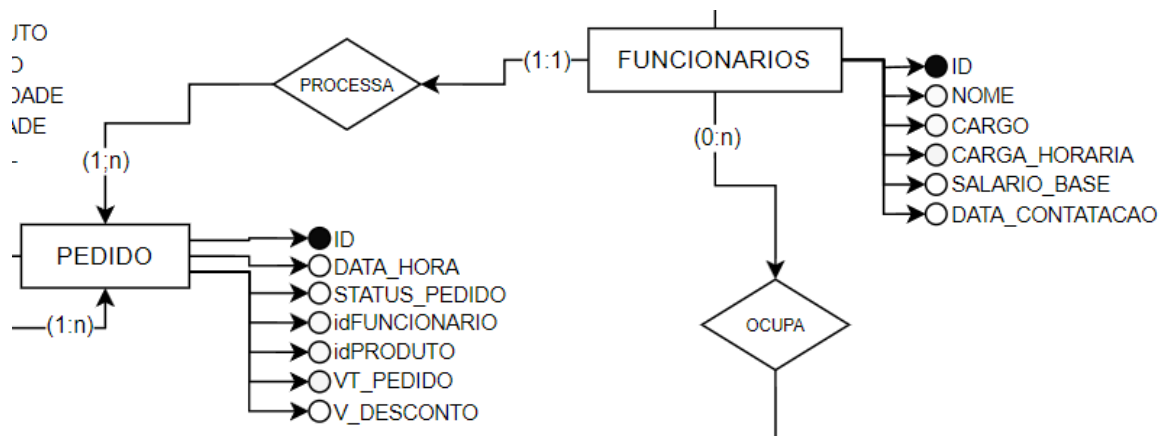
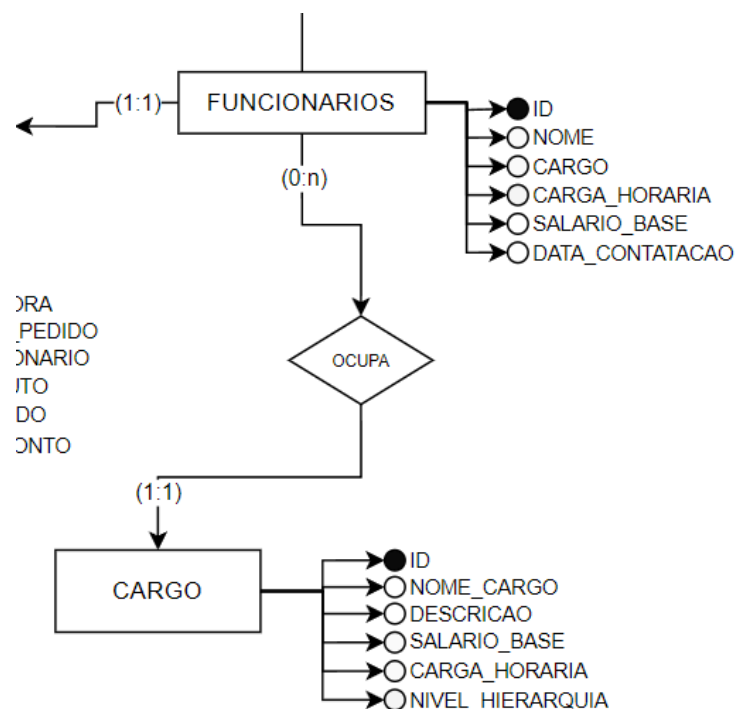


Figura 6: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.

Na **Figura 6**, verificamos a relação entre as entidades PEDIDO e FUNCIONARIOS:

**FUNCIONARIOS:** No mínimo 1 funcionário processa 1 ou  $n$  pedidos. (1: $n$ )

**PEDIDO:** No mínimo 1 pedido é processado por apenas 1 funcionário. (1:1)



7 Figura 7: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.

Na **Figura 7**, verificamos a relação entre as entidades FUNCIONARIOS e CARGO:

**FUNCIONARIOS:** No mínimo 1 funcionário pode ocupar apenas 1 cargo. (1:1)

**CARGO:** No mínimo 1 cargo pode ser ocupado por nenhum ou vários funcionários. (0:n)

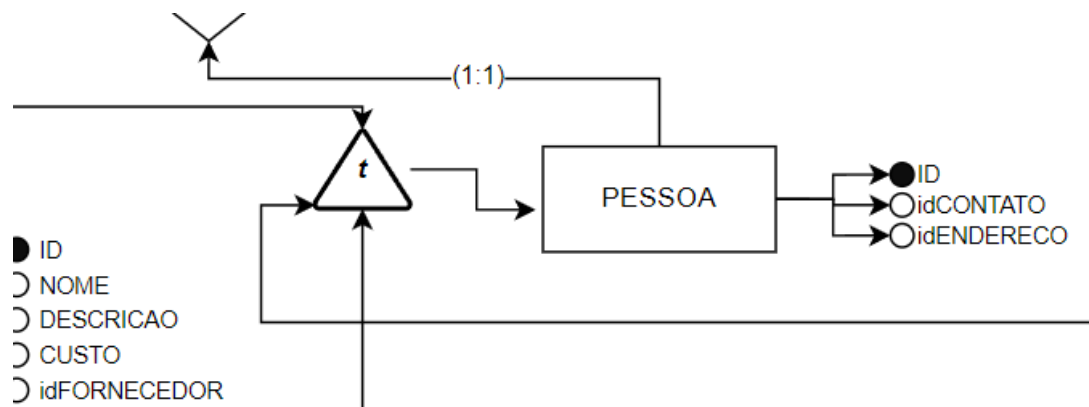


Figura 8: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.

Na **Figura 8**, podemos verificar a existência de uma **especialização não exclusiva** das entidades filhas “CLIENTES”, “FUNCIONARIOS”, e “FORNECEDORES”, com atributos específicos para a entidade pai “PESSOA”.

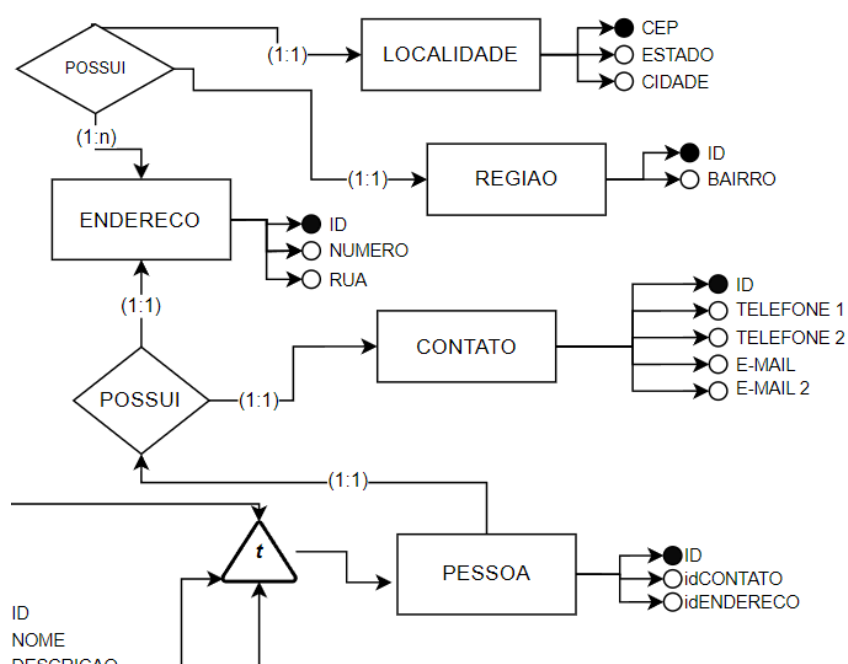


Figura 9: Diagrama ER (Cardinalidade e Relações). Fonte: Autoria Própria.



Na **Figura 9**, verificamos a relação entre as entidades PESSOA, CONTATO E ENDERECO:

**PESSOA:** No mínimo 1 pessoa pode possuir apenas 1 endereço, e 1 contato. (1:1)

**CONTATO:** No mínimo 1 contato pode ser possuído por apenas 1 pessoa. (1:1)

**ENDERECO:** No mínimo 1 endereço pode ser possuído por apenas 1 pessoa. (1:1)

Ainda nesta figura, também podemos notar a relação entre as entidades ENDERECO, LOCALIDADE e REGIAO:

**ENDERECO:** No mínimo 1 endereço possui apenas 1 localidade, e 1 região. (1:n)

**LOCALIDADE:** No mínimo 1 localidade possui 1 ou  $n$  endereços. (1:1)

**REGIÃO:** No mínimo 1 localidade possui 1 ou  $n$  endereços. (1:1)

### 3. Modelo Lógico

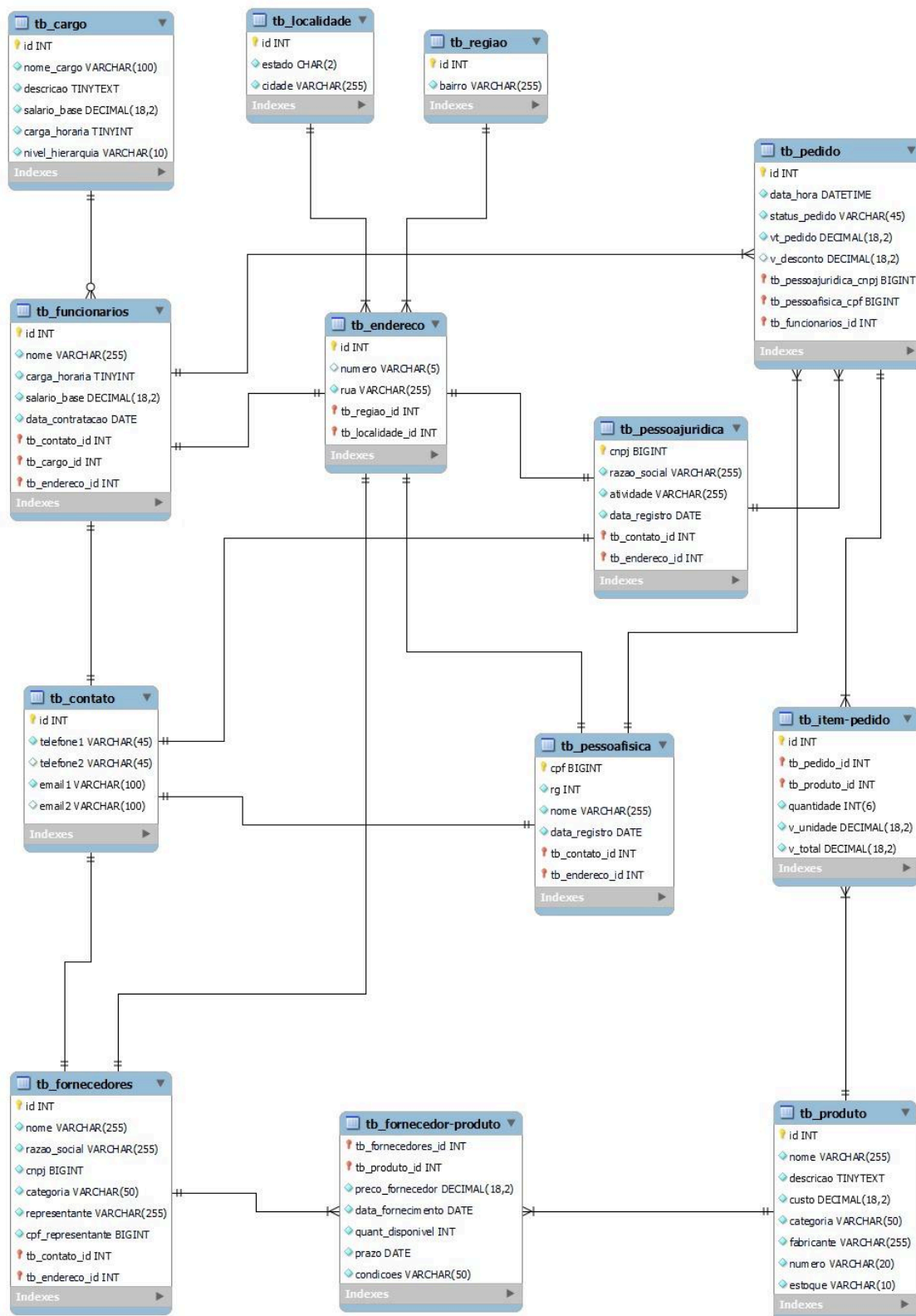


Figura 10: Modelo Lógico. Fonte: Autoria própria.

Conforme a Figura 10, Podemos observar o modelo lógico do banco de dados, obedecendo a estrutura e a devidas cardinalidades contidas no Diagrama ER.

#### 4. Modelo Físico

```
-- -----
-- Schema autopecas
-- -----
CREATE SCHEMA IF NOT EXISTS `autopecas` DEFAULT CHARACTER SET utf8 ;
USE `autopecas` ;
-- -----
-- Table `autopecas`.`tb_contato`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_contato` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `telefone1` VARCHAR(45) NOT NULL,
  `telefone2` VARCHAR(45) NULL,
  `email1` VARCHAR(100) NOT NULL,
  `email2` VARCHAR(100) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----
-- Table `autopecas`.`tb_cargo`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_cargo` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `nome_cargo` VARCHAR(100) NOT NULL,
  `descricao` TINYTEXT NOT NULL,
  `salario_base` DECIMAL(18,2) NOT NULL,
  `carga_horaria` TINYINT NOT NULL,
  `nivel_hierarquia` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----
-- Table `autopecas`.`tb_localidade`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_localidade` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `estado` CHAR(2) NOT NULL,
  `cidade` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----
-- Table `autopecas`.`tb_regiao`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_regiao` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `bairro` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
-- -----
-- Table `autopecas`.`tb_endereco`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_endereco` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `numero` VARCHAR(5) NULL,
  `rua` VARCHAR(255) NOT NULL,
  `tb_regiao_id` INT NOT NULL,
```

```

`tb_localidade_id` INT NOT NULL,
PRIMARY KEY (`id`, `tb_regiao_id`, `tb_localidade_id`),
INDEX `fk_tb_endereco_tb_regiao1_idx` (`tb_regiao_id` ASC),
INDEX `fk_tb_endereco_tb_localidade1_idx` (`tb_localidade_id` ASC),
CONSTRAINT `fk_tb_endereco_tb_regiao1`
FOREIGN KEY (`tb_regiao_id`)
REFERENCES `autopecas`.`tb_localidade` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_endereco_tb_localidade1`
FOREIGN KEY (`tb_localidade_id`)
REFERENCES `autopecas`.`tb_regiao` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_funcionarios`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_funcionarios` (
`id` INT NOT NULL AUTO_INCREMENT,
`nome` VARCHAR(255) NOT NULL,
`carga_horaria` TINYINT NOT NULL,
`salario_base` DECIMAL(18,2) NOT NULL,
`data_contratacao` DATE NOT NULL,
`tb_contato_id` INT NOT NULL,
`tb_cargo_id` INT NOT NULL,
`tb_endereco_id` INT NOT NULL,
PRIMARY KEY (`id`, `tb_contato_id`, `tb_cargo_id`, `tb_endereco_id`),
UNIQUE INDEX `id_UNIQUE` (`id` ASC),
INDEX `fk_tb_funcionarios_tb_contato1_idx` (`tb_contato_id` ASC),
INDEX `fk_tb_funcionarios_tb_cargo1_idx` (`tb_cargo_id` ASC),
INDEX `fk_tb_funcionarios_tb_endereco1_idx` (`tb_endereco_id` ASC),
CONSTRAINT `fk_tb_funcionarios_tb_contato1`
FOREIGN KEY (`tb_contato_id`)
REFERENCES `autopecas`.`tb_contato` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_funcionarios_tb_cargo1`
FOREIGN KEY (`tb_cargo_id`)
REFERENCES `autopecas`.`tb_cargo` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_funcionarios_tb_endereco1`
FOREIGN KEY (`tb_endereco_id`)
REFERENCES `autopecas`.`tb_endereco` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_produto`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_produto` (
`id` INT NOT NULL AUTO_INCREMENT,
`nome` VARCHAR(255) NOT NULL,
`descricao` TINYTEXT NOT NULL,

```

```

`custo` DECIMAL(18,2) NOT NULL,
`categoria` VARCHAR(50) NOT NULL,
`fabricante` VARCHAR(255) NOT NULL,
`numero` VARCHAR(20) NOT NULL,
`estoque` VARCHAR(10) NOT NULL,
PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_fornecedores`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_fornecedores` (
`id` INT NOT NULL AUTO_INCREMENT,
`nome` VARCHAR(255) NOT NULL,
`razao_social` VARCHAR(255) NOT NULL,
`cnpj` BIGINT NOT NULL,
`categoria` VARCHAR(50) NOT NULL,
`representante` VARCHAR(255) NOT NULL,
`cpf_representante` BIGINT NOT NULL,
`tb_contato_id` INT NOT NULL,
`tb_endereco_id` INT NOT NULL,
PRIMARY KEY (`id`, `tb_contato_id`, `tb_endereco_id`),
UNIQUE INDEX `cnpj_UNIQUE` (`cnpj` ASC),
UNIQUE INDEX `cpf_representante_UNIQUE` (`cpf_representante` ASC),
INDEX `fk_tb_fornecedores_tb_contato1_idx` (`tb_contato_id` ASC),
INDEX `fk_tb_fornecedores_tb_endereco1_idx` (`tb_endereco_id` ASC),
CONSTRAINT `fk_tb_fornecedores_tb_contato1`
FOREIGN KEY (`tb_contato_id`)
REFERENCES `autopecas`.`tb_contato` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_fornecedores_tb_endereco1`
FOREIGN KEY (`tb_endereco_id`)
REFERENCES `autopecas`.`tb_endereco` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_pessoajuridica`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_pessoajuridica` (
`cnpj` BIGINT NOT NULL,
`razao_social` VARCHAR(255) NOT NULL,
`atividade` VARCHAR(255) NOT NULL,
`data_registro` DATE NOT NULL,
`tb_contato_id` INT NOT NULL,
`tb_endereco_id` INT NOT NULL,
PRIMARY KEY (`cnpj`, `tb_contato_id`, `tb_endereco_id`),
UNIQUE INDEX `cnpj_UNIQUE` (`cnpj` ASC),
INDEX `fk_tb_pessoajuridica_tb_contato1_idx` (`tb_contato_id` ASC),
INDEX `fk_tb_pessoajuridica_tb_endereco1_idx` (`tb_endereco_id` ASC),
CONSTRAINT `fk_tb_pessoajuridica_tb_contato1`
FOREIGN KEY (`tb_contato_id`)
REFERENCES `autopecas`.`tb_contato` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,

```

```

CONSTRAINT `fk_tb_pessoajuridica_tb_endereco1`
FOREIGN KEY (`tb_endereco_id`)
REFERENCES `autopecas`.`tb_endereco` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_pessoafisica`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_pessoafisica` (
`cpf` BIGINT NOT NULL,
`rg` INT NOT NULL,
`nome` VARCHAR(255) NOT NULL,
`data_registro` DATE NOT NULL,
`tb_contato_id` INT NOT NULL,
`tb_endereco_id` INT NOT NULL,
PRIMARY KEY (`cpf`, `tb_contato_id`, `tb_endereco_id`),
UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC),
INDEX `fk_tb_pessoafisica_tb_contato1_idx` (`tb_contato_id` ASC),
INDEX `fk_tb_pessoafisica_tb_endereco1_idx` (`tb_endereco_id` ASC),
CONSTRAINT `fk_tb_pessoafisica_tb_contato1`
FOREIGN KEY (`tb_contato_id`)
REFERENCES `autopecas`.`tb_contato` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_pessoafisica_tb_endereco1`
FOREIGN KEY (`tb_endereco_id`)
REFERENCES `autopecas`.`tb_endereco` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_pedido`
-- -----
CREATE TABLE IF NOT EXISTS `autopecas`.`tb_pedido` (
`id` INT NOT NULL AUTO_INCREMENT,
`data_hora` DATETIME NOT NULL,
`status_pedido` VARCHAR(45) NOT NULL,
`vt_pedido` DECIMAL(18,2) NOT NULL,
`v_desconto` DECIMAL(18,2) NULL,
`tb_pessoajuridica_cnpj` BIGINT NOT NULL,
`tb_pessoafisica_cpf` BIGINT NOT NULL,
`tb_funcionarios_id` INT NOT NULL,
PRIMARY KEY (`id`, `tb_pessoajuridica_cnpj`, `tb_pessoafisica_cpf`,
`tb_funcionarios_id`),
INDEX `fk_tb_pedido_tb_pessoajuridica1_idx` (`tb_pessoajuridica_cnpj` ASC),
INDEX `fk_tb_pedido_tb_pessoafisica1_idx` (`tb_pessoafisica_cpf` ASC),
INDEX `fk_tb_pedido_tb_funcionarios1_idx` (`tb_funcionarios_id` ASC),
CONSTRAINT `fk_tb_pedido_tb_pessoajuridica1`
FOREIGN KEY (`tb_pessoajuridica_cnpj`)
REFERENCES `autopecas`.`tb_pessoajuridica` (`cnpj`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_pedido_tb_pessoafisica1`
FOREIGN KEY (`tb_pessoafisica_cpf`)

```

```

REFERENCES `autopecas`.`tb_pessoafisica` (`cpf`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_pedido_tb_funcionarios1`
FOREIGN KEY (`tb_funcionarios_id`)
REFERENCES `autopecas`.`tb_funcionarios` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_item-pedido`
-- -----

CREATE TABLE IF NOT EXISTS `autopecas`.`tb_item-pedido` (
`id` INT NOT NULL AUTO_INCREMENT,
`tb_pedido_id` INT NOT NULL,
`tb_produto_id` INT NOT NULL,
`quantidade` INT(6) NOT NULL,
`v_unidade` DECIMAL(18,2) NOT NULL,
`v_total` DECIMAL(18,2) NOT NULL,
PRIMARY KEY (`id`, `tb_pedido_id`, `tb_produto_id`),
INDEX `fk_tb_pedido_has_tb_produto_tb_produto1_idx` (`tb_produto_id` ASC),
INDEX `fk_tb_pedido_has_tb_produto_tb_pedido1_idx` (`tb_pedido_id` ASC),
UNIQUE INDEX `id_UNIQUE` (`id` ASC),
CONSTRAINT `fk_tb_pedido_has_tb_produto_tb_pedido1`
FOREIGN KEY (`tb_pedido_id`)
REFERENCES `autopecas`.`tb_pedido` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_pedido_has_tb_produto_tb_produto1`
FOREIGN KEY (`tb_produto_id`)
REFERENCES `autopecas`.`tb_produto` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `autopecas`.`tb_fornecedor-produto`
-- -----

CREATE TABLE IF NOT EXISTS `autopecas`.`tb_fornecedor-produto` (
`tb_fornecedores_id` INT NOT NULL,
`tb_produto_id` INT NOT NULL,
`preco_fornecedor` DECIMAL(18,2) NOT NULL,
`data_fornecimento` DATE NOT NULL,
`quant_disponivel` INT NOT NULL,
`prazo` DATE NOT NULL,
`condicoes` VARCHAR(50) NOT NULL,
PRIMARY KEY (`tb_fornecedores_id`, `tb_produto_id`),
INDEX `fk_tb_fornecedores_has_tb_produto_tb_produto1_idx` (`tb_produto_id`
ASC),
INDEX `fk_tb_fornecedores_has_tb_produto_tb_fornecedores1_idx`
(`tb_fornecedores_id`
ASC),
CONSTRAINT `fk_tb_fornecedores_has_tb_produto_tb_fornecedores1`
FOREIGN KEY (`tb_fornecedores_id`)
REFERENCES `autopecas`.`tb_fornecedores` (`id`)
ON DELETE NO ACTION

```

```

ON UPDATE NO ACTION,
CONSTRAINT `fk_tb_fornecedores_has_tb_produto_tb_produto1`
FOREIGN KEY (`tb_produto_id`)
REFERENCES `autopecas`.`tb_produto` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECK

```

## 5. Conclusão

O trabalho nos permitiu a implementação de funcionalidades importantes para a criação de um banco de dados, no futuro, esperamos continuar a desenvolver habilidades em banco de dados e a aplicar os conhecimentos adquiridos em projetos mais complexos. Abordou diversas fases do projeto, incluindo modelagem e normalização. Aprendemos a importância de criar uma estrutura eficiente para armazenar informações cruciais, garantir a segurança e otimizar as consultas, como também, compreender a importância da normalização para manter a integridade dos dados. Essa experiência nos proporcionou conhecimentos sólidos e habilidades relevantes, preparando-nos para enfrentar desafios futuros e explorar novas tecnologias nessa área.

Em suma, o trabalho foi uma oportunidade valiosa para desenvolver habilidades e conhecimentos relevantes e a experiência adquirida será aplicada em projetos futuros para garantir a eficiência e a segurança dos bancos de dados.

## 6. Referências

REPOSITÓRIO DO GITHUB DO PROFESSOR HELENO CARDOSO SILVA FILHO.