

# DOCUMENTO DE ESPECIFICAÇÃO TÉCNICA: SISTEMA DE HELPDESK ENTERPRISE

## SUMÁRIO

|  |   |
|--|---|
| <b>1. VISÃO GERAL DA ARQUITETURA</b>                 | 2 |
| 1.1. ESTRUTURA DA SOLUÇÃO                            | 2 |
| <b>2. MODELAGEM DE DADOS</b>                         | 2 |
| 2.1. NÚCLEO DE ACESSO E ORGANIZAÇÃO                  | 3 |
| 2.2. NÚCLEO DE CATÁLOGO E WORKFLOW                   | 3 |
| 2.3. NÚCLEO OPERACIONAL (TICKET)                     | 4 |
| <b>3. DESCRIÇÃO DOS PROCESSOS CHAVE</b>              | 4 |
| 3.1. PROCESSO DE CRIAÇÃO E ROTEAMENTO                | 4 |
| 3.2. PROCESSO DE TRANSIÇÃO LINEAR (AVANÇAR)          | 5 |
| 3.3. PROCESSO DE REDIRECIONAMENTO (EXCEÇÃO)          | 5 |
| <b>4. CRONOGRAMA DE IMPLEMENTAÇÃO (USER STORIES)</b> | 5 |
| FASE 1: FUNDAÇÃO E CADASTROS (BACKOFFICE)            | 5 |
| FASE 2: O MOTOR DE WORKFLOW (O DIFERENCIAL)          | 6 |
| FASE 3: OPERAÇÃO DE TICKETS (CORE BUSINESS)          | 6 |
| FASE 4: CICLO DE VIDA E SLA                          | 7 |
| FASE 5: EXCEÇÕES (IMPLEMENTAÇÃO FUTURA)              | 7 |

## 1. VISÃO GERAL DA ARQUITETURA

Adotaremos a **Clean Architecture** para garantir desacoplamento e testabilidade. A solução será dividida em projetos (assemblies) dentro do Visual Studio:

### 1.1. ESTRUTURA DA SOLUÇÃO

#### 1. HDE.Domain (Core):

- Contém as Entidades, Enums, Interfaces de Repositório e Regras de Negócio Puras (Domain Services).
- *Dependências*: Nenhuma.

#### 2. HDE.Application (Orquestração):

- Implementa os Casos de Uso (CQRS com MediatR é recomendado aqui).
- Contém DTOs, Mappers (AutoMapper) e Validações (FluentValidation).
- *Dependências*: Domain.

#### 3. HDE.Infra.Data (Persistência):

- Implementação do Entity Framework Core, Contexto do Banco, Mapeamentos e Repositórios.
- *Dependências*: Domain.

#### 4. HDE.Infra.ioC (Injeção de Dependência):

- Centraliza a configuração do container de DI.
- *Dependências*: Application, Infra.Data.

#### 5. HDE.API (Entrada):

- Controllers RESTful, Middlewares de Tratamento de Erro, Configuração de JWT.
- *Dependências*: Infra.ioC.

## 2. MODELAGEM DE DADOS

Abaixo, o desenho das entidades fundamentais para atender aos requisitos de Filas Dinâmicas, Isolamento Estrito e SLA por Etapa.

## 2.1. NÚCLEO DE ACESSO E ORGANIZAÇÃO

**ApplicationUser** (Extends IdentityUser):

- FullName (string)
- IsActive (bool)
- **ServiceQueue** (As Filas de Atendimento):
  - Id (Guid)
  - Name (string) - Ex: "N1 - Infra", "Financeiro - Pagamentos"
  - Description (string)
- **QueueMember** (Tabela de Junção - Segurança):
  - QueueId (FK)
  - UserId (FK)
  - *Regra*: Um usuário só pode visualizar/atuar em tickets que estão em filas onde ele é membro.

## 2.2. NÚCLEO DE CATÁLOGO E WORKFLOW

- **ServiceCategory**:
  - Name (string) - Ex: "Hardware"
- **ServiceSubcategory**:
  - Name (string) - Ex: "Solicitação de Notebook"
  - CategoryId (FK)
  - DefaultWorkflowId (FK) -> O "**Cérebro**" do roteamento.
- **Workflow**:
  - Name (string)
  - Version (int)
- **WorkflowStep** (As Etapas do Processo):
  - WorkflowId (FK)
  - TargetQueueId (FK) -> Define para qual fila o ticket vai nesta etapa.
  - StepName (string) - Ex: "Aprovação Gestor", "Preparação Técnica".

- SLAInHours (int) -> **Define o tempo limite desta etapa específica.**
- OrderIndex (int) -> Para definir a sequência linear (1, 2, 3...).

## 2.3. NÚCLEO OPERACIONAL (TICKET)

- **Ticket:**

- ProtocolNumber (string/long) - Gerado automaticamente.
- Title, Description
- RequesterId (FK User)
- **Controle de Estado:**
  - CurrentStepId (FK WorkflowStep) -> Onde estou no fluxo?
  - CurrentQueueId (FK ServiceQueue) -> Em qual fila estou? (Desnormalizado para performance de filtro).
  - CurrentAssignedUserId (FK User, Nullable) -> Quem pegou o ticket?
  - Status (Enum: New, InProgress, OnHold, Resolved, Closed). **Controle de SLA:**
  - CurrentStepEntryDate (DateTime) -> Quando entrou nesta etapa.
  - CurrentStepDeadline (DateTime) -> Calculado: EntryDate + WorkflowStep.SLAInHours.

- **TicketHistory:**

- Log imutável de todas as trocas de etapa e alterações.

## 3. DESCRIÇÃO DOS PROCESSOS CHAVE

### 3.1. PROCESSO DE CRIAÇÃO E ROTEAMENTO

1. Usuário seleciona Categoria e Subcategoria.
2. Sistema busca a ServiceSubcategory e identifica o DefaultWorkflowId.
3. Sistema carrega o **primeiro** WorkflowStep desse Workflow (OrderIndex = 1).

4. Ticket é salvo com:

- CurrentStepId = Passo 1.
- CurrentQueueId = Fila definida no Passo 1.
- CurrentStepEntryDate = Agora.
- CurrentStepDeadline = Agora + SLA do Passo 1.

5. Resultado: O ticket "aparece magicamente" na fila correta sem intervenção humana.

### 3.2. PROCESSO DE TRANSIÇÃO LINEAR (AVANÇAR)

1. Técnico clica em "Concluir Etapa/Avançar".
2. Sistema busca o próximo WorkflowStep (OrderIndex + 1).
  - *Se existir*: Atualiza o Ticket para o novo Step, nova Fila (Queue) e recalcula o SLA (CurrentStepDeadline).
  - *Se não existir*: O fluxo acabou. Ticket muda status para Resolved.

### 3.3. PROCESSO DE REDIRECIONAMENTO (EXCEÇÃO)

1. Técnico aciona função "Transferir".
2. Seleciona uma ServiceQueue de destino manualmente.
3. O sistema mantém o CurrentStepId (ou move para um step genérico de "Análise") mas altera forçosamente o CurrentQueueId.
4. Gera log em TicketHistory indicando "Transferência Manual".

## 4. CRONOGRAMA DE IMPLEMENTAÇÃO (USER STORIES)

Organizado para que você tenha um sistema funcional o mais rápido possível (MVP incremental).

### FASE 1: FUNDAÇÃO E CADASTROS (BACKOFFICE)

*O sistema precisa existir antes de operar.*

1. **[US-001] Setup da Arquitetura**: Criar a Solution .NET com as camadas (Domain, Application, Infra, API), configurar EF Core com SQL Server e Swagger.

2. **[US-002] Gestão de Filas (Queues):** Como Admin, quero criar Filas de Atendimento (ex: N1, N2, RH) para segregar os tickets.
3. **[US-003] Gestão de Usuários e Permissões:** Como Admin, quero cadastrar usuários e vinculá-los a uma ou mais Filas (QueueMember), para garantir o isolamento de acesso.
4. **[US-004] Catálogo de Serviços:** Como Admin, quero cadastrar Categorias e Subcategorias para organizar o menu de solicitação.

## FASE 2: O MOTOR DE WORKFLOW (O DIFERENCIAL)

*Aqui implementamos a inteligência do sistema.*

5. **[US-005] Criação de Workflow:** Como Admin, quero criar um cabeçalho de Workflow (Nome/Versão) e associá-lo a uma Subcategoria.
6. **[US-006] Definição de Etapas (Steps):** Como Admin, quero adicionar etapas sequenciais a um Workflow, definindo para cada etapa: Nome, Fila Responsável e SLA em horas.

## FASE 3: OPERAÇÃO DE TICKETS (CORE BUSINESS)

*Onde o usuário final interage.*

7. **[US-007] Abertura de Chamado (Roteamento):** Como Usuário, quero abrir um chamado selecionando uma Subcategoria, para que o sistema o encaminhe automaticamente para a Fila da primeira etapa do fluxo configurado.
8. **[US-008] Minha Fila de Trabalho:** Como Técnico, quero ver uma lista de tickets APENAS das filas as quais pertenço (QueueMember), ordenados por SLA (mais urgente primeiro).
9. **[US-009] Detalhes do Ticket:** Como Técnico, quero visualizar os dados do ticket, sabendo em qual etapa ele está e qual o prazo (SLA) para aquela etapa.
10. **[US-010] Atribuição (Self-Assign):** Como Técnico, quero atribuir um ticket da fila para mim (Assign to me), sinalizando que estou trabalhando nele.

## FASE 4: CICLO DE VIDA E SLA

11. **[US-011] Avanço de Etapa:** Como Técnico, quero "Avançar" o ticket para que ele vá para a próxima etapa do fluxo (e próxima fila responsável), recalculando o SLA automaticamente.
12. **[US-012] Encerramento:** O sistema deve detectar quando não há mais etapas e marcar o ticket como "Resolvido".
13. **[US-013] Monitor de SLA:** Como Gestor, quero visualizar visualmente (ex: cor vermelha) tickets que estouraram o CurrentStepDeadline.

## FASE 5: EXCEÇÕES (IMPLEMENTAÇÃO FUTURA)

14. **[US-014] Redirecionamento Manual:** Como Técnico, quero transferir um ticket para outra fila manualmente, ignorando o fluxo padrão, caso o ticket tenha sido classificado errado.