

SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL
CURSO SUPERIOR EM TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

CLODOEGE HELENA MARTINKOSKI
REINALD MENDES DOS SANTOS

SISTEMA PORTARIA

PONTA GROSSA
2023

SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL
CURSO SUPERIOR EM TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

CLODOEGE HELENA MARTINKOSKI
REINALD MENDES DOS SANTOS

SISTEMA PORTARIA

Trabalho apresentado como requisito parcial
para obtenção de aprovação na disciplina de
Banco de Dados 1, no Curso de graduação em
Tecnologia em Análise e Desenvolvimento de
Sistemas na Faculdade SENAC Paraná -
Ponta Grossa
Profª Ma. Luma Alves Lopes

PONTA GROSSA
2023

LISTA DE FIGURAS

Figura 1 – Modelo Conceitual.....	8
Figura 2 – Modelo Lógico.....	9

ANEXOS

Anexo 1 – Modelo Conceitual.....	28
Anexo 2 – Modelo Lógico.....	29

SUMÁRIO

1. INTRODUÇÃO.....	1
2. REVISÃO DA BIBLIOGRAFIA.....	1
2.1. Modelagem conceitual.....	1
2.2. Modelagem lógica.....	1-2
2.3. Álgebra relacional.....	2-3
2.4. Linguagem de Definição de dados.....	3-4
2.5. Linguagem de Manipulação de dados.....	4
3. DESENVOLVIMENTO.....	5
3.1. Descrição do cenário de negócio.....	5-7
3.2. Modelagem conceitual.....	7-8
3.3. Modelagem lógica.....	9-10
3.4. Álgebra relacional.....	10-12
3.5. Linguagem de Definição de dados.....	13-22
3.6. Linguagem de Manipulação de dados.....	22-25
4. CONSIDERAÇÕES FINAIS.....	26
REFERÊNCIAS.....	27

1. INTRODUÇÃO

Esse trabalho tem como objetivo apresentar o desenvolvimento de uma plataforma para controle e gestão de um sistema de portaria de um condomínio. Nesse sentido, a modelagem de banco de dados propõe monitorar e controlar a entrada e saída de moradores, funcionários e visitantes, assim como os veículos em um ambiente. Além disso, o sistema permitirá o registro de ocorrências de segurança, emergência e comportamentos suspeitos que podem acontecer em um condomínio residencial ou qualquer outro local que necessite de um controle rigoroso de segurança. Unindo a teoria e a prática desses conceitos sendo aplicados em situações reais podemos alcançar uma compreensão mais aprofundada das melhores práticas na modelagem de banco de dados.

2. REVISÃO DA BIBLIOGRAFIA

2.1. Modelagem conceitual

A modelagem conceitual é a descrição da informação que o sistema irá gerenciar, sendo um artefato do domínio do problema e não do domínio da solução. A modelagem conceitual apresenta o problema (o que precisa ser feito) a ser resolvido e não a solução (o como deve ser feito).

O objetivo aqui é criar um modelo de forma gráfica, sendo este chamado de Diagrama Entidade e Relacionamento (DER), o qual irá facilitar a identificação das entidades e relacionamentos de uma forma global.

Neste modelo evitamos qualquer detalhamento específico do modelo de banco de dados. Sua principal finalidade é fazer o levantamento dos requisitos e regras de negócio. No desenvolvimento de soluções é o primeiro modelo que deve ser desenvolvido.

2.2. Modelagem lógica

A modelagem lógica é necessária para compilar os requisitos de negócio e representá-los como um modelo.

Está principalmente associada à coleta de necessidades do negócio, como informações sobre unidades organizacionais, entidades de negócios e processos de negócios e não ao design do banco de dados. Descreve como os dados serão armazenados no banco e também seus relacionamentos. Esse modelo utiliza alguma tecnologia que pode ser relacional, orientado a objetos, orientado a colunas, entre outros. Na modelagem lógica podem haver chaves primárias e chaves estrangeiras.

Os modelos lógicos basicamente determinam se todos os requisitos do negócio foram reunidos. Ele é revisado pelos desenvolvedores, pelo gerenciamento e, por fim, pelos usuários finais para ver se é necessário coletar mais informações antes do início da modelagem física.

O modelo lógico também modela as informações coletadas dos requisitos do negócio.

2.3. Álgebra relacional

A Álgebra Relacional é uma linguagem de consulta formal, porém procedimental, ou seja, o usuário dá as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para calcular o resultado desejado.

Na terminologia formal de modelo relacional temos os seguintes conceitos:

- Uma linha é chamada de tupla;
- O cabeçalho da coluna é chamado de atributo;
- Tabela é chamada de relação;
- O tipo de dados que descreve os tipos de valores que podem aparecer em cada coluna é chamado de domínio;

Há seis operações fundamentais na álgebra relacional:

1. Seleção: A seleção pode ser entendida como uma operação que filtra as linhas de uma relação(tabela), e é uma operação unária, pois opera sobre um único conjunto de dados.

2. **Projeção:** Pode ser entendida como uma operação que filtra as colunas de uma tabela. Por operar sobre apenas um conjunto de entrada é classificada como uma operação unária.
3. **Produto cartesiano:** O resultado do produto cartesiano de duas relações é uma terceira relação contendo todas as combinações possíveis entre os elementos das relações originais.
4. **União:** Produz como resultado uma Relação que contém todas as linhas da primeira Relação seguidas de todas as linhas da segunda tabela. A Relação resultante possui a mesma quantidade de colunas que as relações originais, e tem um número de linhas que é no máximo igual à soma das linhas das relações fornecidas como operandos, já que as linhas que são comuns a ambas as relações aparecem uma única vez no resultado.
5. **Diferença entre conjuntos:** É uma operação que requer como operandos duas relações união-compatíveis, ou seja, estruturalmente idênticas. O resultado é uma relação que possui todas as linhas que existem na primeira relação e não existem na segunda.
6. **Renomeação:** Esta é uma operação adicional que produz como resultado uma tabela que contém, sem repetições, todos os elementos que são comuns às duas tabelas fornecidas como operandos. As tabelas devem ser união-compatíveis.

2.4. Linguagem de definição de dados (DDL)

A linguagem de definição de dados, mais conhecida como DDL é um grupo de comandos dentro da linguagem SQL que é utilizada para criação, alteração e exclusão de objetos em um banco de dados, seus principais comandos são: CREATE, ALTER e DROP.

CREATE: este comando é utilizado para a criação de estruturas de objetos do banco de dados, pode ser utilizado para N situações como criação de um novo database, tabela, índice, entre outros. Um dos modelos mais comuns do uso do comando CREATE é para a criação de tabelas.

ALTER: este comando é utilizado para alteração/modificação de estruturas de objetos do banco de dados, também pode ser utilizado para N situações de modificação de um objeto existente no seu banco de dados como adicionar ou remover uma coluna em uma tabela.

DROP: este comando é utilizado para remoção de objetos e até mesmo de um banco de dados, portanto, é a maneira em que é feito a exclusão de tabelas, usuários, databases, entre outras estruturas.

2.5. Linguagem de manipulação de dados

A linguagem de manipulação de dados, mais conhecida como DML é um grupo de comandos dentro da linguagem SQL que é utilizada para inclusão, remoção e modificar o conteúdo das tabelas em um banco de dados, seus principais comandos são: INSERT, UPDATE e DELETE.

INSERT: este é o comando utilizado para inserir registros em uma tabela já existente no seu banco de dados.

UPDATE: este é o comando utilizado para modificar as informações existentes em uma tabela, sejam dados individuais ou grupos de dados, portanto você pode atualizar um único registro ou vários dentro de uma tabela em seu banco de dados.

DELETE: este é o comando utilizado para exclusão de dados de uma tabela em seu banco de dados.

3. DESENVOLVIMENTO

3.1. Descrição do cenário de negócio: Sistema de Controle de Acesso de Portaria

Um sistema de controle de acesso de portaria visa oferecer um ambiente seguro e controlado, proporcionando tranquilidade aos moradores e colaboradores, além de facilitar a gestão da segurança do local. A modelagem de banco de dados será fundamental para garantir a integridade, eficiência e escalabilidade do sistema, possibilitando a implementação de todas as funcionalidades descritas acima. O projeto de modelagem de banco de dados proposto é para um sistema de controle de acesso de portaria, que tem como objetivo monitorar e controlar a entrada e saída de pessoas, veículos e visitantes em um ambiente, como um condomínio residencial, edifício comercial, empresa ou qualquer outro local que necessite de um controle rigoroso de segurança. As principais funcionalidades do Sistema de Controle de Acesso:

- **Cadastro de Moradores/Colaboradores:** Os moradores ou colaboradores do local serão cadastrados no sistema, fornecendo informações como nome, sobrenome, foto, número de documento, telefone, endereço e dados de veículo (se aplicável).
- **Cadastro de Visitantes:** Os visitantes que desejam entrar no local temporariamente serão cadastrados no sistema, informando nome, sobrenome, motivo da visita, nome do morador ou colaborador a ser visitado e dados de veículo (se aplicável).
- **Controle de Acesso de Veículos:** O sistema registrará os veículos dos moradores e visitantes, incluindo informações como placa, modelo, cor e proprietário. A entrada e saída dos veículos serão controladas, registrando data e hora.
- **Monitoramento de Entradas e Saídas:** Todas as entradas e saídas do local serão monitoradas e registradas pelo sistema, com carimbo de data e hora, permitindo um rastreamento detalhado do fluxo de pessoas.
- **Autorizações Especiais:** Os moradores poderão conceder autorizações especiais para visitantes, como permissões para estacionar em vagas reservadas ou acesso a áreas restritas dentro do local.
- **Controle Biométrico:** O sistema poderá ser integrado com sistemas de controle biométrico, permitindo o acesso por meio de impressões digitais, reconhecimento facial ou outra forma de identificação única.

- **Registro de Ocorrências:** O sistema permitirá o registro de ocorrências de segurança ou emergência, como acidentes, incidentes de segurança ou comportamentos suspeitos.
- **Notificações e Alertas:** Os moradores e administradores receberão notificações e alertas em tempo real sobre eventos importantes, como chegada de visitantes ou tentativas de acesso não autorizadas.
- **Gestão de Visitas Recorrentes:** Para visitantes frequentes, como prestadores de serviço ou funcionários de empresas terceirizadas, o sistema permitirá o cadastro de visitas recorrentes para agilizar o processo de entrada.
- **Relatórios de Acesso:** O sistema gerará relatórios detalhados sobre o histórico de acessos, incluindo registros de entradas e saídas, tempo de permanência, visitas mais frequentes e outros dados relevantes.
- **Controle de Portões e Catracas:** O sistema poderá ser integrado aos sistemas de controle de portões e catracas para permitir o acesso apenas aos indivíduos autorizados.
- **Integração com Câmeras de Segurança:** O sistema poderá ser integrado a câmeras de segurança, permitindo a visualização das imagens em tempo real e a gravação de eventos importantes.
- **Autorização Remota:** Os moradores poderão autorizar remotamente a entrada de visitantes, por meio de um aplicativo ou plataforma web, proporcionando maior comodidade e flexibilidade.
- **Cadastro de Funcionários da Portaria:** Os funcionários responsáveis pelo controle de acesso também serão cadastrados no sistema, incluindo informações pessoais e atribuições específicas.
- **Controle de Frequência dos Funcionários:** O sistema poderá registrar a frequência e horário de entrada e saída dos funcionários da portaria.
- **Controle de Credenciais de Acesso:** O sistema gerenciará as credenciais de acesso, como cartões de proximidade, tags de RFID ou chaves eletrônicas, garantindo a segurança física.
- **Gestão de Visitas Pré-Agendadas:** Os moradores poderão pré-agendar visitas através do sistema, agilizando o processo de entrada.
- **Integração com Sistemas de Alarme:** O sistema poderá ser integrado com sistemas de alarme para reforçar a segurança em caso de ocorrências emergenciais.

- **Monitoramento de Acesso Remoto:** Os administradores poderão monitorar o acesso em tempo real de qualquer lugar, através de dispositivos móveis ou computadores.
- **Registro de Acesso de Funcionários:** O sistema registrará a entrada e saída dos funcionários autorizados, permitindo o controle de frequência e pontualidade.
- **Controle de Acesso a Áreas Restritas:** O sistema permitirá a definição de áreas com acesso restrito, autorizando apenas indivíduos autorizados a entrar em tais locais.
- **Registro de Multas e Penalidades:** O sistema permitirá o registro de multas e penalidades para visitantes que violem as regras de acesso do local.
- **Integração com Controle Financeiro:** O sistema poderá estar integrado ao controle financeiro para gerenciar o pagamento de taxas de visitantes ou serviços adicionais.
- **Gestão de Autorizações Permanentes:** O sistema permitirá a criação de autorizações permanentes para moradores, permitindo o acesso contínuo sem a necessidade de registro constante.
- **Controle de Estacionamento:** O sistema poderá gerenciar o acesso e a disponibilidade de vagas de estacionamento para moradores, funcionários e visitantes.
- **Controle de Acesso a Áreas Restritas:** O sistema permitirá o controle de acesso a áreas restritas, como salas de reunião, data centers ou áreas sensíveis.

3.2. Modelagem conceitual

Nessa etapa da modelagem descreveremos o comportamento de apenas alguns elementos do Sistema de Portaria. Os elementos escolhidos para serem descritos são as seguintes entidades: pessoa, morador, visitante, multa, funcionário, endereço, agenda, biometria, veículo e registro. Na figura abaixo podemos visualizar essas entidades e seus respectivos atributos.

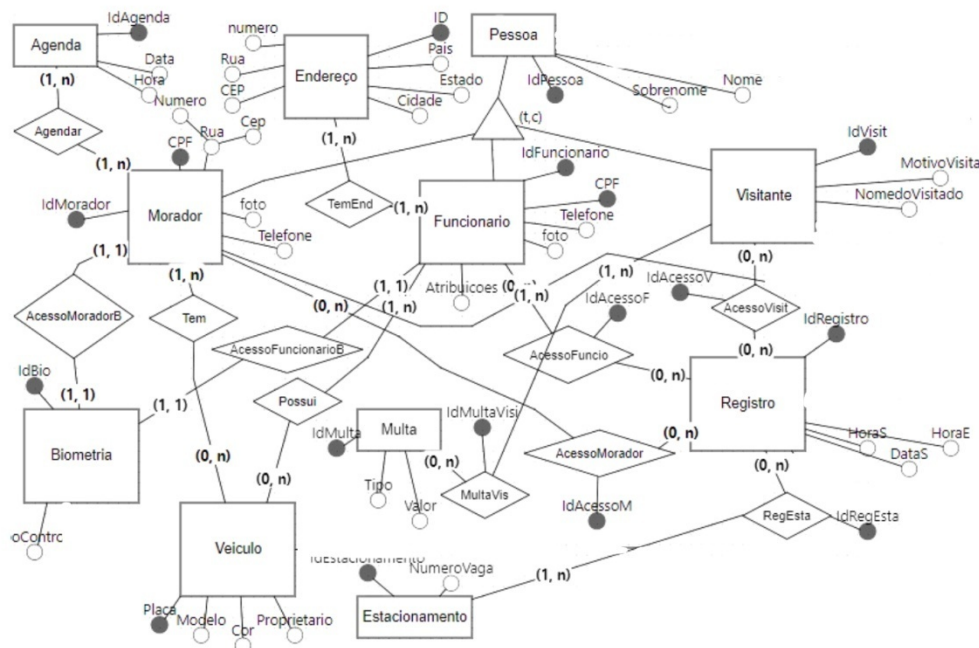


Figura 1

A entidade Pessoa tem atributos comuns que serão utilizados pelas entidades Morador, Funcionário e Visitante, sendo que cada um desses terá seus atributos próprios. O relacionamento configura uma herança.

A entidade Funcionário se relaciona com Endereço, pois os funcionários sempre terão seus endereços registrados no sistema, pois nem sempre serão moradores do condomínio.

A entidade Morador se relaciona com Agenda, Biometria, Veículo e Registro. A relação com Agenda significa que o Morador poderá pré-agendar visitas através do sistema, agilizando o processo de entrada. A relação com Biometria permitirá o acesso por meio de impressões digitais, reconhecimento facial ou outra forma de identificação única. O relacionamento de Morador com Veículo significa que cada morador terá sua vaga de estacionamento. Na relação com Registro, a entidade Morador poderá ter acesso a registros de visitantes, funcionários, vagas de estacionamento.

A entidade Visitante tem também relacionamento com Multa que irá permitir o registro de multas e penalidades para visitantes que violem as regras de acesso do local.

tabela), modelo, proprietário, cor, FK da tabela Morador, FK da tabela Funcionário e FK da tabela Visitante. A tabela Multa possui os seguintes atributos: ID de Multa, tipo, valor e a FK da tabela MultaVis. A tabela Registro tem como atributos ID de Registro, data de entrada, hora de entrada, data de saída, hora de saída, FK da tabela AcessoVisit, FK da tabela AcessoMorador, FK da tabela AcessoFuncionario e FK da tabela RegistroEstacionamento.

3.4. Álgebra Relacional

Conforme os conceitos de álgebra relacional já tratados neste trabalho, segue abaixo exemplos de scripts sobre união, interseção, junção natural, seleção e projeção. Todos os exemplos são relacionados ao Sistema de Controle de Acesso de Portaria.

União:

1. União de funcionários e veículos para saber quais funcionários têm o veículo com a cor preta

- $\sigma_{\text{Cor}=\text{Preta}}(\text{Funcionários}) \cup (\text{Veículos})$

2. União de tabelas morador e agenda para saber os agendamentos do mês de janeiro:

- $\sigma_{\text{Data}=\text{Janeiro}}(\text{Morador}) \cup (\text{Agenda})$

3. União das tabelas registro e relatório, para saber o relatório de registros do Morador com Id 10:

- $\sigma_{\text{IdAcessoM}=10}(\text{Registro}) \cup (\text{Relatorio})$

4. União das tabelas Endereço e Funcionário, para saber os residentes em Ponta Grossa:

- $\sigma_{\text{Cidade}=\text{Ponta Grossa}}(\text{Endereço}) \cup (\text{Funcionário})$

5. União tabelas filmagem e RegistroFilmagem, para saber as filmagens do mês de abril e seus respectivos horários:

- $\sigma_{\text{horário, data}=\text{Abril}}(\text{Filmagem}) \cup (\text{RegistroFilmagem})$

Interseção:

1. Interseção da tabela multa e da tabela Multavis, nas tuplas que contém o idMultVis de ambas:

- $\text{Multa} \cap \text{MultiVis}$

2. Interseção da tabela Dispara e da tabela Alarme, nas tuplas que contém o idAlarme de ambas:

- $\text{Dispara} \cap \text{Alarme}$

3. Interseção da tabela Estacionamento e da tabela RegEsta, nas tuplas que contém o idEstacionamento de ambas:

- $\text{Estacionamento} \cap \text{RegEsta}$

4. Interseção da tabela Financeiro e da tabela RegistroFinan, nas tuplas que contém o idRegistroFinan de ambas:

- $\text{Financeiro} \cap \text{RegistroFinan}$

5. Interseção da tabela Biometria e da tabela Libera, nas tuplas que contém o idBio de ambas:

- $\text{Biometria} \cap \text{Libera}$

Seleção :

1. Seleção de todos os endereços do estado do Paraná :

- $\sigma \text{ Estado} = \text{Paraná} (\text{Endereço})$

2. Seleção do número das vagas maior que 20:

- $\sigma \text{ NumeroVaga} > 20 (\text{Estacionamento})$

3. Seleção do horário da filmagem igual às 08:00 horas :

- $\sigma \text{ horario} = 08:00 (\text{Filmagem})$

4. Seleção de todos os moradores que estão localizados na Rua XV:

- $\sigma \text{ rua} = ' \text{Rua XV} ' (\text{Moradores})$

5. Seleção das datas das ocorrências no mês de agosto:

- $\sigma \text{ Data} = \text{Agosto} (\text{Ocorrência})$

Projeção:

1. Projeção dos nomes e telefones dos funcionários:

- π Nome, telefone (Funcionário)

2. Projeção das atribuições e fotos dos nossos funcionários:

- π Atribuições, Foto (Funcionário)

3. Projeção das horas e datas da agenda:

- π Hora, Data (Agenda)

4. Projeção dos modelos e proprietários dos veículos:

- π Modelo, Proprietário (Veículo)

5. Projeção do motivo da visita e nome do visitante:

- π MotivoVisita, NomeDoVisitado (Visitante)

Junção Natural:

1. Junção natural de moradores e veículos para obter moradores com seus respectivos veículos:

- Morador \bowtie Veículo

2. Junção natural de funcionário e biometria para obter informações sobre os tipos de controle que os funcionários utilizam:

- Funcionário \bowtie Biometria

3. Junção natural de visitante e registro financeiro para saber valor foi pago pelo visitante:

- Visitante \bowtie Registro Financeiro

4. Junção natural de visitante e multavis para saber de quem a multa pertence:

- Visitante \bowtie MultaVis

5. Junção natural de endereço e funcionário para identificar os funcionários de cada rua:

- Endereço \bowtie Funcionário

3.5. Linguagem de Definição de Dados

Conforme os conceitos de definição de dados, segue abaixo o script completo das tabelas criadas no banco de dados relacional Postgree. O script contém em suas tabelas restrições de integridade para proporcionar maior segurança de usabilidade.

```
CREATE TABLE Pessoa (  
  
    IdPessoa INT PRIMARY KEY,  
  
    Nome VARCHAR NOT NULL,  
  
    Sobrenome VARCHAR NOT NULL  
  
);  
  
CREATE TABLE Morador (  
  
    IdMorador INT PRIMARY KEY,  
  
    CPF VARCHAR,  
  
    Telefone INT,  
  
    foto INT,  
  
    Numero INT,  
  
    Cep INT,  
  
    IdPessoa INT,  
  
    FOREIGN KEY (IdPessoa) REFERENCES Pessoa (IdPessoa) ON DELETE  
    CASCADE ON UPDATE SET NULL  
  
);  
  
CREATE TABLE Funcionario (  
  
    IdFuncionario INT PRIMARY KEY,  
  
    CPF VARCHAR,  
  
    Telefone INT,  
  
    foto INT,  
  
    Atribuicoes INT,
```

IdPessoa INT,

FOREIGN KEY (IdPessoa) REFERENCES Pessoa (IdPessoa) ON DELETE SET
NULL ON UPDATE CASCADE

);

CREATE TABLE Visitante (

IdVisit INT PRIMARY KEY,

MotivoVisita VARCHAR,

NomedoVisitado VARCHAR,

IdPessoa INT,

FOREIGN KEY (IdPessoa) REFERENCES Pessoa (IdPessoa) ON DELETE SET
NULL ON UPDATE SET NULL

);

CREATE TABLE Veiculo (

Placa INT PRIMARY KEY,

Modelo VARCHAR,

Proprietario VARCHAR,

Cor INT,

IdMorador INT,

IdFuncionario INT,

IdVisit INT,

FOREIGN KEY (IdMorador) REFERENCES Morador (IdMorador),

FOREIGN KEY (IdFuncionario) REFERENCES Funcionario (IdFuncionario),

FOREIGN KEY (IdVisit) REFERENCES Visitante (IdVisit)

);

CREATE TABLE Endereço (

```
numero INT,

    Rua VARCHAR,

    CEP VARCHAR,

    ID INT PRIMARY KEY,

    Pais VARCHAR,

    Estado VARCHAR,

    Cidade VARCHAR,

    IdFuncionario INT,

        FOREIGN KEY (IdFuncionario) REFERENCES Funcionario (IdFuncionario)

);

CREATE TABLE AcessoVisit (

    IdAcessoVisit INT PRIMARY KEY,

    IdVisit INT,

        FOREIGN KEY (IdVisit) REFERENCES Visitante (IdVisit)

);

CREATE TABLE AcessoFuncio (

    IdAcessoFuncio INT PRIMARY KEY,

    IdFuncionario INT,

        FOREIGN KEY (IdFuncionario) REFERENCES Funcionario (IdFuncionario)

);

CREATE TABLE AcessoMorador (

    IdAcessoM INT PRIMARY KEY,

    IdMorador INT,

        FOREIGN KEY (IdMorador) REFERENCES Morador (IdMorador)

);
```

```
CREATE TABLE AreaRestrita (  
  
    IdAreaR INT PRIMARY KEY,  
  
    NomeLocal VARCHAR,  
  
    IdPessoa INT,  
  
        FOREIGN KEY (IdPessoa) REFERENCES Pessoa (IdPessoa)  
  
    );  
  
CREATE TABLE Biometria (  
  
    IdBio INT PRIMARY KEY,  
  
    tipoControle VARCHAR,  
  
    IdMorador INT,  
  
    IdFuncionario INT,  
  
        FOREIGN KEY (IdMorador) REFERENCES Morador (IdMorador),  
  
        FOREIGN KEY (IdFuncionario) REFERENCES Funcionario (IdFuncionario)  
  
    );  
  
CREATE TABLE Filmagem (  
  
    idFilmagem SERIAL PRIMARY KEY,  
  
    horario TIME,  
  
    dataf DATE  
  
    );  
  
CREATE TABLE Portao (  
  
    IdPortao INT PRIMARY KEY,  
  
        tipo VARCHAR,  
  
    IdBio INT,  
  
    IdPessoa INT,  
  
    DataAcesso DATE,
```

```
FOREIGN KEY (IdBio) REFERENCES Biometria (IdBio),

FOREIGN KEY (IdPessoa) REFERENCES Pessoa (IdPessoa)

);

CREATE TABLE Agenda (

    Hora TIME,

    DataAgenda DATE,

    IdAgenda SERIAL PRIMARY KEY

);

CREATE TABLE Alarme (

    IdAlarme INT PRIMARY KEY

);

CREATE TABLE Admin (

    IdAdmin INT PRIMARY KEY,

    Nome VARCHAR,

    permissao VARCHAR

);

CREATE TABLE Estacionamento (

    IdEstacionamento INT PRIMARY KEY,

    NumeroVaga INT

);

CREATE TABLE AdminFilme (

    idAdminFilm INT PRIMARY KEY,

    idFilmagem INT,

    IdAdmin INT,

    FOREIGN KEY (idFilmagem) REFERENCES Filmagem (idFilmagem),
```

```
        FOREIGN KEY (IdAdmin) REFERENCES Admin (IdAdmin)

);

CREATE TABLE RegistroFinan (

IdRegistroFinan INT PRIMARY KEY,

IdVisit INT,

        Valor FLOAT,

        FOREIGN KEY (IdVisit) REFERENCES Visitante (IdVisit)

);

CREATE TABLE Financeiro (

IdFinanceiro INT PRIMARY KEY,

idRegistrofinan INT,

        FOREIGN      KEY      (idRegistrofinan)      REFERENCES      RegistroFinan
(IdRegistroFinan)

);

CREATE TABLE RegistroFilma (

idRegistroFilma INT PRIMARY KEY,

idFilmagem INT,

        FOREIGN KEY (idFilmagem) REFERENCES Filmagem (idFilmagem)

);

CREATE TABLE Dispara (

IdDispara INT PRIMARY KEY,

IdAlarme INT,

        FOREIGN KEY (IdAlarme) REFERENCES Alarme (IdAlarme)

);

CREATE TABLE Ocorrencia (
```

```

idOcorrencia SERIAL PRIMARY KEY,

    Tipo VARCHAR(255),

dataO TIMESTAMP,

    Hora TIME,

idRegistroFilma INT,

idDispara INT,

    FOREIGN KEY (idRegistroFilma) REFERENCES RegistroFilma
(idRegistroFilma),

    FOREIGN KEY (idDispara) REFERENCES Dispara (IdDispara)

);

CREATE TABLE NotificaOco (

IdNotificaOco INT PRIMARY KEY,

idOcorrencia INT,

    FOREIGN KEY (idOcorrencia) REFERENCES Ocorrencia (idOcorrencia)

);

CREATE TABLE Notificacao (

IdNotificacao INT PRIMARY KEY,

idNotificaOco INT,

    FOREIGN KEY (idNotificaOco) REFERENCES NotificaOco (IdNotificaOco)

);

CREATE TABLE NotificaMorador (

IdNotificaM INT PRIMARY KEY,

IdMorador INT,

idNotificacao INT,

    FOREIGN KEY (IdMorador) REFERENCES Morador (IdMorador),

```



```
        FOREIGN KEY (idNotificacao) REFERENCES Notificacao (IdNotificacao)

);

CREATE TABLE NotificaADM (

IdNotAdm INT PRIMARY KEY,

IdAdmin INT,

idNotificacao INT,

        FOREIGN KEY (IdAdmin) REFERENCES Admin (IdAdmin),

        FOREIGN KEY (idNotificacao) REFERENCES Notificacao (IdNotificacao)

);

CREATE TABLE RegistraOcorrencia (

idRegiOco INT PRIMARY KEY,

idPessoa INT,

idOcorrencia INT,

        FOREIGN KEY (idPessoa) REFERENCES Pessoa (IdPessoa),

        FOREIGN KEY (idOcorrencia) REFERENCES Ocorrencia (idOcorrencia)

);

CREATE TABLE Agendar (

IdAgendar INT PRIMARY KEY,

IdAgenda INT,

idMorador INT,

        FOREIGN KEY (IdAgenda) REFERENCES Agenda (IdAgenda),

        FOREIGN KEY (idMorador) REFERENCES Morador (IdMorador)

);
```

```
CREATE TABLE MultaVis (  
  
  IdMultaVis INT PRIMARY KEY,  
  
  IdVisit INT,  
  
    FOREIGN KEY (IdVisit) REFERENCES Visitante (IdVisit)  
  
);  
  
CREATE TABLE Multa (  
  
  IdMulta INT PRIMARY KEY,  
  
    Tipo VARCHAR,  
  
    Valor INT,  
  
  idMultaVis INT,  
  
    FOREIGN KEY (idMultaVis) REFERENCES MultaVis (IdMultaVis)  
  
);  
  
CREATE TABLE RegEsta (  
  
  IdRegEsta INT PRIMARY KEY,  
  
  IdEstacionamento INT,  
  
    FOREIGN KEY (IdEstacionamento) REFERENCES Estacionamento  
  (IdEstacionamento)  
  
);  
  
CREATE TABLE Registro (  
  
  IdRegistro SERIAL PRIMARY KEY,  
  
  DataE DATE,  
  
  HoraE TIME,  
  
  DataS DATE,  
  
  HoraS TIME,  
  
  idAcessoVisit INT,
```

```

idAcessoFuncio INT,

idAcessoM INT,

idRegEsta INT,

    FOREIGN KEY (idAcessoVisit) REFERENCES AcessoVisit (idAcessoVisit),

    FOREIGN KEY(idAcessoFuncio) REFERENCES AcessoFuncio(idAcessoFuncio),

    FOREIGN KEY (idAcessoM) REFERENCES AcessoMorador (idAcessoM),

    FOREIGN KEY (idRegEsta) REFERENCES RegEsta (IdRegEsta)

);

CREATE TABLE Relatorio (

idRelatorio INT PRIMARY KEY,

idRegistro INT,

    FOREIGN KEY (idRegistro) REFERENCES Registro (idRegistro)

);

ALTER TABLE Portao

ADD COLUMN IdPessoa INT,

ADD FOREIGN KEY (IdPessoa) REFERENCES Pessoa (IdPessoa);

ALTER TABLE Portao

ADD COLUMN DataAcesso DATE;

```

3.6. Linguagem de Manipulação de Dados

Conforme os conceitos de manipulação de dados, segue abaixo exemplos de scripts para inserção, seleção e amostra de dados, relacionados com as tabelas já criadas no banco de dados relacional Postgree. O script contém um enunciado que explica o que é desejado de retorno ao final da consulta.

Inserção de 10 Cadastros de Moradores Fictícios:

```
INSERT INTO Pessoa (IdPessoa, Nome, Sobrenome)
VALUES
(1, 'João', 'Silva'),
(2, 'Maria', 'Oliveira'),
(3, 'Carlos', 'Santos'),
(4, 'Ana', 'Pereira'),
(5, 'Luiz', 'Mendes'),
(6, 'Paula', 'Lima'),
(7, 'Pedro', 'Alves'),
(8, 'Camila', 'Costa'),
(9, 'Rafaela', 'Rocha'),
(10, 'Daniel', 'Oliveira');
```

Inserção de 10 visitas programadas:

```
INSERT INTO Visitante (IdVisit, MotivoVisita, NomedoVisitado, IdPessoa)
VALUES
(1, 'Entrega de encomenda', 'João Silva', 1),
(2, 'Reunião de negócios', 'Maria Oliveira', 2),
(3, 'Manutenção predial', 'Carlos Santos', 3),
(4, 'Visita familiar', 'Ana Pereira', 4),
(5, 'Entrega de documentos', 'Luiz Mendes', 5),
(6, 'Instalação de serviços', 'Paula Lima', 6),
(7, 'Entrega de convite', 'Pedro Alves', 7),
(8, 'Visita de amigos', 'Camila Costa', 8),
(9, 'Reparo elétrico', 'Rafaela Rocha', 9),
(10, 'Consulta médica', 'Daniel Oliveira', 10);
```

Consulta SQL - informações sobre as pessoas que acessaram a portaria nos últimos 6 meses:

Temos que inserir dados fictícios na tabela Biometria:

```
INSERT INTO Biometria (IdBio, tipoControle, idMorador, idFuncionario)
VALUES

(1, 1, NULL, NULL) ,

(2, 2, NULL, NULL) ,

(3, 1, NULL, NULL) ,

(4, 2, NULL, NULL) ,

(5, 1, NULL, NULL) ;
```

Temos que inserir dados fictícios na tabela Portão com base nos dados da tabela Pessoa:

```
INSERT INTO Portao (IdPortao, tipo, idBio, idPessoa, DataAcesso) VALUES

(6, 1, 1, 1, '2023-11-17') ,

(7, 2, 2, 2, '2023-11-16') ,

(8, 1, 3, 3, '2023-11-15') ,

(9, 2, 4, 4, '2023-11-14') ,

(10, 1, 5, 5, '2023-11-13') ;
```

Consulta SQL para as pessoas que acessaram a portaria nos últimos 6 meses:

```
SELECT

IdPessoa

FROM

portao

WHERE

dataAcesso >= CURRENT_DATE - INTERVAL '6 month' ;
```

Consulta SQL - as informações sobre os incidentes da última semana:

Inserir dados fictícios na tabela Ocorrência:

```
INSERT INTO Ocorrencia (idOcorrencia, Tipo, dataO, Hora,  
idRegistroFilma, idDispara)
```

```
VALUES
```

```
(1, 'Intrusão', '2023-11-17', '12', NULL, NULL),  
(2, 'Roubo', '2023-11-18', '15', NULL, NULL),  
(3, 'Incêndio', '2023-11-19', '08', NULL, NULL),  
(4, 'Vandalismo', '2023-11-20', '18', NULL, NULL),  
(5, 'Alarme Falso', '2023-11-21', '10', NULL, NULL);
```

Consulta dos incidentes da última semana:

```
SELECT *  
  
FROM Ocorrencia  
  
WHERE dataO >= CURRENT_DATE - INTERVAL '1 week';
```

Consulta SQL - Número de Vagas Disponíveis:

```
SELECT COUNT(*) AS VagasDisponiveis  
  
FROM Estacionamento;
```

4. CONSIDERAÇÕES FINAIS

Desenvolver o sistema de portaria proporcionou uma melhor compreensão da modelagem de banco de dados. A transição fluida da modelagem conceitual para a lógica, aliada à aplicação precisa da Linguagem de Definição de Dados (DDL), revelou-se crucial para estabelecer uma estrutura sólida. A manipulação de dados por meio da Linguagem de Manipulação de Dados (DML) não apenas facilitou a interação, mas também ressaltou a importância de consultas eficientes. A implementação do controle de acesso e agendamento revelou-se um ponto forte, proporcionando uma gestão flexível e adaptável a diferentes cenários. A integração do sistema com o registro de ocorrências e as câmeras de monitoramento ampliou significativamente a abordagem de segurança, permitindo análises mais abrangentes. O processo enfatizou a escolha de critérios de tipos de dados e a aplicação de restrições, essenciais para a integridade dos dados. Para concluir, o projeto não apenas solidificou os conhecimentos teóricos sobre banco de dados, mas também destacou a aplicabilidade prática de conceitos desses cenários do mundo real.

REFERÊNCIAS

Fonseca, Leonardo. **Linguagem de Definição de Dados (DDL)**. Disponível em: MEDIUM. Fast Casual: o que é. Disponível em: <https://medium.com/@All182/fast-casual-o-que-%C3%A9-9e5c205d47ca> . Acesso em: 27 Jun. 2023. Acesso em: 20 de novembro de 2023.

Fonseca, Leonardo. **Linguagem de Manipulação de Dados (DML)**. Disponível em: <https://leonardofonseca.com.br/2021/05/10/linguagem-de-manipulacao-de-dados-dml/>. Acesso em: 22 de novembro de 2023.

Macoratti.net. **SQL - Arquitetura de banco de dados..** Disponível em: https://www.macoratti.net/13/06/sql_arcb.htm. Acesso em: 15 de novembro de 2023.

Stack Overflow. **Qual a diferença entre modelagem conceitual, lógica e física?**. Disponível em: COSTA, Edwaldo et al. Esg (environmental, social and corporate governance) e a Comunicação: O tripé da sustentabilidade aplicado às organizações globalizadas, São Paulo, junho 2021. Disponível em: <https://www.revistas.usp.br/alterjor/article/view/187464/174551>. Acesso em 26 jun. 2023.. Acesso em: 11 de novembro de 2023.

ANEXO I

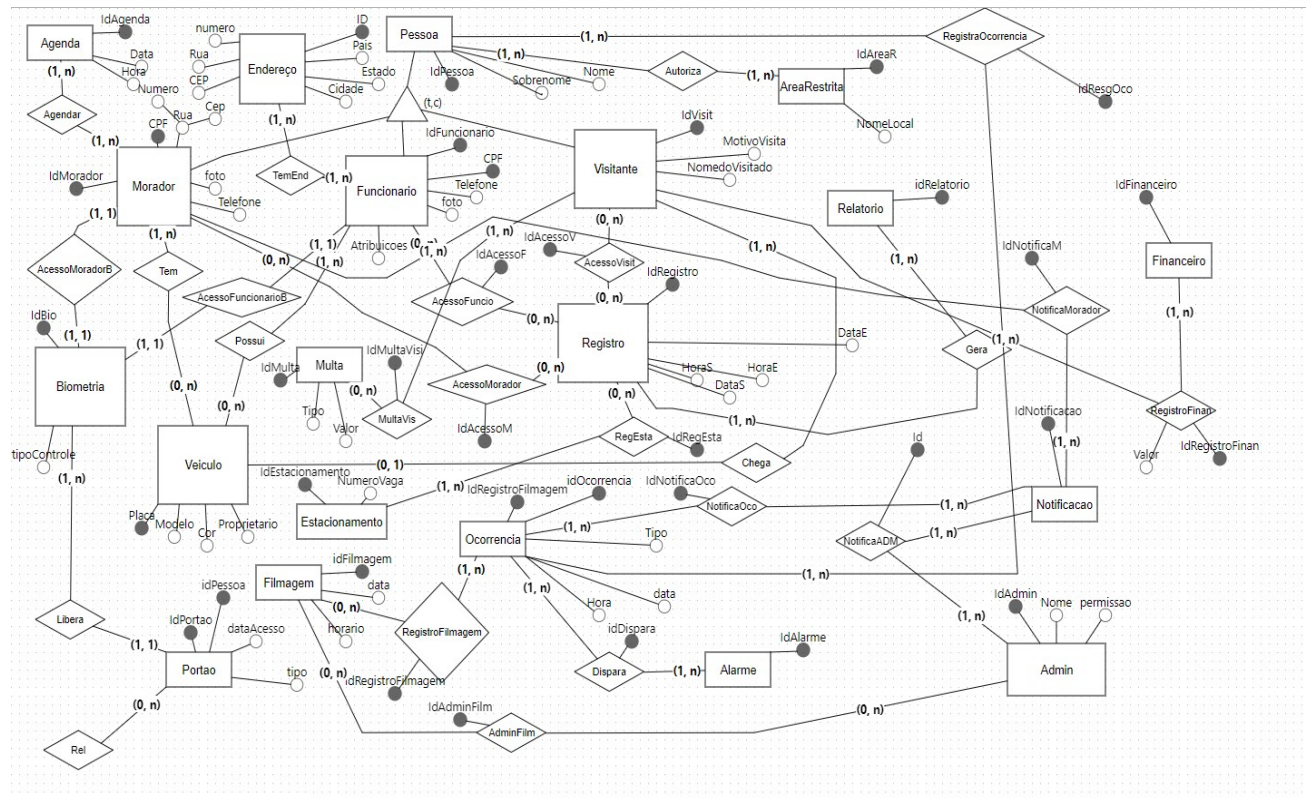


Diagrama Conceitual

ANEXO II

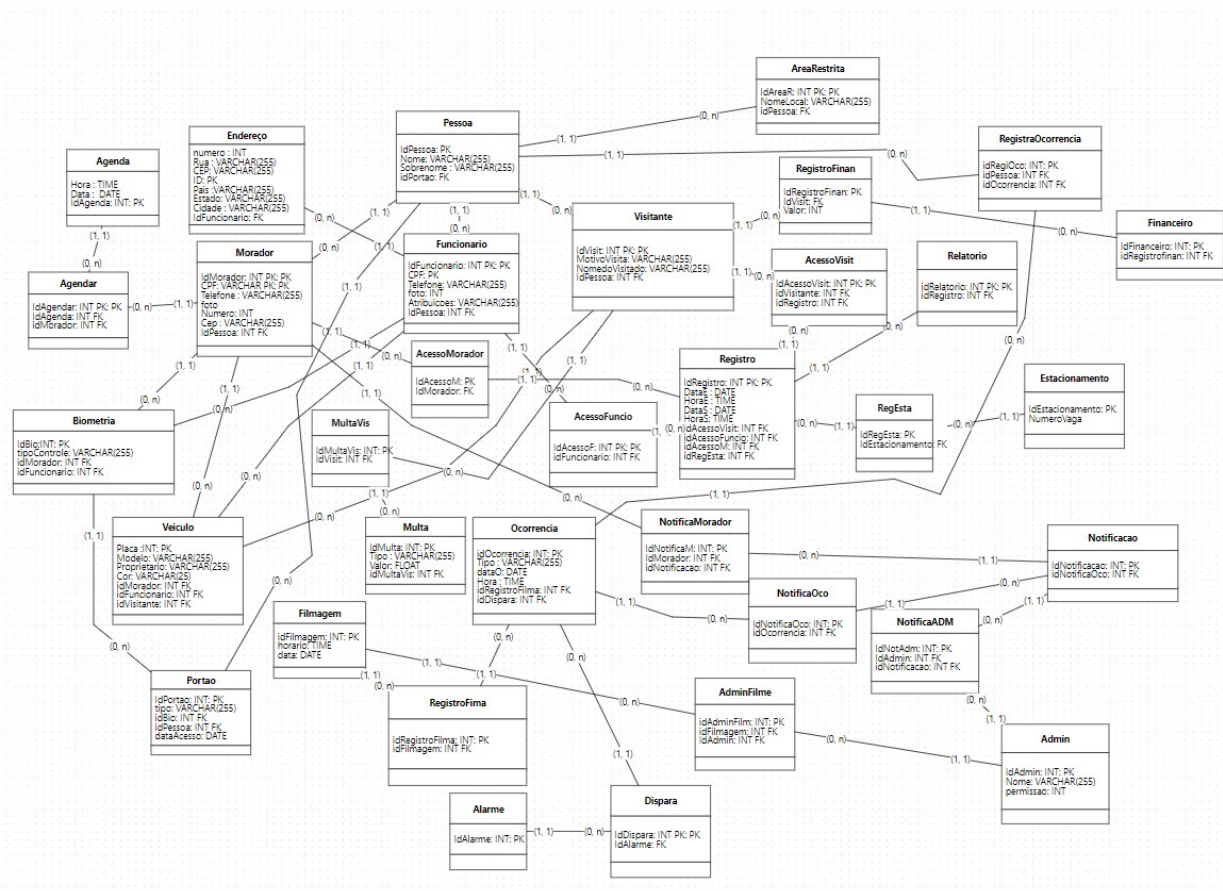


Diagrama Lógico