

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CTC - CENTRO TECNOLÓGICO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Aline Cristina Meyer
Elianthony Gonçalves Sabino Gabriel**

SECTASK - SECURITY TASK MANAGER

**Florianópolis
2023**

O SecTask é um gerenciador de tarefas focado para equipes de Segurança da Informação. O sistema visa permitir que os usuários possam adicionar novas tarefas, acompanhar o progresso, definir prioridades, atribuir responsabilidades e entre outras funcionalidades. Nele poderemos registrar equipes, projetos, documentação de orientação, funcionários e tarefas para acompanhar o dia a dia do desenvolvimento dos projetos internos das equipes responsáveis, esperamos que os times de segurança consigam obter um ganho em organização, produtividade e priorização das atividades pendentes com o SecTask.

Nós escolhemos as entidades Funcionário, Tarefa, Projeto, Equipe, Categoria e Documentação. Abaixo estão os atributos das respectivas entidades, segue:

- Funcionário
 - ID do funcionário (chave primária)
 - Nome
 - Endereço de e-mail
 - Telefone
 - Endereço
 - Data de registro
 - Supervisor(FK proprio)
 - **Equipe (chave estrangeira referenciando a equipe)**
- Tarefa
 - ID da tarefa (chave primária)
 - Título
 - Descrição
 - Data de vencimento
 - Status (pendente, concluída, atrasada)
 - Prioridade (baixa, média, alta)
 - Data de criação
 - Data de conclusão (quando aplicável)
 - **Projeto (Chave estrangeira referenciando o projeto)**
- Projeto
 - ID do projeto (chave primária)
 - Nome do projeto
 - Descrição do projeto
 - Data de início
 - Data de conclusão prevista

- **Equipe Responsável (chave estrangeira referenciando a Equipe)**
 - **Categoria (chave estrangeira referenciando a categoria)**
- Equipe
 - ID da equipe (chave primária)
 - Nome da equipe
 - Descrição da equipe
 - Data de formação
 - Apoio
- Categoria
 - ID da categoria (chave primária)
 - Nome da categoria
 - Descrição da categoria
- Documentação
 - ID do documento (chave primária)
 - Título do documento
 - Data da publicação
 - Blob (armazena o documento em binário)
 - **Projeto (chave estrangeira referenciando o projeto)**
 - **Autor do documento (chave estrangeira referenciando o funcionário)**
- FuncionarioEquipe
 - dtAlocacao (chave primária)
 - **Equipe (chave estrangeira referenciando a Equipe)**
 - **Funcionário (chave estrangeira referenciando o funcionário)**
- TarefaAlocada
 - dtAlocacao (chave primária)
 - **Funcionário (chave estrangeira referenciando o funcionário)**
 - **Tarefa ((chave estrangeira referenciando a Tarefa)**
- DocumentoProjeto
 - dtAssociacao
 - **Projeto (chave estrangeira referenciando o projeto)**
 - **Documentacao (chave estrangeira referenciando o documento)**

As entidades possuem relacionamentos e estes são expressos por meio de chaves estrangeiras e cardinalidades. A modelagem de banco Entidade-Relacionamento facilita a organização e a recuperação de informações no sistema. Apresentamos a modelagem conceitual do diagrama ER e a descrição de seus relacionamentos entre as entidades.

- Uma Categoria pode ter no mínimo zero Projeto e no máximo N.
- Um Projeto pode ter no mínimo uma Categoria e no máximo uma.
- Um Projeto pode ter no mínimo uma Equipe e no máximo uma.
- Um Projeto pode ter no mínimo zero Documentação e no máximo N.
- Uma Equipe pode ter no mínimo zero Projeto e no máximo N.
- Uma Equipe pode ter no mínimo zero Funcionário e no máximo N.
- Um Funcionário pode ter no mínimo zero Tarefa e no máximo N.
- Um Funcionário pode ter no mínimo uma Equipe e no máximo uma.
- Um Funcionário pode ter no mínimo zero Documentação e no máximo N.
- Uma Tarefa pode ter no mínimo um Projeto e no máximo um.
- Uma Tarefa pode ter no mínimo zero Funcionário e no máximo um.
- Uma Documentação pode ter no mínimo um Projeto e no máximo um.
- Uma Documentação pode ter no mínimo um Funcionário e no máximo um.

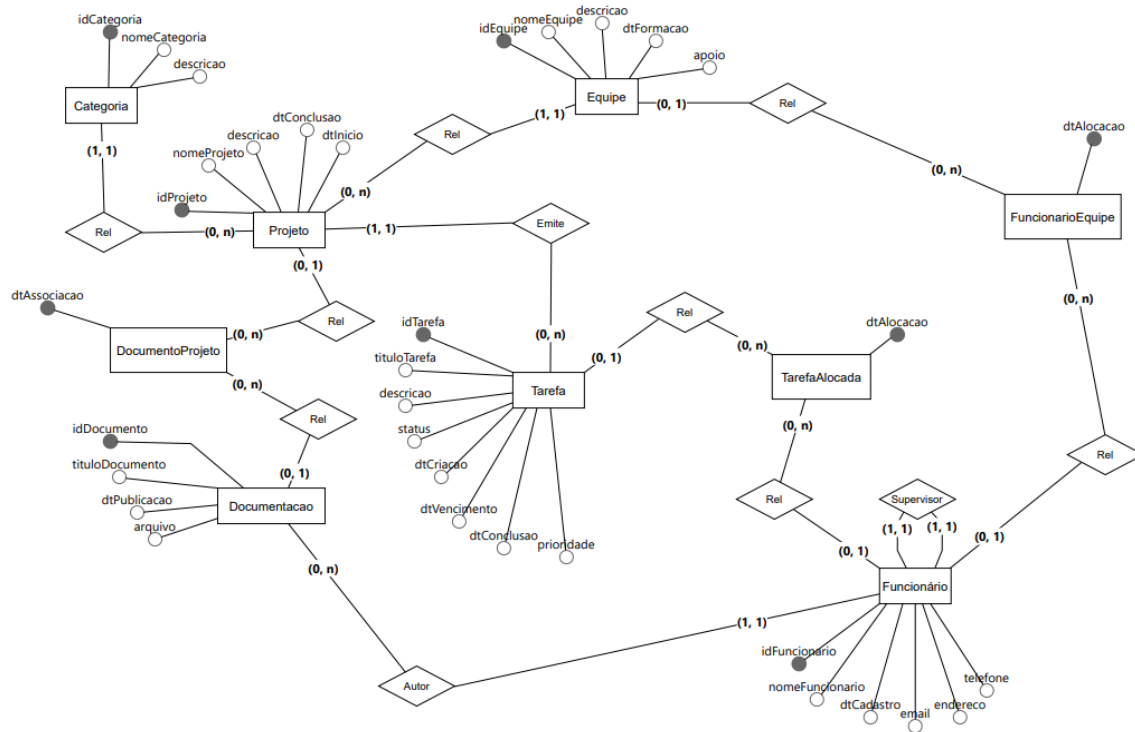


Imagem 1 - Modelagem Conceitual

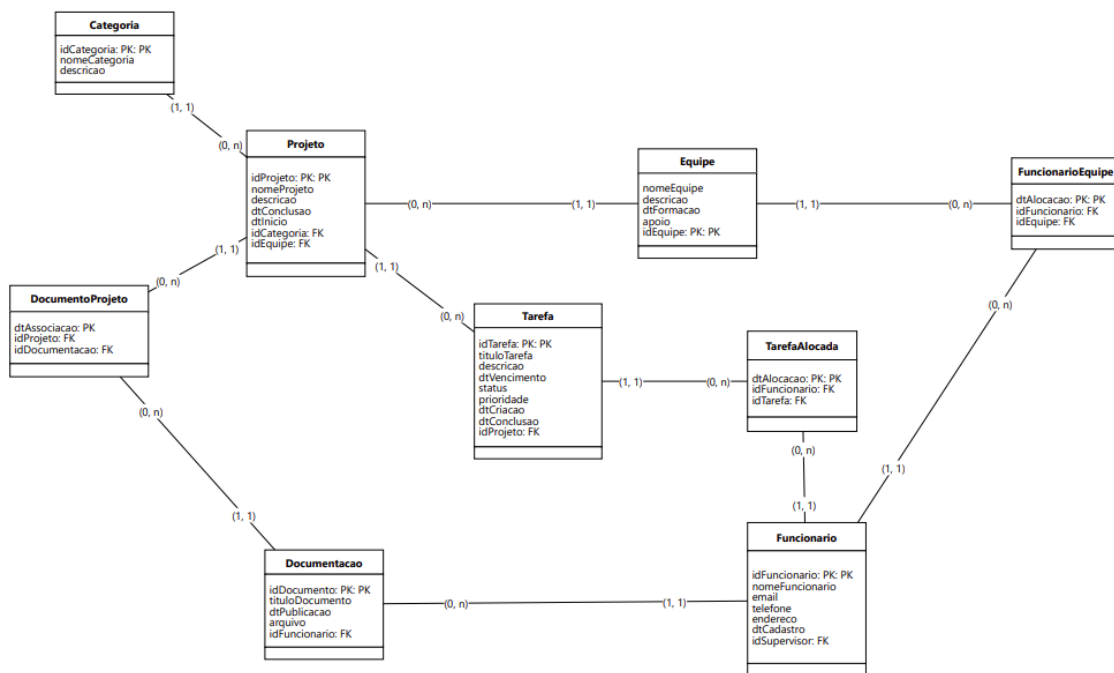


Imagem 2 - Modelagem Logica

5. Script DDL obtido a partir da Modelagem Logica:

```
CREATE TABLE Funcionario (  
  idFuncionario INT PRIMARY KEY AUTO_INCREMENT,  
  nomeFuncionario VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL,  
  telefone INT NOT NULL,  
  endereco VARCHAR(255) NOT NULL,  
  dtCadastro DATE NOT NULL,  
  idSupervisor INT,  
  CHECK (email LIKE '%_@__%.__%'),  
  UNIQUE (email)  
);
```

```
CREATE TABLE Tarefa (  
  idTarefa INT PRIMARY KEY AUTO_INCREMENT,  
  tituloTarefa VARCHAR(255) NOT NULL,  
  descricao VARCHAR(255) NOT NULL,  
  dtVencimento DATE,  
  status CHAR(1) NOT NULL DEFAULT 'P',  
  prioridade CHAR(1) NOT NULL,  
  dtCriacao DATE NOT NULL,  
  dtConclusao DATE,  
  idProjeto INT NOT NULL,  
  CHECK (status IN ('P', 'A', 'C') AND prioridade IN ('A', 'C'))  
);
```

```
CREATE TABLE Projeto (  
  idProjeto INT PRIMARY KEY AUTO_INCREMENT,  
  nomeProjeto VARCHAR(255) NOT NULL,  
  descricao VARCHAR(255) NOT NULL,  
  dtConclusao DATE,  
  dtInicio DATE NOT NULL,  
  idCategoria INT NOT NULL,  
  idEquipe INT,  
  UNIQUE (nomeProjeto)  
);
```

```
CREATE TABLE Equipe (  
  nomeEquipe VARCHAR(255) NOT NULL,  
  descricao VARCHAR(255) NOT NULL,  
  dtFormacao DATE NOT NULL,  
  apoio CHAR(1) NOT NULL DEFAULT 'N',  
  idEquipe INT PRIMARY KEY AUTO_INCREMENT,
```

```
CHECK (apoio IN ('N', 'S'))  
);
```

```
CREATE TABLE Categoria (  
  idCategoria INT PRIMARY KEY AUTO_INCREMENT,  
  nomeCategoria VARCHAR(255) NOT NULL,  
  descricao VARCHAR(255) NOT NULL,  
  UNIQUE (nomeCategoria)  
);
```

```
CREATE TABLE Documentacao (  
  idDocumento INT PRIMARY KEY AUTO_INCREMENT,  
  tituloDocumento VARCHAR(255) NOT NULL,  
  dtPublicacao DATE NOT NULL,  
  arquivo VARCHAR(255) NOT NULL,  
  idFuncionario INT NOT NULL  
);
```

```
CREATE TABLE DocumentoProjeto (  
  dtAssociacao INT PRIMARY KEY,  
  idProjeto INT NOT NULL,  
  idDocumentacao INT NOT NULL  
);
```

```
CREATE TABLE TarefaAlocada (  
  dtAlocacao INT PRIMARY KEY,  
  idFuncionario INT NOT NULL,  
  idTarefa INT NOT NULL  
);
```

```
CREATE TABLE FuncionarioEquipe (  
  dtAlocacao INT PRIMARY KEY,  
  idFuncionario INT NOT NULL,  
  idEquipe INT NOT NULL  
);
```

```
ALTER TABLE Funcionario ADD FOREIGN KEY(idSupervisor) REFERENCES Funcionario  
(idFuncionario);  
ALTER TABLE Tarefa ADD FOREIGN KEY(idProjeto) REFERENCES Projeto (idProjeto);  
ALTER TABLE Projeto ADD FOREIGN KEY(idCategoria) REFERENCES Categoria  
(idCategoria);  
ALTER TABLE Projeto ADD FOREIGN KEY(idEquipe) REFERENCES Equipe (idEquipe);  
ALTER TABLE Documentacao ADD FOREIGN KEY(idFuncionario) REFERENCES Funcionario  
(idFuncionario);
```

```
ALTER TABLE DocumentoProjeto ADD FOREIGN KEY(idProjeto) REFERENCES Projeto
(idProjeto);
ALTER TABLE DocumentoProjeto ADD FOREIGN KEY(idDocumentacao) REFERENCES
Documentacao (idDocumento);
ALTER TABLE TarefaAlocada ADD FOREIGN KEY(idFuncionario) REFERENCES Funcionario
(idFuncionario);
ALTER TABLE TarefaAlocada ADD FOREIGN KEY(idTarefa) REFERENCES Tarefa (idTarefa);
ALTER TABLE FuncionarioEquipe ADD FOREIGN KEY(idFuncionario) REFERENCES
Funcionario (idFuncionario);
ALTER TABLE FuncionarioEquipe ADD FOREIGN KEY(idEquipe) REFERENCES Equipe
(idEquipe);
```