

# Projeto Rotina Diária

**Título:** Sistema de Gestão de Tarefas Diárias

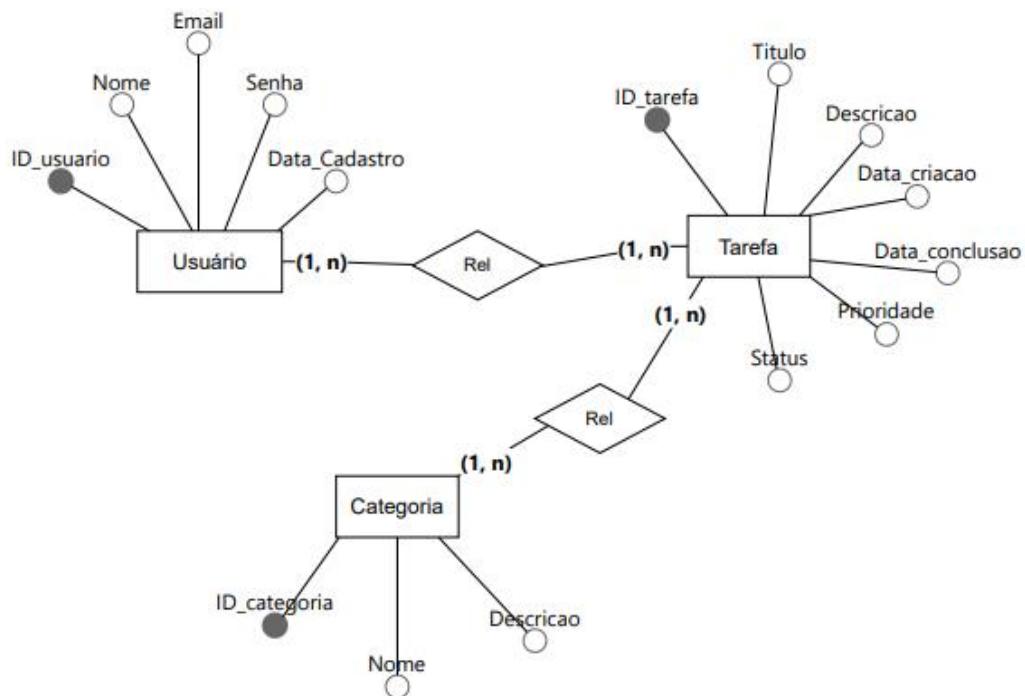
**Autor:** Valdemir Valentin Santos, João Pedro Fonseca

## 1. Introdução

**Objetivo do banco de dados:**

Gerenciar tarefas de usuários, categorizando-as e acompanhando status e prioridade.

## 2. Modelo Conceitual



**Entidades e Atributos:**

## **Usuário**

- id\_usuario (PK)
- nome
- email
- senha
- data\_cadastro

## **Tarefa**

- id\_tarefa (PK)
- titulo
- descricao
- data\_criacao
- data\_conclusao
- prioridade
- status

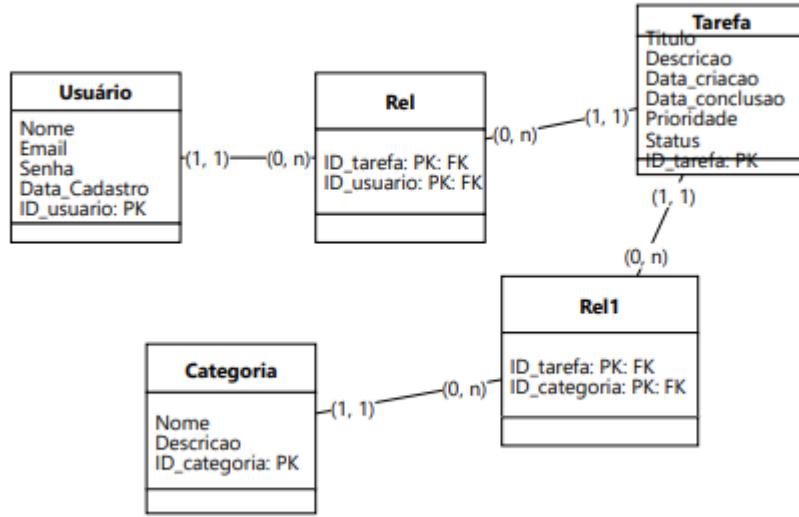
## **Categoria**

- id\_categoria (PK)
- nome
- descricao

## **Relacionamentos:**

- Usuário → Tarefa: 1:N
- Tarefa → Categoria: 1:N

## **3. Modelo Lógico**



### Entidades, Atributos e Tipos de Dados (genéricos):

#### Usuário

- id\_usuario (PK, INT)
- nome (VARCHAR)
- email (VARCHAR)
- senha (VARCHAR)
- data\_cadastro (DATETIME)

#### Tarefa

- id\_tarefa (PK, INT)
- titulo (VARCHAR)
- descricao (TEXT)
- data\_criacao (DATETIME)
- data\_conclusao (DATETIME)
- prioridade (ENUM/VARCHAR)
- status (ENUM/VARCHAR)
- id\_usuario (FK → Usuário)

#### Categoria

- id\_categoria (PK, INT)
- nome (VARCHAR)

- descricao (TEXT)

### **Tarefa\_Categoria**

- id\_tarefa (FK → Tarefa)
- id\_categoria (FK → Categoria)
- PK composta (id\_tarefa, id\_categoria)

### **Relacionamentos:**

- Usuário → Tarefa: 1:N
- Tarefa → Categoria: 1:N

## **4. Modelo Físico**

**(MySQL):**

```
CREATE TABLE Usuario (
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(255),
    data_cadastro DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE Categoria (
    id_categoria INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL UNIQUE,
    descricao TEXT
);

CREATE TABLE Tarefa (
    id_tarefa INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(100) NOT NULL,
    descricao TEXT,
    data_criacao DATETIME DEFAULT CURRENT_TIMESTAMP,
    data_conclusao DATETIME NULL,
    prioridade ENUM('baixa','media','alta') DEFAULT 'media',
    status ENUM('pendente','em_andamento','concluida') DEFAULT
    'pendente',
    id_usuario INT NOT NULL,
```

```

FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

## 5. Diagrama de caso de uso

```

@startuml
!theme plain

title Diagrama de Caso de Uso - Sistema de Rotina Diária

' ===== Atores =====
actor "Usuário" as Usuario
actor "Administrador" as Admin

' ===== Sistema =====
rectangle "Sistema de Rotina Diária" {

    (Cadastrar-se) as UC_Cadastrar
    (Fazer login) as UC_Login
    (Gerenciar tarefas) as UC_Gerenciar
    (Criar tarefa) as UC_Criar
    (Editar tarefa) as UC_Editar
    (Excluir tarefa) as UC_Excluir
    (Marcar tarefa como concluída) as UC_Concluir
    (Visualizar rotina diária) as UC_Visualizar
    (Gerenciar usuários) as UC_Usuarios

    ' Relacionamentos internos
    UC_Gerenciar --> UC_Criar : <<include>>
    UC_Gerenciar --> UC_Editar : <<include>>
    UC_Gerenciar --> UC_Excluir : <<include>>
    UC_Gerenciar --> UC_Concluir : <<include>>
}

' ===== Ligações dos atores =====
Usuario --> UC_Cadastrar
Usuario --> UC_Login
Usuario --> UC_Gerenciar
Usuario --> UC_Visualizar

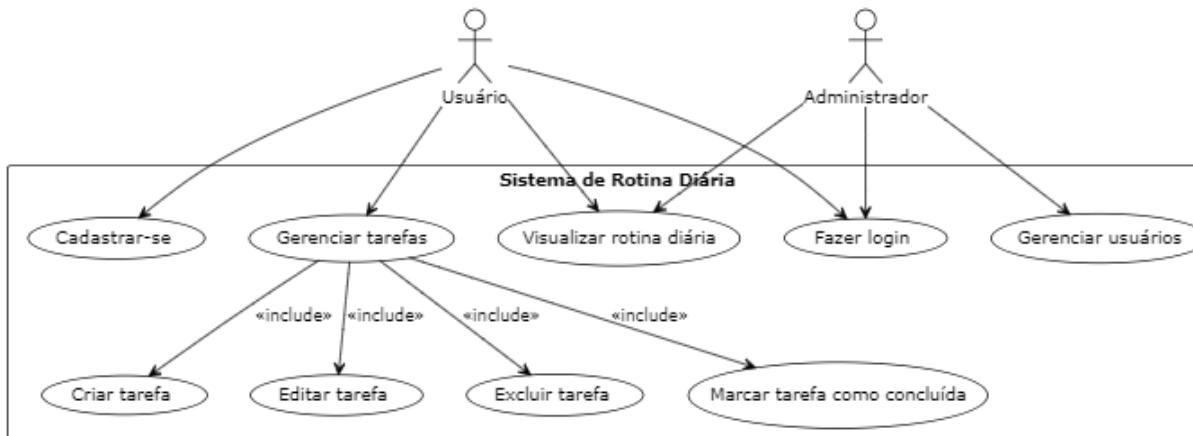
Admin --> UC_Login

```

```
Admin --> UC_Usuarios  
Admin --> UC_Visualizar
```

```
@enduml
```

**Diagrama de Caso de Uso - Sistema de Rotina Diária**



## 6. Diagrama de classe

```
@startuml  
!theme plain  
left to right direction  
skinparam class {  
    BackgroundColor #F9F9F9  
    BorderColor #444  
    ArrowColor #000  
}  
title Diagrama de Classes - Sistema de Rotina Diária  
  
' === Clase Usuario ===  
class Usuario {  
    - id : int  
    - nome : string  
    - email : string  
    - senha : string  
    - tipo : enum('admin','tecnico','usuario')  
    - criado_em : timestamp  
  
    + cadastrar()  
    + fazerLogin()  
    + gerenciarTarefas()  
}  
  
' === Clase Tarefa ===
```

```

class Tarefa {
    - id : int
    - usuario_id : int
    - titulo : string
    - descricao : text
    - data : date
    - hora : time
    - icone : string
    - cor : string
    - status : enum('pendente','concluida')
    - criado_em : timestamp
    - atualizado_em : timestamp

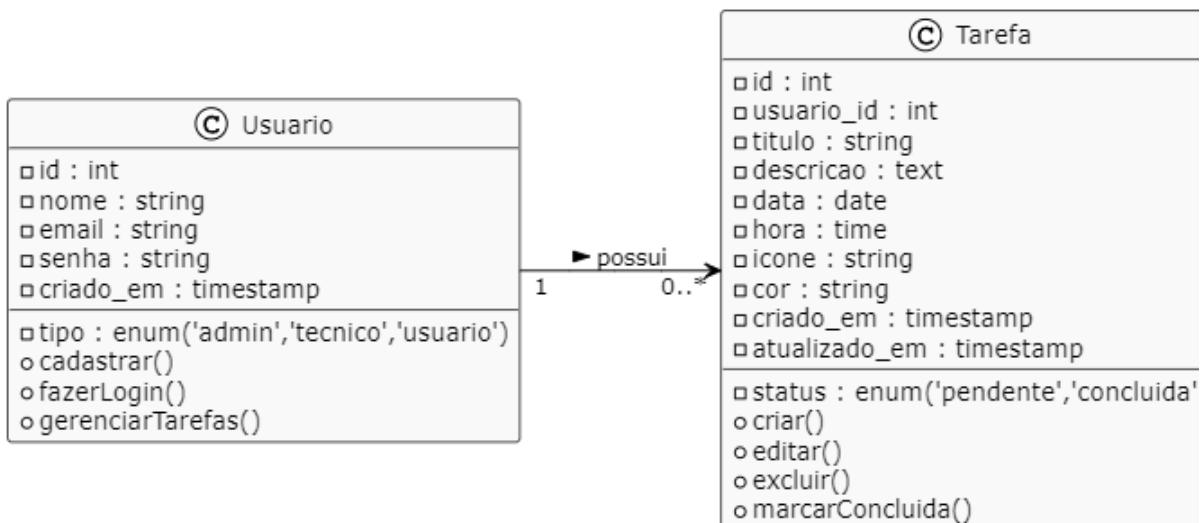
    + criar()
    + editar()
    + excluir()
    + marcarConcluida()
}

' ===== Relacionamento =====
Usuario "1" --> "0..*" Tarefa : possui >

@enduml

```

## Diagrama de Classes - Sistema de Rotina Diária



## 7. Considerações Finais

- Integridade referencial garantida via PKs e FKs.

- Normalização aplicada (1:N e N:N).
- Possíveis melhorias futuras: índices adicionais, triggers, otimizações de performance.
- Modelo físico serve como base para implementação real no SGBD.