

# **REQUISITOS, MÓDELOS E FRONT-END**

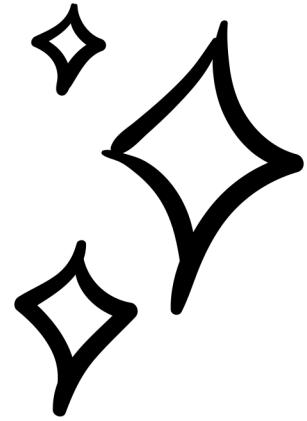
**Bernardo Lara – Felipe Campolina –  
Gabriela Colen Marcelo Augusto –  
Leandro Guido**



# ESPECIFICAÇÃO FUNCIONAL

- Organização por meio de um banco de dados**
- Auxiliar a navegação do idoso**
- Estimular o feedback de familiares**
- Manipulação de conteúdos/usuários**
- Estipular 2 tipos de usuário**
- Ensinar como usar as ferramentas do próprio site**

# COMO SERÃO IMPLEMENTADOS?



## Auxiliar a navegação do idoso

Layout simples, site interativo,

## Estimular o feedback de familiares

Benefícios ao responder as pesquisas

## Estipular 2 tipos de usuários

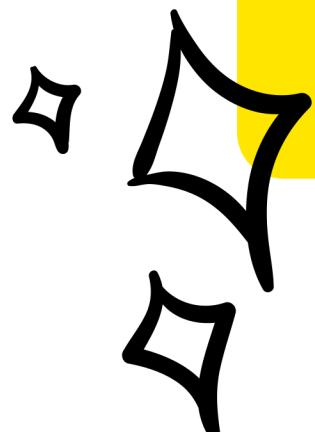
Disponibilização de material gratuito e pago

## Manipulação de conteúdos/usuários

Sistema de administração para controle de segurança e alteração de conteúdos

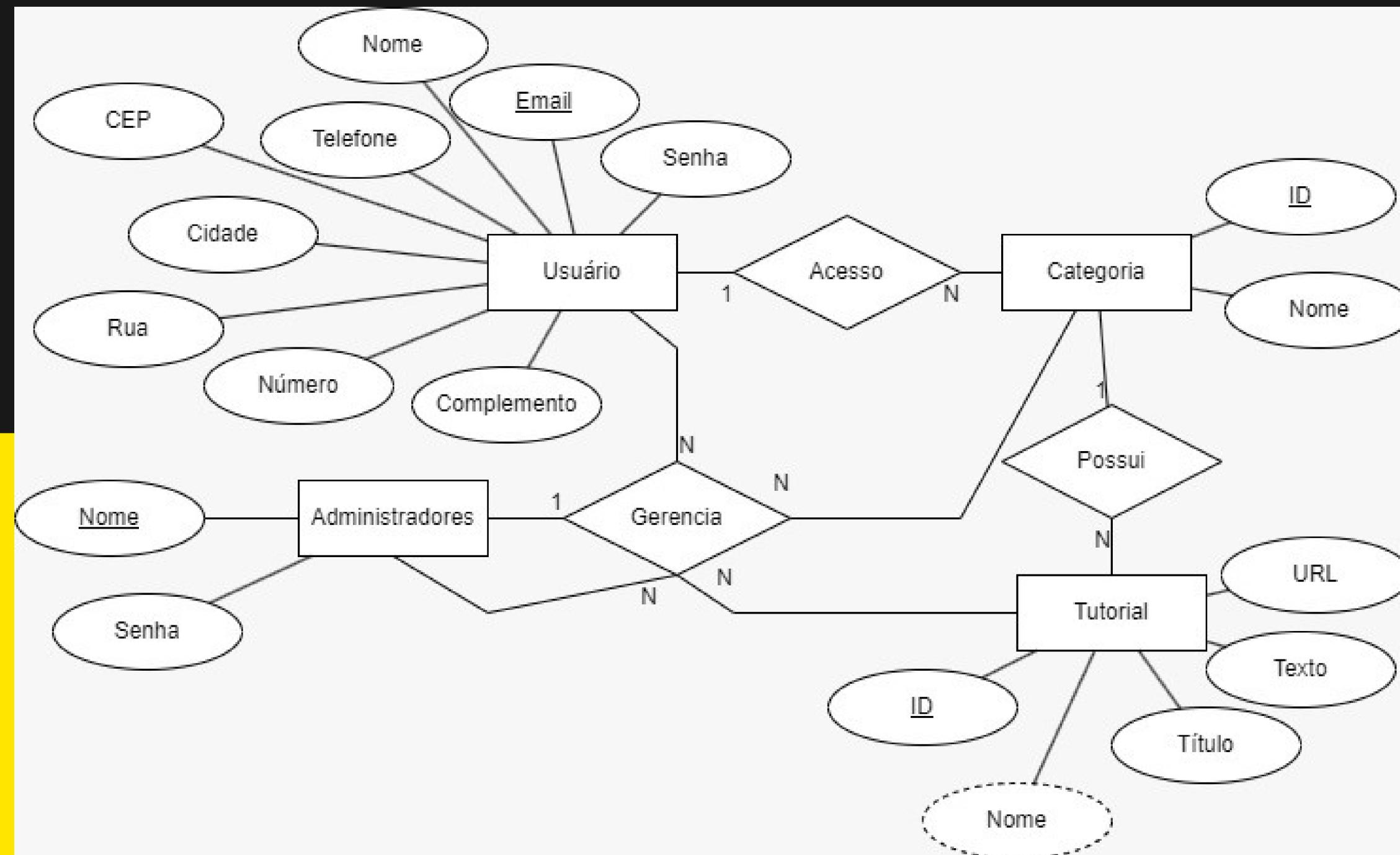
## Ensinar como usar as ferramentas do próprio site

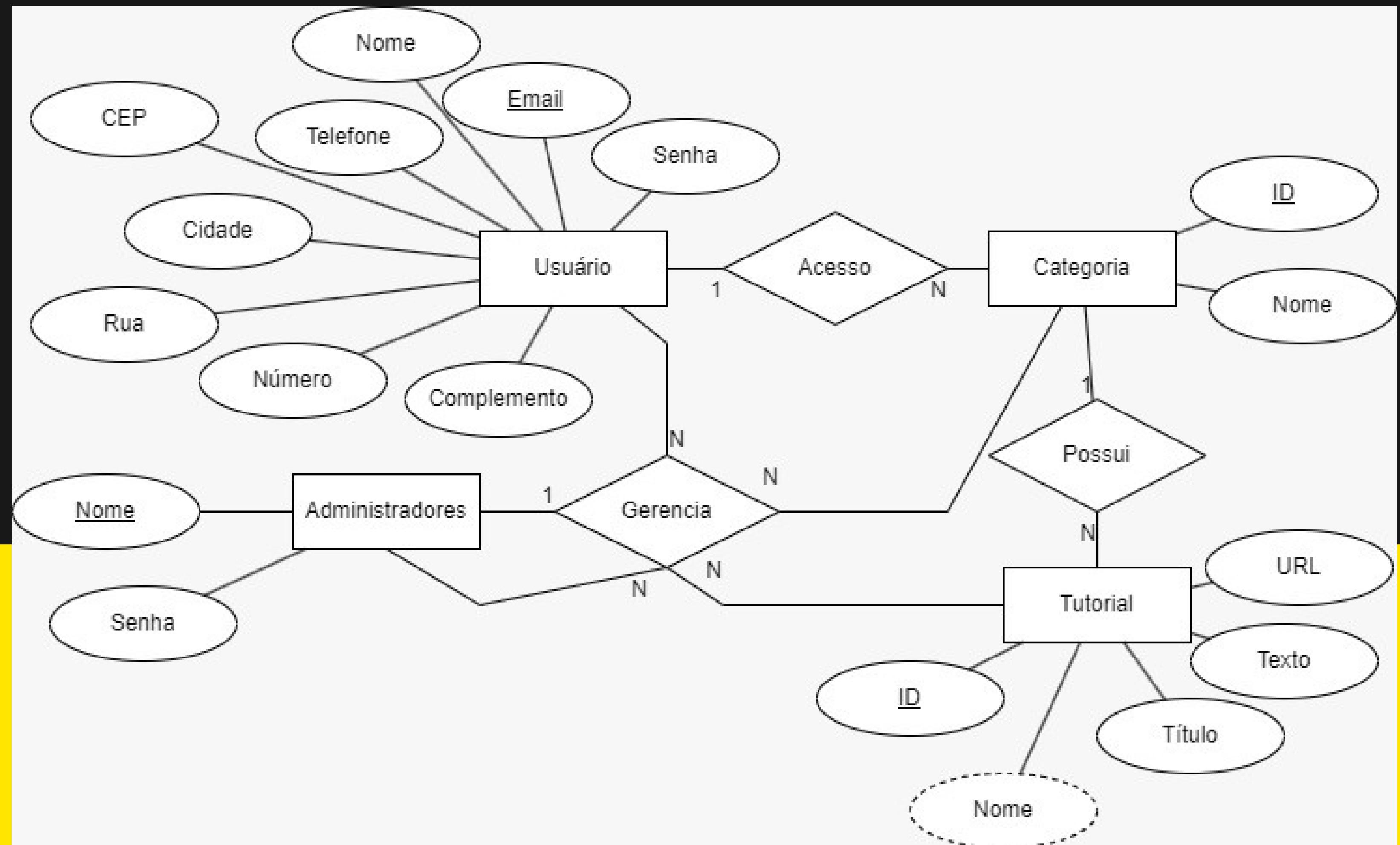
Implementação de um boot educacional e tutoriais



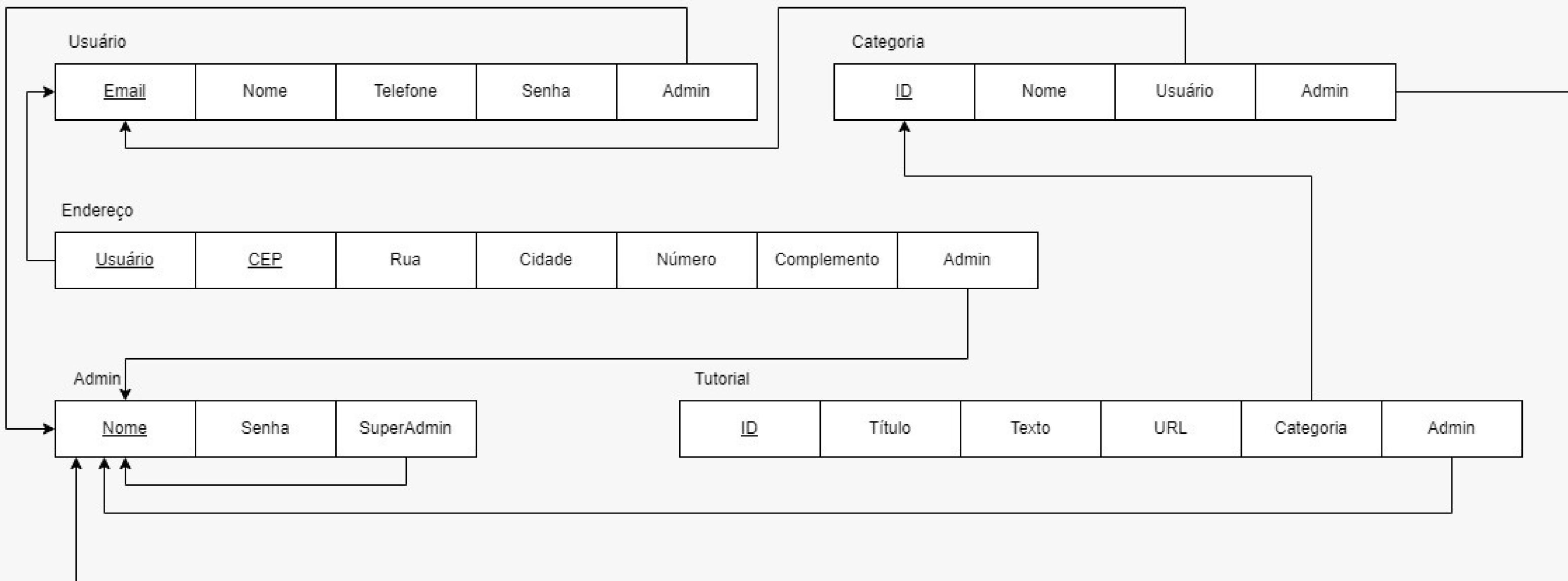
**QUAIS SÃO AS ENTIDADES,  
ATRIBUTOS, RELACIONAMENTO  
E RESTRIÇÕES DO BANCO DE  
DADOS?**

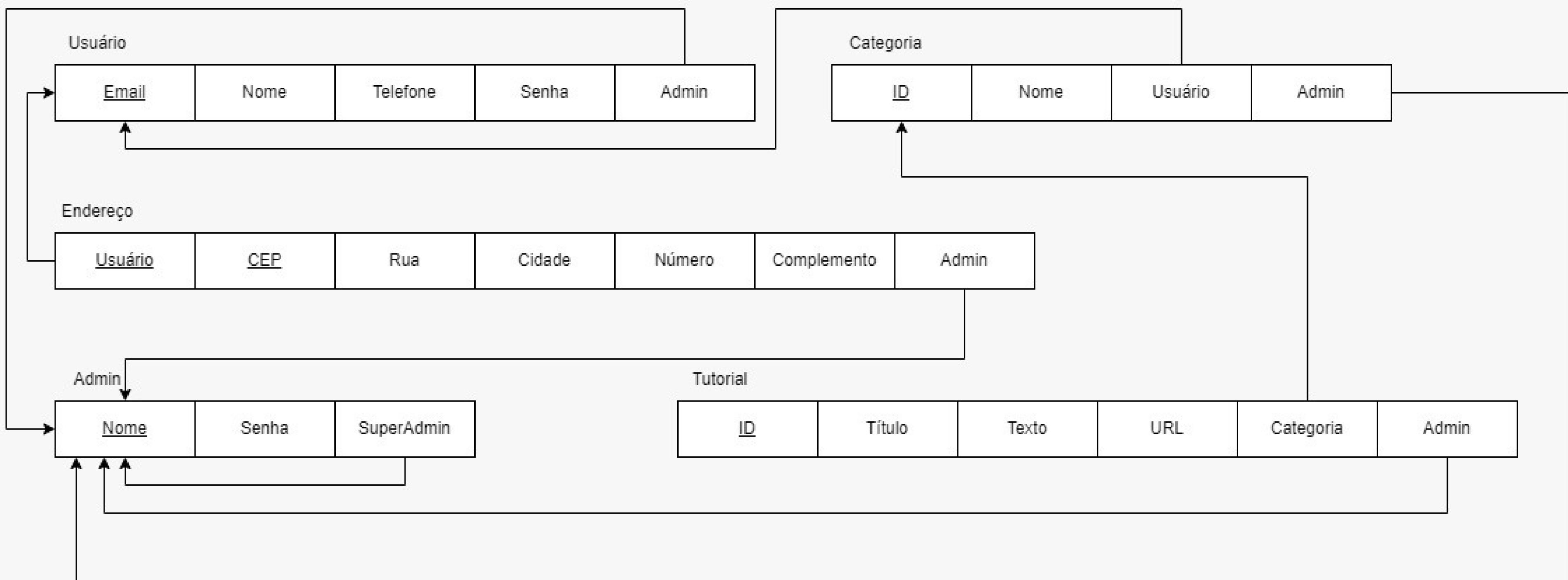
# MODELO CONCEITUAL

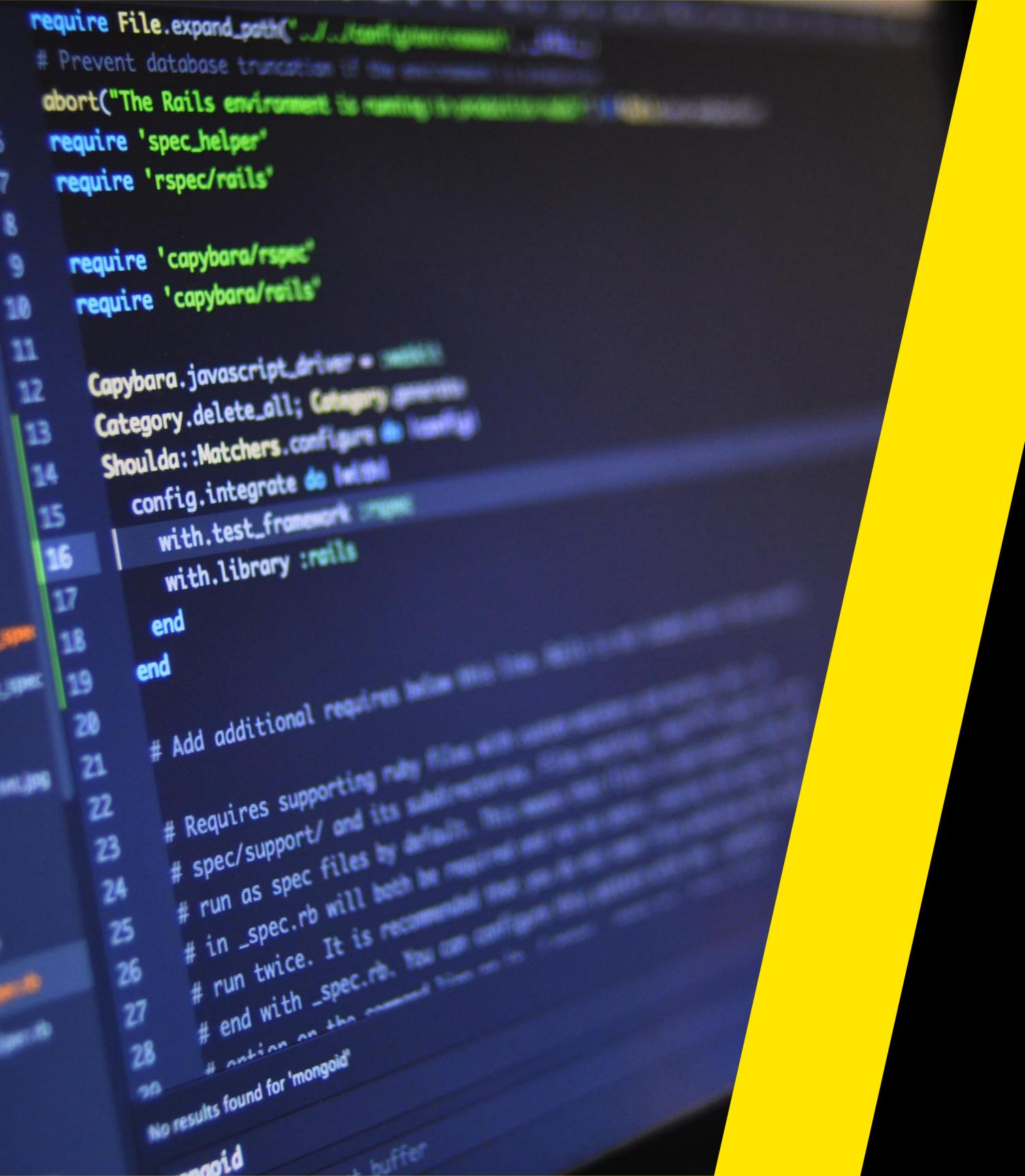




# MODELO LÓGICO

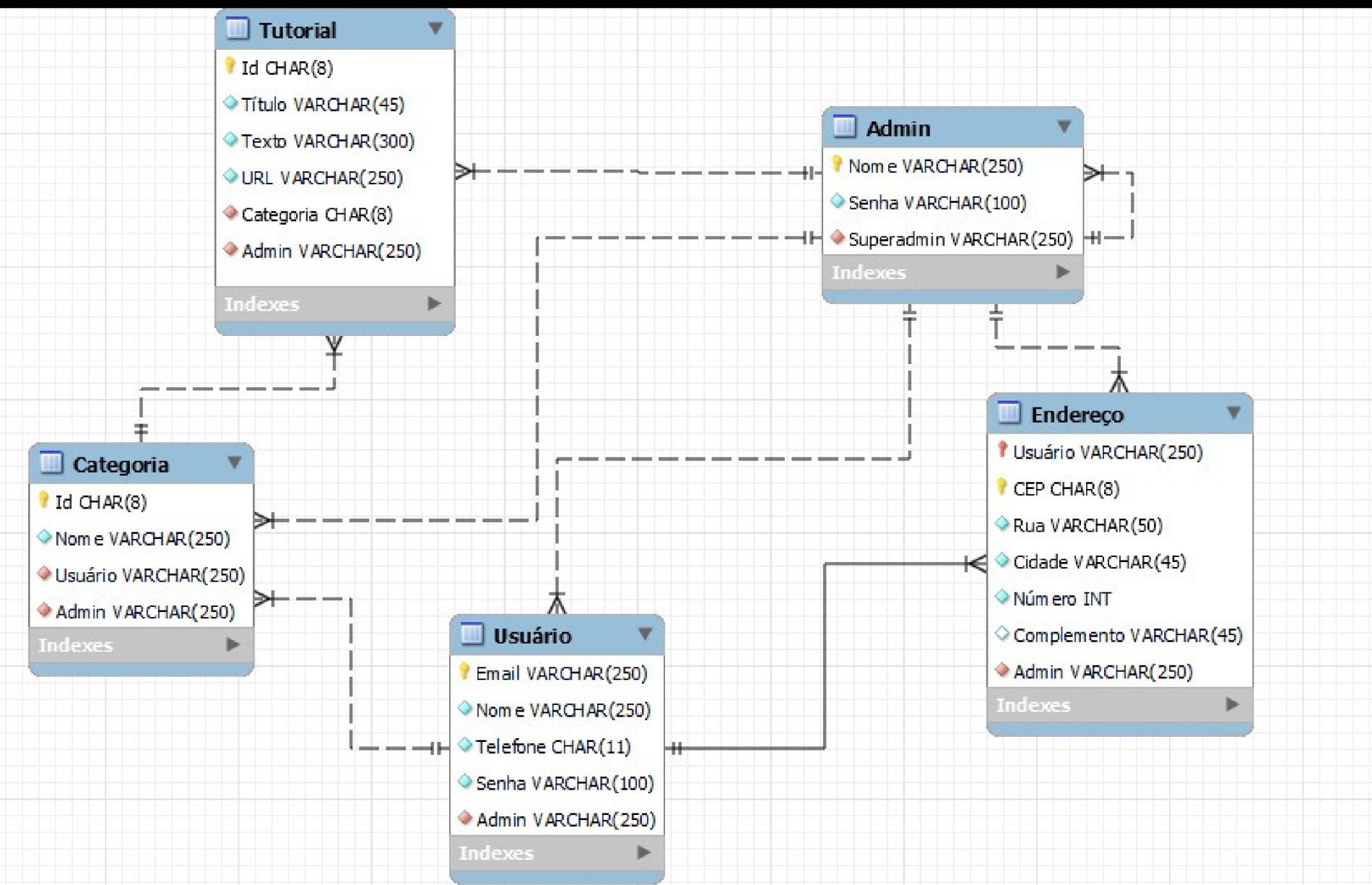


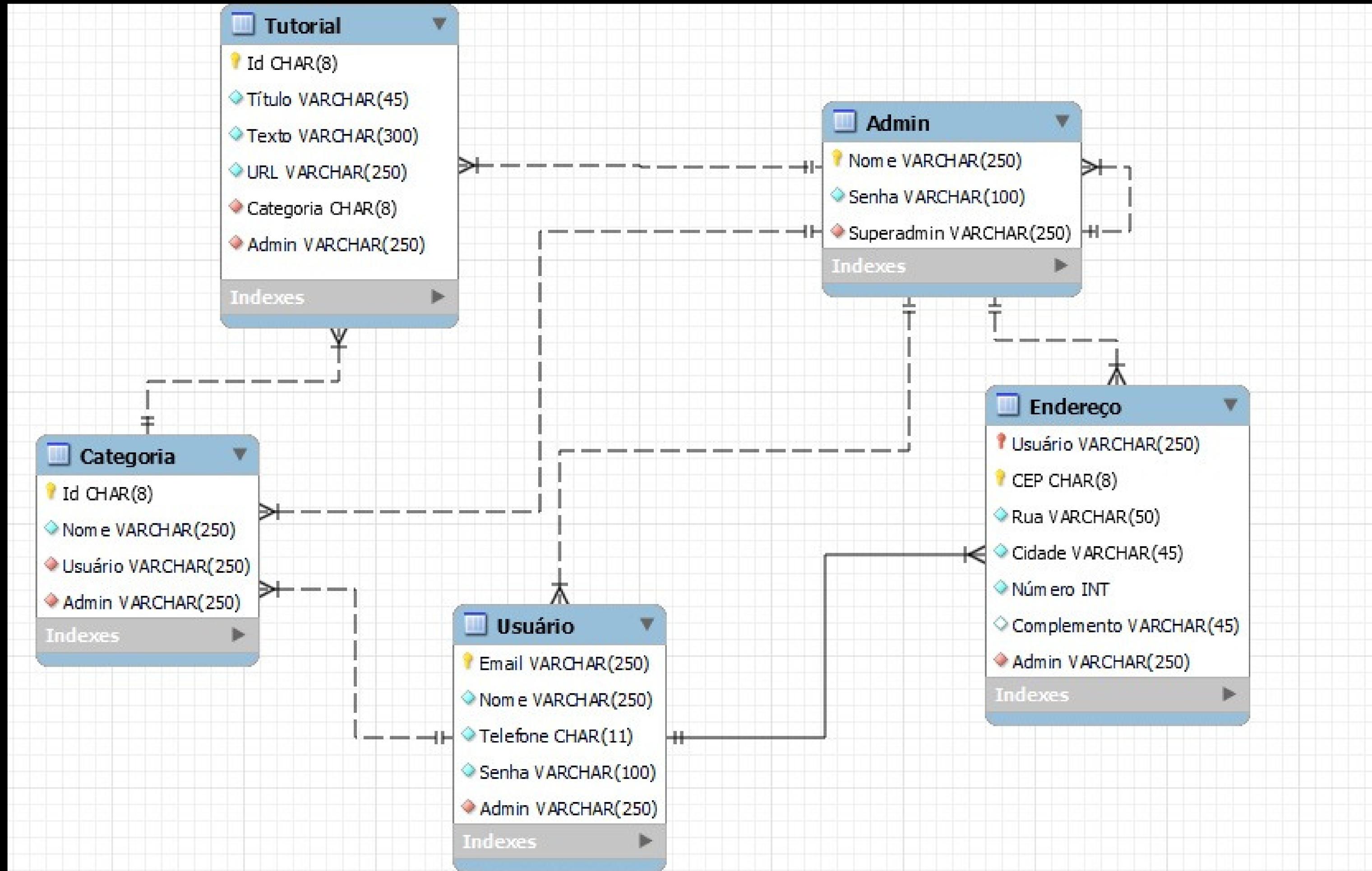




# **COMO ESSES ELEMENTOS CONCEITUAIS DO BANCO DE DADOS SERÃO IMPLEMENTADOS (MAPERDOS) NUM SGBD RELACIONAL?**

# MODELO DE IMPLEMENTAÇÃO





## SCRIPTS - BANCO DE DADOS

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRORS_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----
-- Schema mydb
```

```
-----  
DROP SCHEMA IF EXISTS `mydb`;
```

```
-- -----
-- Schema mydb
```

```
-----  
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb`;
```

```
-- -----
-- Table `mydb`.`Admin`
```

```
-----  
DROP TABLE IF EXISTS `mydb`.`Admin`;
```

## SCRIPTS - BANCO DE DADOS

```
CREATE TABLE IF NOT EXISTS `mydb`.`Admin` (
  `Nome` VARCHAR(250) NOT NULL,
  `Senha` VARCHAR(100) NOT NULL,
  `Superadmin` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`Nome`),
  INDEX `fk_Admin_Admin1_idx` (`Superadmin` ASC) VISIBLE,
  CONSTRAINT `fk_Admin_Admin1`
    FOREIGN KEY (`Superadmin`)
    REFERENCES `mydb`.`Admin` (`Nome`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Categoria`
```

```
DROP TABLE IF EXISTS `mydb`.`Categoria`;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Categoria` (
  `Id` CHAR(8) NOT NULL,
```

```
  `Nome` VARCHAR(250) NOT NULL,
  `Usuário` VARCHAR(250) NOT NULL,
  `Admin` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`Id`),
  INDEX `fk_Categoria_Usuário1_idx` (`Usuário` ASC) VISIBLE,
  INDEX `fk_Categoria_Admin1_idx` (`Admin` ASC) VISIBLE,
  CONSTRAINT `fk_Categoria_Usuário1`
    FOREIGN KEY (`Usuário`)
    REFERENCES `mydb`.`Usuário` (`Email`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Categoria_Admin1`
    FOREIGN KEY (`Admin`)
    REFERENCES `mydb`.`Admin` (`Nome`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Endereço`
```

```
DROP TABLE IF EXISTS `mydb`.`Endereço`;
```

## SCRIPTS - BANCO DE DADOS

```
CREATE TABLE IF NOT EXISTS `mydb`.`Endereço` (
  `Usuário` VARCHAR(250) NOT NULL,
  `CEP` CHAR(8) NOT NULL,
  `Rua` VARCHAR(50) NOT NULL,
  `Cidade` VARCHAR(45) NOT NULL,
  `Número` INT NOT NULL,
  `Complemento` VARCHAR(45) NULL,
  `Admin` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`CEP`, `Usuário`),
  INDEX `fk_Endereço_Usuário_idx` (`Usuário` ASC) VISIBLE,
  INDEX `fk_Endereço_Admin1_idx` (`Admin` ASC) VISIBLE,
  CONSTRAINT `fk_Endereço_Usuário`
    FOREIGN KEY (`Usuário`)
    REFERENCES `mydb`.`Usuário` (`Email`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Endereço_Admin1`
    FOREIGN KEY (`Admin`)
    REFERENCES `mydb`.`Admin` (`Nome`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `mydb`.`Tutorial`
-----
DROP TABLE IF EXISTS `mydb`.`Tutorial`;

CREATE TABLE IF NOT EXISTS `mydb`.`Tutorial` (
  `Id` CHAR(8) NOT NULL,
  `Título` VARCHAR(45) NOT NULL,
  `Texto` VARCHAR(300) NOT NULL,
  `URL` VARCHAR(250) NOT NULL,
  `Categoria` CHAR(8) NOT NULL,
  `Admin` VARCHAR(250) NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `Id_UNIQUE` (`Id` ASC) VISIBLE,
  INDEX `fk_Tutorial_Categoria1_idx` (`Categoria` ASC)
  VISIBLE,
  INDEX `fk_Tutorial_Admin1_idx` (`Admin` ASC) VISIBLE,
  CONSTRAINT `fk_Tutorial_Categoria1`
    FOREIGN KEY (`Categoria`)
    REFERENCES `mydb`.`Categoria` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

## SCRIPTS - BANCO DE DADOS

```
CONSTRAINT `fk_Tutorial_Admin1`  
FOREIGN KEY (`Admin`)  
REFERENCES `mydb`.`Admin` (`Nome`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`Usuário`
```

```
DROP TABLE IF EXISTS `mydb`.`Usuário`;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Usuário` (  
`Email` VARCHAR(250) NOT NULL,  
`Nome` VARCHAR(250) NOT NULL,  
`Telefone` CHAR(11) NOT NULL,  
`Senha` VARCHAR(100) NOT NULL,  
`Admin` VARCHAR(250) NOT NULL,  
PRIMARY KEY (`Email`),  
UNIQUE INDEX `EmailUsuário_UNIQUE` (`Email` ASC) VISIBLE,  
UNIQUE INDEX `Telefone_UNIQUE` (`Telefone` ASC) VISIBLE,
```

```
INDEX `fk_Usuário_Admin1_idx` (`Admin` ASC) VISIBLE,  
CONSTRAINT `fk_Usuário_Admin1`  
FOREIGN KEY (`Admin`)  
REFERENCES `mydb`.`Admin` (`Nome`)  
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_CHECKS;
```



**SITE**

**QUAIS SÃO OS ELEMENTOS DE  
INTERFACE (FRONT-END) DO  
SISTEMA E COMO ELES SE  
RELACIONAM (NAVEGAÇÃO)  
ENTRE SI?**



**PUC Minas**