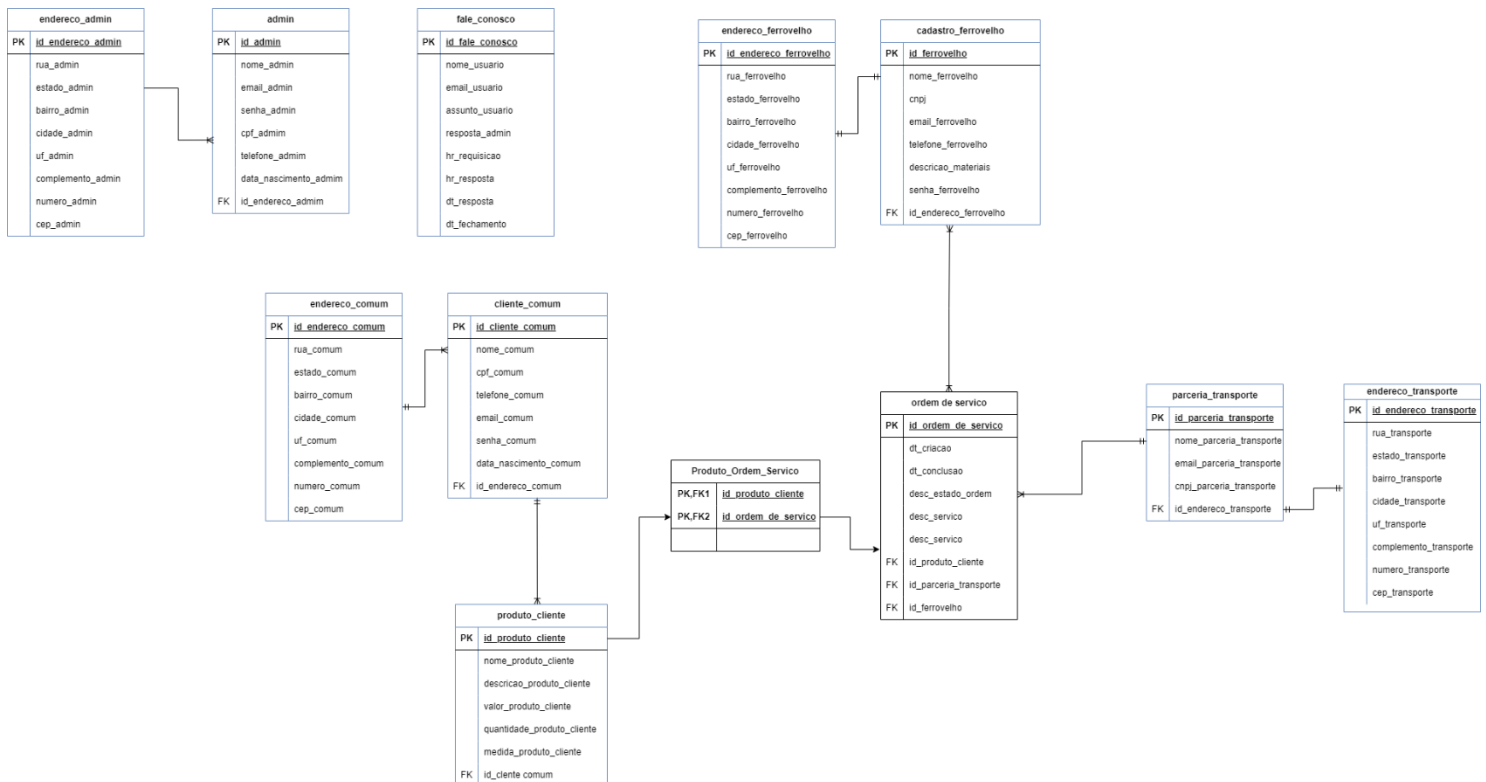


Banco de Dados UpCycle

Na elaboração do nosso banco de dados, iniciamos pelo desenvolvimento da modelagem lógica. Esta etapa envolve a definição precisa e detalhada de elementos como tipos de dados, chaves primárias, chaves estrangeiras, relacionamentos, atributos, e outros detalhes específicos.

Após efetuar o login, o usuário terá a opção de solicitar a coleta dos seus resíduos e acompanhar o status da solicitação. Posteriormente, uma ordem de serviço será gerada, estabelecendo relações com as tabelas de produto, transporte e ferro velho.



Modelo Físico

Por fim, elaboramos o **modelo físico**, partindo do modelo lógico. Nessa etapa, detalhamos como os dados serão armazenados na prática, considerando aspectos como o tamanho dos campos, identificadores, método de preenchimento desses campos e nomenclaturas.

```
create database if not exists upcycle_data;
use upcycle_data;
```

```
create table if not exists endereco_admin(
    id_endereco_admin int primary key auto_increment,
    rua_admin varchar(120) not null,
    estado_admin varchar(120) not null,
    bairro_admin varchar(120) not null,
    cidade_admin varchar(120) not null,
    uf_admin varchar(2) not null,
    complemento_admin varchar(120) not null,
    numero_admin int not null,
    cep_admin varchar(10) not null
);
```

```
CREATE TABLE IF NOT EXISTS fale_conosco (
    id_fale_conosco INT PRIMARY KEY AUTO_INCREMENT,
    nome_usuario VARCHAR(255) NOT NULL,
    email_usuario VARCHAR(255) NOT NULL,
    assunto_usuario VARCHAR(255) NOT NULL,
    resposta_admin VARCHAR(255) NOT NULL,
    hr_requisicao VARCHAR(15) NOT NULL,
    hr_resposta VARCHAR(15) NOT NULL,
    dt_resposta VARCHAR(15) NOT NULL,
    dt_fechamento VARCHAR(15) NOT NULL
);
```

```

create table if not exists endereco_comum (
    id_endereco_comum int primary key auto_increment,
    rua_comum varchar (120) ,
    numero_comum varchar (10) ,
    bairro_comum varchar(120) ,
    cidade_comum varchar(120) ,
    estado_comum varchar(120) ,
    uf_comum varchar(2) ,
    cep_comum varchar(120) ,
    complemento_comum varchar(120)
);

```

```

create table endereco_ferrovelho(
    id_enderecoferrovelho int primary key auto_increment,
    rua_ferrovelho varchar(150) not null,
    numero_ferrovelho varchar(10) not null,
    bairro_ferrovelho varchar(150) not null,
    cidade_ferrovelho varchar(100) not null,
    estado_ferrovelho varchar(100) not null,
    uf_ferrovelho varchar(3) not null,
    cep_ferrovelho varchar(50) not null,
    complemento_ferrovelho varchar(150)
);

```

```

create table if not exists admim (
    id_admin int primary key auto_increment,
    nome_admin varchar (120) not null,
    email_admin varchar(120) not null,
    senha_admin varchar(120) not null,
    cpf_admin varchar(15) not null,
    telefone_admin varchar(20) not null,
    data_nascimento_admin date null,
    id_endereco_admin int,
    foreign key (id_endereco_admin) references endereco_admin(id_endereco_admin) on delete cascade on update cascade
);

```

```

create table cadastro_ferrovelho(
    id_ferrovelho int primary key auto_increment,
    nome_ferrovelho varchar(150) not null,
    cnpj varchar(20) not null unique,
    email_ferrovelho varchar(150) unique,
    telefone_ferrovelho varchar(20),
    descricao_materiais text not null,
    senha_ferrovelho varchar (150) not null,
    id_enderecoferrovelho int,
    foreign key(id_enderecoferrovelho) references enderecoFerrovelho(id_enderecoferrovelho) on delete cascade on update cascade
);

```

```

create table cliente_comum (
    id_cliente_comum int primary key auto_increment,
    nome_comum varchar (120) not null,
    cpf_comum varchar(15) not null,
    email_comum varchar(120) not null,
    telefone_comum varchar(20) not null,
    data_nascimento_comum date not null,
    senha_comum varchar(120) not null,
    id_endereco_comum int,
    foreign key (id_endereco_comum) references endereco_comum(id_endereco_comum) on delete cascade on update cascade
);

```

```

create table produto_cliente (
    id_produto_cliente int primary key auto_increment,
    nome_produto_cliente varchar(120) not null,
    descricao_produto_cliente varchar(120) not null,
    valor_produto_cliente varchar(120) not null,
    quantidade_produto_cliente int(255) not null, |
    medida_produto_cliente varchar(120) not null,
    id_cliente_comum int not null,
    foreign key (id_cliente_comum) references cliente_comum(id_cliente_comum) on delete cascade on update cascade
);

```

```

create table if not exists endereco_transporte (
    id_endereco_transporte int primary key auto_increment,
    rua_transporte varchar (120) not null,
    estado_transporte varchar(120) not null,
    bairro_transporte varchar(120) not null,
    cidade_transporte varchar(120) not null,
    uf_transporte varchar(2) not null,
    complemento_transporte varchar(120) not null,
    numero_transporte int not null,
    cep_transporte varchar(120) not null
);

```

```

create table parceria_transporte (
    id_parceria_transporte int primary key auto_increment,
    nome_parceria_transporte varchar(120) not null,
    email_parceria_transporte varchar(120) not null,
    cnpj_parceria_transporte varchar(15) not null,
    id_endereco_transporte int not null,
    foreign key (id_endereco_transporte) references endereco_transporte(id_endereco_transporte) on delete cascade on update cascade
);

```

```

create table ordem_servico (
    id_ordem_servico int(11) primary key auto_increment,
    dt_criacao date not null,
    dt_conclusao date not null,
    desc_estado_ordem text not null,
    desc_servico text not null,
    id_produto_cliente int not null,
    id_parceria_transporte int not null,
    id_ferrovelho int not null,
    id_cliente_comum int not null,
    foreign key (id_produto_cliente) references produto_cliente(id_produto_cliente) on delete cascade on update cascade,
    foreign key (id_parceria_transporte) references parceria_transporte(id_parceria_transporte) on delete cascade on update cascade,
    foreign key (id_ferrovelho) references cadastroFerrovelho(id_ferrovelho) on delete cascade on update cascade
);

create table if not exists produto_ordem_servico (
    produto_ordem_servico int auto_increment primary key,
    id_produto_cliente int(11) not null,
    id_ordem_servico int(11) not null,
    foreign key (id_produto_cliente) references produto_cliente (id_produto_cliente) on delete cascade on update cascade ,
    foreign key (id_ordem_servico) references ordem_servico (id_ordem_servico) on delete cascade on update cascade
);

```

Lei Geral da Proteção de Dados

O propósito da Lei Geral de Proteção de Dados (LGPD) é estabelecer um conjunto regulatório para o manejo de informações pessoais no Brasil, abrangendo tanto ambientes online quanto offline, tanto nos setores públicos quanto privados. De maneira geral, a LGPD estipula que dados pessoais devem ser processados exclusivamente para fins lícitos, específicos, explícitos e claramente definidos. Assim como no Regulamento Geral de Proteção de Dados (GDPR), os princípios de transparência e minimização de dados (utilizando apenas as informações estritamente necessárias) são aplicáveis.

A LGPD assegura a todos os cidadãos o direito à privacidade de suas informações pessoais, tais como nome, endereço, e-mail, idade e estado civil. Além disso, impõe obrigações às plataformas online, exigindo que estas esclareçam de forma transparente como os dados são tratados, armazenados e com qual finalidade.

Na UpCycle, zelamos pela privacidade e segurança das informações fornecidas em nossa plataforma, em conformidade com a Lei Geral de Proteção de Dados (LGPD). Ressaltamos que algumas informações podem ser compartilhadas com parceiros estratégicos para melhor atender às

necessidades dos usuários. No entanto, asseguramos que tais compartilhamentos são realizados de maneira responsável e em conformidade com as normas de privacidade estabelecidas pela LGPD, visando sempre a proteção dos dados e a transparência em nossas práticas.