

## Documentação do Projeto Service Desk

- **Resumo do Projeto:**

O projeto ServiceDesk é uma aplicação em C# que visa automatizar o processo de abertura de chamados e gerenciamento de tickets de suporte em uma organização. Ele oferece um sistema completo para categorizar, priorizar e resolver problemas relatados por usuários, permitindo uma melhor organização e eficiência na equipe de suporte ao incluir informações sobre dispositivos, funcionários, departamentos, tempo de duração e, também, possibilitando o retorno dos usuários em relação ao suporte fornecido.

- **Principais Funcionalidades:**

1. **Abertura de Chamados:** Usuários podem abrir novos chamados descrevendo os problemas que estão enfrentando.
2. **Gerenciamento de Tickets:** A equipe de suporte pode visualizar, atualizar e atribuir tickets para resolver os problemas.
3. **Categorização:** Tickets podem ser categorizados com base em categorias, prioridades e status.
4. **Monitoramento de SLA:** O sistema permite o acompanhamento do tempo de resposta e resolução de acordo com os Acordos de Nível de Serviço (SLAs).
5. **Registro de Atividades:** O sistema registra todas as atividades relacionadas a tickets, incluindo atualizações, feedbacks e soluções.
6. **Filtragem Avançada:** Os usuários podem filtrar tickets com base em diversos critérios para encontrar informações específicas.
7. **Gestão de Usuários:** Administradores podem criar e gerenciar contas de usuários.

- **Classes Model:**

1. **Categoria.cs:**

- 1.1. **Descrição:** Representa uma categoria que pode ser atribuída a um ticket.
- 1.2. **Atributos:** 'ID': Identificador único da categoria; 'Nome': O nome da categoria.

2. **CentroDeCusto**

- 2.1. **Descrição:** Representa um centro de custo para o qual os custos podem ser atribuídos.
- 2.2. **Atributos:** 'ID': Identificador único do centro de custo; 'Nome': O nome do centro de custo; 'Usuario': Usuário atrelado ao centro de custo.

3. **Dispositivo.cs:**

- 3.1. **Descrição:** Representa um dispositivo ou equipamento que pode ser relacionado a um chamado ou ticket de suporte (vinculado a um usuário).
- 3.2. **Atributos:** 'ID': Identificador único do dispositivo; 'Nome': O nome ou modelo do dispositivo.

4. Feedback.cs:

- 4.1. Descrição: Representa o feedback fornecido pelos usuários em relação ao atendimento ou à resolução de um ticket.
- 4.2. Atributos: 'ID': Identificador único do feedback; 'FeedbackText': Comentário ou descrição do feedback (realizada pelo usuário); 'Solucaold': O identificador único da solução atrelada ao feedback; 'Solucao': A solução que está atribuída ao ticket ao qual o feedback será enviado.

5. Filtro.cs:

- 5.1. Descrição: Representa um filtro que pode ser aplicado para pesquisar e filtrar tickets de suporte.
- 5.2. Atributos: 'ID': Identificador único do filtro; 'Nome': O nome do filtro.

6. Funcionario.cs:

- 6.1. Descrição: Representa um funcionário da equipe de suporte ou da organização.
- 6.2. Atributos: 'ID': Identificador único do funcionário; 'Nome': O nome do funcionário; 'Cargo': O cargo ou função do funcionário; 'Email': O email do funcionário; 'Senha': A senha do usuário.

7. Prioridade.cs:

- 7.1. Descrição: Representa a prioridade de um ticket de suporte.
- 7.2. Atributos: 'ID': Identificador único da prioridade; 'Nome': O nome da prioridade (por exemplo, Alta, Média, Baixa).

8. Sla.cs:

- 8.1. Descrição: Representa um Acordo de Nível de Serviço (SLA) que define os prazos de atendimento e resolução para tickets de suporte.
- 8.2. Atributos: 'ID': Identificador único do SLA; 'Nome': O nome do SLA; 'Tickets': Os tickets aos quais o SLA se conecta.

9. Solucao.cs:

- 9.1. Descrição: Representa a solução ou resolução de um ticket de suporte.
- 9.2. Atributos: 'ID': Identificador único da solução; 'DescSolucao': A descrição da solução.

10. Status.cs:

- 10.1. Descrição: Representa o status atual de um ticket de suporte.
- 10.2. Atributos: 'ID' Identificador único do status; 'Nome': O nome do status (por exemplo, Aberto, Em Andamento, Resolvido).

11. Ticket.cs:

- 11.1. Descrição: Representa um ticket de suporte ou chamado aberto pelos usuários.
- 11.2. Atributos: 'ID': Identificador único do ticket; 'Titulo': O título ou assunto do ticket; 'Descricao': A descrição detalhada do problema ou solicitação; 'DataAbertura': A data em que o ticket foi aberto; 'Status': O status atual do ticket; 'Prioridade': A prioridade do ticket; 'Categoria': A categoria atribuída ao ticket; 'Dispositivo': O dispositivo relacionado ao ticket; 'FuncionarioResponsavel': O funcionário responsável pelo ticket; 'UsuarioCriador': O usuário que criou o ticket.

12. Usuario.cs:

- 12.1. Descrição: Representa um usuário que pode abrir tickets de suporte.
- 12.2. Atributos: 'ID': Identificador único do usuário; 'Nome': O nome do usuário; 'Email': O endereço de email do usuário; 'CentroDeCusto': O departamento ao qual o

usuário está associado; 'Dispositivo': O dispositivo que pertence ao usuário; 'CentroDeCustold': O identificador único do departamento; 'Usuariold': O identificador único do usuário

- Classes Controller:

1. CategoriaController

- 1.1. Descrição: Controla as operações relacionadas a categorias.

- 1.2. Principais Métodos: setCategoria(); associarCategoriaAoTicket(); getCategoria(); excluirCategoria();

2. CentroDeCustoController

- 2.1. Descrição: Controla as operações relacionadas a centros de custo.

- 2.2. Principais Métodos: associarCategoriaAoTicket(); associarUsuarioAoCentroDeCusto(); listarCentrosDeCusto();

3. Dispositivo

- 3.1. Descrição: Controla as operações relacionadas a feedbacks dos usuários.

- 3.2. Principais Métodos: cadastrar(); listar(); buscar(); alterar();

4. Feedback

- 4.1. Descrição: Controla as operações relacionadas a feedbacks dos usuários.

- 4.2. Principais Métodos: getFeedback(); setFeedbackAssociarASolucao();

5. Filtro

- 5.1. Descrição: Controla as operações de criação e gerenciamento de filtros para pesquisa de tickets.

- 5.2. Principais Métodos: setFiltro(); getFiltro();

6. Funcionario

- 6.1. Descrição: Controla as operações relacionadas a funcionários da equipe de suporte.

- 6.2. Principais Métodos: cadastrar(); fazerLogin(); atualizarPerfil(); visualizarTickets(); listarFuncionarios();

7. Prioridade

- 7.1. Descrição: Gerencia as operações relacionadas a prioridades de tickets.

- 7.2. Principais Métodos: getPrioridade(); setPrioridade(); associarPrioridadeAoTicket();

8. Sla

- 8.1. Descrição: Controla as operações relacionadas aos Acordos de Nível de Serviço (SLAs).

- 8.2. Principais Métodos: getSla(); setSla(); associarSlaAoTicket();

9. Solucao

- 9.1. Descrição: Gerencia as operações relacionadas a soluções ou resoluções de tickets.

- 9.2. Principais Métodos: getSolucao(); setSolucao(); associarSolucaoAoTicket(); associarCategoriaAoTicket();

10. Status

- 10.1. Descrição: Controla as operações relacionadas aos status de tickets.

- 10.2. Principais Métodos: getStatus(); setStatus(); associarStatusAoTicket(); excluirStatus();

11. Ticket

- 11.1. Descrição: Gerencia as operações relacionadas a tickets de suporte.

11.2. Principais Métodos: criarTicket(); atribuirTicket(); getTicket(); listarTickets();  
excluirTicket();

## 12. Usuario

12.1. Descrição: Controla as operações relacionadas a usuários que podem abrir tickets.

12.2. Principais Métodos: cadastrar(); buscar(); alterar(); listarUsuarios(); excluir();

- Relatório de Atividades com o ChatGPT:

Neste projeto, o ChatGPT foi utilizado para auxiliar nas seguintes atividades:

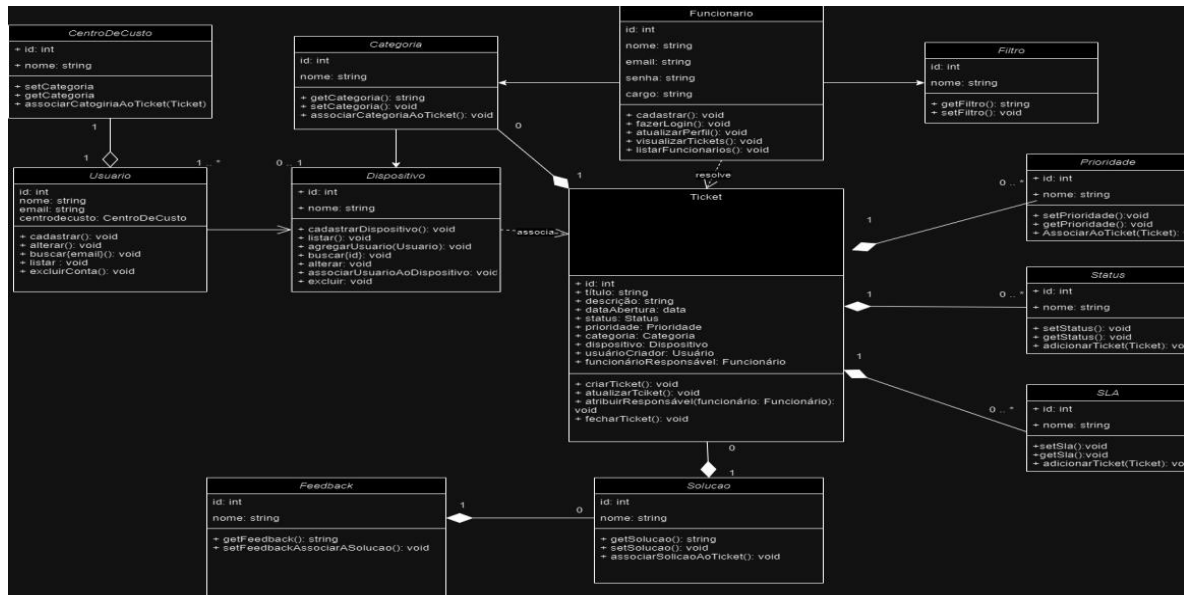
- Definição de Requisitos: O ChatGPT ajudou na elaboração dos requisitos iniciais do projeto, sugerindo funcionalidades e recursos.

- Design do Diagrama de classes: O modelo auxiliou na criação do esquema do diagrama de classes, recomendando tabelas e relacionamentos.

- Solução de Desafios de Implementação: Durante a implementação, a inteligência artificial foi usada para resolver problemas de codificação e otimização de código.

- Diagrama de Classes UML:

A seguir, apresentamos o diagrama de classes UML que representa as principais classes do projeto:



Obs: arquivo jpeg salvo separadamente no diretório do projeto.