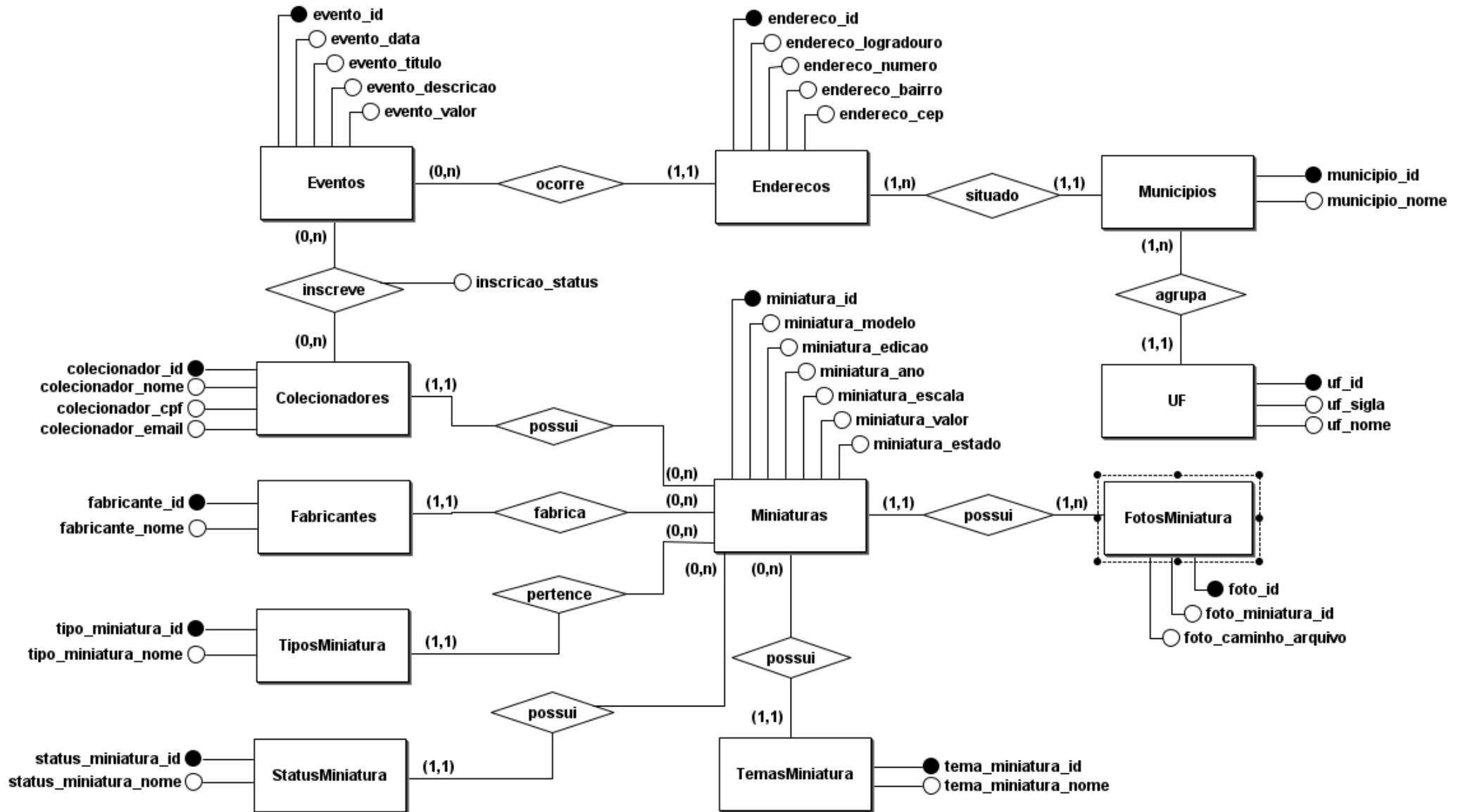
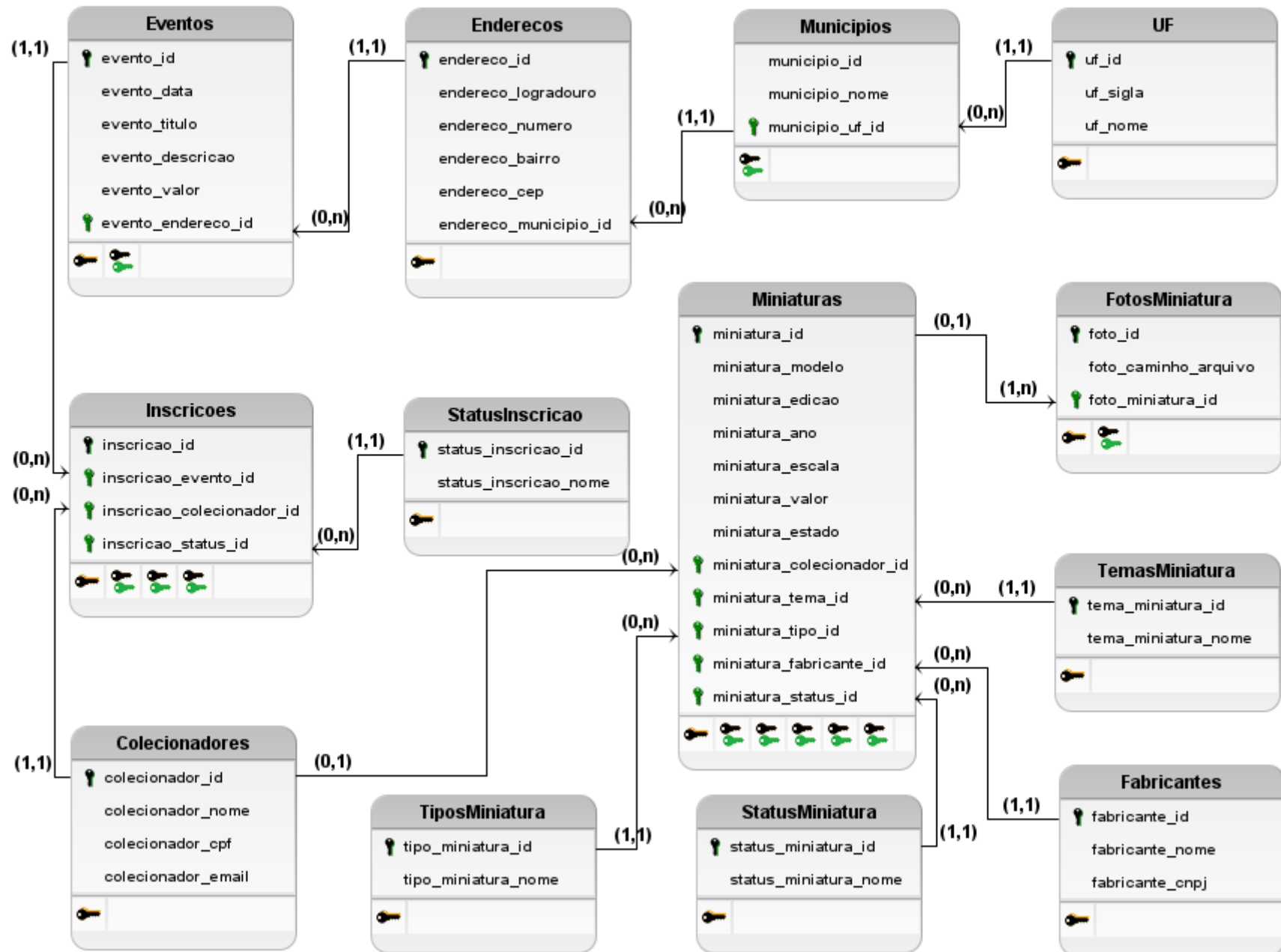


```
-- ATIVIDADE MINIATURA
--
-- ALUNOS: Lucas Araujo
--          Marcos Job Dias Lima
--          Marcos Paulo Paixao
--          Vinicius Araujo Lopes
--
-- Um grupo de colecionadores de miniaturas de veículos (carros, motos, aviões, etc) desejam cadastrar suas coleções.
--
-- Cada mini terá um modelo, edição, ano, fabricante, escala, tema (sem tema, Star Wars, Simpsons, Batman, etc ),
-- status (coleção própria, venda, procuro ou troca), em caso de venda, informar o valor e por fim as observações do estado do mini.
--
-- Todos os minis registrados deverão conter no mínimo uma foto para demonstrar o real estado do mesmo.Porém, quanto mais fotos melhor.
--
-- Os membros do grupo podem interagir com outros membros informando que existe um interesse em um determinado mini (venda) ou que possua
-- um mini a qual o outro membro esteja a procura/troca.
--
-- Além disso, ao longo do ano existem vários eventos do grupo a fim de estreitar os laços de amizade entre os membros.
--
-- O registro dos eventos devem conter data, título, descrição, endereço e valor.
-- Para cada evento é registrado se tal membro possui a inscrição paga ou não.
--
--
```

# Conceitual



## Lógico



## Físico

-- DROPS para refazer a estrutura do banco.

--

```
DROP TABLE "FotosMiniatura";
DROP TABLE "Miniaturas";
DROP TABLE "TemasMiniatura";
DROP TABLE "Fabricantes";
DROP TABLE "StatusMiniatura";
DROP TABLE "TiposMiniatura";
DROP TABLE "Inscricoes";
DROP TABLE "StatusInscricao";
DROP TABLE "Colectionadores";
DROP TABLE "Eventos";
DROP TABLE "Enderecos";
DROP TABLE "Municipios";
DROP TABLE "UF";
```

-- Unidades da federação.

-- O registro dos eventos devem conter data, título, descrição, endereço e valor.

CREATE TABLE "UF"

```
(
    uf_id SERIAL NOT NULL,
    uf_sigla CHAR(2) UNIQUE NOT NULL,
    uf_nome VARCHAR(19) UNIQUE NOT NULL,
    PRIMARY KEY (uf_id)
);
```

-- Municípios.

-- O registro dos eventos devem conter data, título, descrição, endereço e valor.

CREATE TABLE "Municipios"

```
(
    municipio_id SERIAL NOT NULL,
    municipio_nome VARCHAR(50) NOT NULL,
    municipio_uf_id INTEGER NOT NULL,
    PRIMARY KEY (municipio_id),
    FOREIGN KEY (municipio_uf_id) REFERENCES "UF" (uf_id)
);
```

```
-- Endereços de eventos.  
-- 0 registro dos eventos devem conter data, título, descrição, endereço e valor.
```

```
CREATE TABLE "Enderecos"  
(  
    endereco_id SERIAL NOT NULL,  
    endereco_logradouro VARCHAR(255) NOT NULL,  
    endereco_numero INTEGER,  
    endereco_bairro VARCHAR(50) NOT NULL,  
    endereco_cep char(8) NOT NULL,  
    endereco_municipio_id INTEGER NOT NULL,  
    PRIMARY KEY (endereco_id),  
    FOREIGN KEY (endereco_municipio_id) REFERENCES "Municipios" (municipio_id)  
);
```

```
-- Eventos de colecionadores.  
-- Além disso, ao longo do ano existem vários eventos do grupo a fim de estreitar os laços de amizade entre os membros.  
-- 0 registro dos eventos devem conter data, título, descrição, endereço e valor.
```

```
CREATE TABLE "Eventos"  
(  
    evento_id SERIAL NOT NULL,  
    evento_data TIMESTAMP NOT NULL,  
    evento_titulo VARCHAR(50) NOT NULL,  
    evento_descricao VARCHAR(255) NOT NULL,  
    evento_valor REAL NOT NULL,  
    evento_endereco_id INTEGER NOT NULL,  
    PRIMARY KEY (evento_id),  
    FOREIGN KEY (evento_endereco_id) REFERENCES "Enderecos" (endereco_id)  
);
```

```
-- Colecionadores de miniaturas.  
-- Um grupo de colecionadores de miniaturas de veículos (carros, motos, aviões, etc) desejam cadastrar suas coleções.
```

```
CREATE TABLE "Colecionadores"  
(  
    colecionador_id SERIAL NOT NULL,  
    colecionador_nome VARCHAR(50) NOT NULL,  
    colecionador_cpf CHAR(11) UNIQUE NOT NULL,  
    colecionador_email VARCHAR(50) UNIQUE NOT NULL,  
    PRIMARY KEY (coleccionador_id)  
);
```

```
-- Status de inscrição em eventos.  
-- Para cada evento é registrado se tal membro possui a inscrição paga ou não.
```

```
CREATE TABLE "StatusInscricao"  
(  
    status_inscricao_id SERIAL NOT NULL,  
    status_inscricao_nome VARCHAR(50) UNIQUE NOT NULL,  
    PRIMARY KEY (status_inscricao_id)  
);
```

```
-- Inscrições em eventos.  
-- Para cada evento é registrado se tal membro possui a inscrição paga ou não.
```

```
CREATE TABLE "Inscricoes"  
(  
    inscricao_id SERIAL NOT NULL,  
    inscricao_evento_id INTEGER NOT NULL,  
    inscricao_coleccionador_id INTEGER NOT NULL,  
    inscricao_status_id INTEGER NOT NULL,  
    PRIMARY KEY (inscricao_id),  
    FOREIGN KEY (inscricao_evento_id) REFERENCES "Eventos" (evento_id),  
    FOREIGN KEY (inscricao_coleccionador_id) REFERENCES "Coleccionadores" (coleccionador_id),  
    FOREIGN KEY (inscricao_status_id) REFERENCES "StatusInscricao" (status_inscricao_id)  
);
```

```
-- Miniaturas.  
-- Um grupo de colecionadores de miniaturas de veículos (carros, motos, aviões, etc) desejam cadastrar suas coleções.
```

```
CREATE TABLE "TiposMiniatura"  
(  
    tipo_miniatura_id SERIAL NOT NULL,  
    tipo_miniatura_nome VARCHAR(50) UNIQUE NOT NULL,  
    PRIMARY KEY (tipo_miniatura_id)  
);
```

```
-- Status de miniaturas.  
-- Cada mini terá um modelo, edição, ano, fabricante, escala, tema (sem tema, Star Wars, Simpsons, Batman, etc ),  
-- status (coleção própria, venda, procuro ou troca), em caso de venda, informar o valor e por fim as observações do estado do mini.
```

```
CREATE TABLE "StatusMiniatura"  
(  
    status_miniatura_id SERIAL NOT NULL,  
    status_miniatura_nome VARCHAR(50) UNIQUE NOT NULL,  
    PRIMARY KEY (status_miniatura_id)  
);
```

```
-- Fabricantes de miniaturas.  
-- Cada mini terá um modelo, edição, ano, fabricante, escala, tema (sem tema, Star Wars, Simpsons, Batman, etc ),  
-- status (coleção própria, venda, procuro ou troca), em caso de venda, informar o valor e por fim as observações do estado do mini.
```

```
CREATE TABLE "Fabricantes"  
(  
    fabricante_id SERIAL NOT NULL,  
    fabricante_nome VARCHAR(50) NOT NULL,  
    fabricante_cnpj CHAR(14) UNIQUE NOT NULL,  
    PRIMARY KEY (fabricante_id)  
);
```

```

-- Temas de miniaturas.
-- Cada mini terá um modelo, edição, ano, fabricante, escala, tema (sem tema, Star Wars, Simpsons, Batman, etc ),
-- status (coleção própria, venda, procuro ou troca), em caso de venda, informar o valor e por fim as observações do estado do mini.

CREATE TABLE "TemasMiniatura"
(
    tema_miniatura_id SERIAL NOT NULL,
    tema_miniatura_nome VARCHAR(50) UNIQUE NOT NULL,
    PRIMARY KEY (tema_miniatura_id)
);

-- Miniaturas.
-- Um grupo de colecionadores de miniaturas de veículos (carros, motos, aviões, etc) desejam cadastrar suas coleções.
-- Cada mini terá um modelo, edição, ano, fabricante, escala, tema (sem tema, Star Wars, Simpsons, Batman, etc ),
-- status (coleção própria, venda, procuro ou troca), em caso de venda, informar o valor e por fim as observações do estado do mini.
-- Todos os minis registrados deverão conter no mínimo uma foto para demonstrar o real estado do mesmo. Porém, quanto mais fotos melhor.
-- Os membros do grupo podem interagir com outros membros informando que existe um interesse em um determinado mini (venda) ou que possua
-- um mini a qual o outro membro esteja a procura/troca.

CREATE TABLE "Miniaturas"
(
    miniatura_id SERIAL NOT NULL,
    miniatura_modelo VARCHAR(50) NOT NULL,
    miniatura_edicao VARCHAR(50) NOT NULL,
    miniatura_ano INTEGER NOT NULL,
    miniatura_escala VARCHAR(50) NOT NULL,
    miniatura_VALOR REAL NOT NULL,
    miniatura_estado VARCHAR(255),
    miniatura_colecionador_id INTEGER NOT NULL,
    miniatura_tema_id INTEGER NOT NULL,
    miniatura_tipo_id INTEGER NOT NULL,
    miniatura_fabricante_id INTEGER NOT NULL,
    miniatura_status_id INTEGER NOT NULL,
    PRIMARY KEY (miniatura_id),
    FOREIGN KEY (miniatura_colecionador_id) REFERENCES "Colecionadores" (colecionador_id),
    FOREIGN KEY (miniatura_tema_id) REFERENCES "TemasMiniatura" (tema_miniatura_id),
    FOREIGN KEY (miniatura_tipo_id) REFERENCES "TiposMiniatura" (tipo_miniatura_id),
    FOREIGN KEY (miniatura_fabricante_id) REFERENCES "Fabricantes" (fabricante_id),
    FOREIGN KEY (miniatura_status_id) REFERENCES "StatusMiniatura" (status_miniatura_id)
);

-- Fotos de miniaturas.
-- Todos os minis registrados deverão conter no mínimo uma foto para demonstrar o real estado do mesmo. Porém, quanto mais fotos melhor.

CREATE TABLE "FotosMiniatura"
(
    foto_id SERIAL NOT NULL,
    foto_caminho_arquivo VARCHAR(255) NOT NULL,
    foto_miniatura_id INTEGER NOT NULL,
    PRIMARY KEY (foto_id),
    FOREIGN KEY (foto_miniatura_id) REFERENCES "Miniaturas" (miniatura_id)
);

-- Fim.

```