# **ZELOS**

# **Documento de Requisitos do Sistema (DRS)**

## **1. Introdução**

### **1.1 Objetivo**

Este documento tem como objetivo descrever os requisitos necessários para o desenvolvimento do sistema **Zelos**, que será responsável por controlar os chamados de manutenção e suporte dentro da escola SENAI Armando de Arruda Pereira, utilizando o número de patrimônio dos itens da escola como identificador principal para os chamados.

### **1.2 Escopo**

O sistema **Zelos** visa gerenciar a solicitação, acompanhamento e resolução de chamados relacionados aos itens da escola, como móveis, equipamentos e outros recursos, identificados pelo número de patrimônio. O sistema permitirá a criação, atribuição, atualização e fechamento de chamados, fornecendo visibilidade aos responsáveis.

### **1.3 Definições, Acrônimos e Abreviações**

* **Patrimônio**: Código único que identifica um item específico da escola.
* **Chamado**: Solicitação de manutenção ou suporte relacionado a um item da escola.
* **Administrador**: Usuário com privilégios de gerenciamento e configuração do sistema.

## **2. Visão Geral do Sistema**

### **2.1 Funcionalidades**

A seguir, estão as funcionalidades principais que o sistema deverá suportar:

* **Criação de Chamados**: Registrar novos chamados a partir do número de patrimônio.
* **Acompanhamento de Chamados**: Acompanhar o status de chamados criados, com possibilidade de atualizações.
* **Fechamento de Chamados**: Encerrar chamados quando a solicitação for atendida.
* **Relatórios**: Gerar relatórios com o histórico de chamados e manutenção de itens.

### **2.2 Usuários e Permissões**

O sistema será utilizado pelos seguintes tipos de usuários:

* **Administrador**: Pode criar, atualizar, atribuir, fechar chamados e gerar relatórios.
* **Tecnico**:
* **Usuário Comum**: Pode criar chamados e visualizar o status de chamados associados ao número de patrimônio que gerou a solicitação.

### **2.3 Arquitetura do Sistema**

O sistema será baseado na arquitetura **cliente-servidor**, utilizando as seguintes tecnologias:

* **Frontend**: Framework javascript (React.JS, Next.js, etc.) para, com design responsivo.
* **Backend**: Node.js, com Express.
* **Banco de Dados**: MySQL.

## **3. Requisitos Funcionais**

### **3.1 Criação de Chamados**

* **Descrição**: O sistema deve permitir a criação de novos chamados informando o número de patrimônio do item ou, caso não seja possível identificar pelo patrimônio, se faz necessário uma descrição específica do item.
* **Entrada**:
  + Número de patrimônio ou descrição de item
  + Descrição do problema
  + Tipo de chamado (ex: manutenção, apoio técnico, etc.)
* **Saída**: Chamado criado com ID único e status inicial “aberto”.
* **Regras de Negócio**:
  + Cada número de patrimônio pode gerar múltiplos chamados, mas um mesmo número de patrimônio não pode estar associado a dois chamados de mesmo tipo simultaneamente.
  + Caso o usuário tente gerar um chamado do mesmo tipo para o mesmo número de patrimônio, ele será alertado de que já existe um chamado aberto para aquele tipo.

**3.2 Atribuição de Chamados**

* **Descrição:** O sistema deve permitir que os chamados sejam atribuídos aos técnicos responsáveis pela resolução. Os técnicos terão acesso a um pool de chamados e poderão se autoatribuir os tickets disponíveis para os quais têm competência e disponibilidade para resolver.
* **Entrada:**
  + ID do chamado
  + ID do técnico
* **Saída:** Chamado atualizado com o técnico atribuído.
* **Regras de Negócio:**
  + Apenas administradores ou técnicos podem atribuir chamados.
  + Um chamado só pode ser atribuído a um técnico por vez.

### **3.3 Apontamentos**

* **Descrição**: Deve permitir que os técnicos registrem informações detalhadas sobre o andamento de cada chamado. Cada técnico poderá adicionar apontamentos sobre o serviço que está sendo realizado, incluindo a descrição do trabalho, horários de início e fim, e quaisquer observações relevantes.
* **Entrada**:
  + **Chamado**: ID do chamado ao qual o apontamento será vinculado.
  + **Técnico**: ID do técnico que está registrando o apontamento.
  + **Descrição**: Texto explicativo sobre o que foi feito ou observado durante o atendimento.
  + **Começo**: Hora e data de início do serviço.
  + **Fim**: Hora e data de término do serviço.
* **Saída**:
  + Um apontamento registrado, contendo:
    - ID do apontamento.
    - Chamado ao qual está vinculado.
    - Técnico responsável.
    - Descrição do serviço realizado.
    - Horário de início e término.
    - Duração do serviço (calculada automaticamente a partir do início e fim).
    - Data e hora do registro.
* **Regras de Negócio**:
  + Cada chamado pode ter múltiplos apontamentos durante sua execução, com cada apontamento associado a um técnico específico.
  + A duração do serviço será calculada automaticamente com base nos horários de **começo** e **fim** (em horas ou minutos, conforme a configuração).
  + O técnico só pode adicionar apontamentos a chamados que estejam atribuídos a ele e que estejam no status "em andamento".
  + Um apontamento não pode ser editado após ser salvo. No entanto, o técnico pode adicionar novos apontamentos ao longo do tempo.

### **3.4 Acompanhamento de Chamados**

* **Descrição**: O usuário poderá acompanhar os chamados que gerou, verificando o status e as atualizações feitas.
* **Entrada**: ID do chamado ou número de patrimônio.
* **Saída**: Status atual do chamado (ex: "Em andamento", "Concluído").
* **Regras de Negócio**: O status do chamado só pode ser alterado por um administrador ou técnico responsável.

### **3.5 Fechamento de Chamados**

* **Descrição**: O administrador ou técnico responsável poderá fechar um chamado quando o problema for resolvido.
* **Entrada**: ID do chamado, resolução do problema.
* **Saída**: Chamado fechado com data de resolução.
* **Regras de Negócio**: O chamado só pode ser fechado se o status estiver como "Em andamento" ou "Aguardando aprovação".

### **3.6 Seção de Administrador**

* **Descrição**: A seção de administrador permite que os administradores do sistema gerenciem os chamados, técnicos e tipos de chamados. Além disso, os administradores terão acesso a uma **tabela de chamados**, onde podem visualizar todos os chamados em aberto, em andamento e concluídos, bem como filtrar, editar e encerrar chamados. A seção também permitirá a **geração de relatórios básicos** sobre o status e histórico dos chamados e das atividades dos técnicos.

#### **3.6.1 Administração de Chamados via Tabela**

* **Descrição**: Na tabela de chamados, o administrador pode visualizar todos os chamados registrados no sistema, com a capacidade de realizar ações como **editar**, **fechar** e **atribuir técnicos**. A tabela oferece um painel de controle eficiente para a gestão dos chamados em diferentes status.
* **Entrada**:
  + **Filtro**: O administrador pode filtrar os chamados por status (aberto, em andamento, concluído), tipo de chamado, data de criação, técnico atribuído, etc.
  + **Ação**: O administrador pode editar os detalhes de um chamado ou atribuir um técnico.
  + **Fechar Chamado**: O administrador pode fechar um chamado, marcando-o como concluído.
* **Saída**:
  + **Tabela de Chamados**: A tabela exibe as informações de todos os chamados cadastrados, com as seguintes colunas:
    - ID do chamado
    - Título do chamado
    - Tipo de chamado
    - Status (aberto, em andamento, concluído)
    - Técnico atribuído
    - Data de criação
    - Data de fechamento (quando aplicável)
    - Ações (editar, fechar, atribuir técnico)
* **Regras de Negócio**:
  + Os administradores podem visualizar todos os chamados no sistema, independentemente de serem associados a técnicos específicos.
  + O administrador pode atribuir técnicos aos chamados que estão no status "aberto".
  + Chamados que estão no status "em andamento" ou "aguardando aprovação" podem ser fechados pelo administrador.
  + O administrador pode editar qualquer campo de um chamado, exceto o número de patrimônio (que é único e imutável após a criação).

#### **3.6.2 Geração de Relatórios Básicos**

* **Descrição**: O sistema deve permitir que os administradores gerem relatórios básicos sobre os chamados e as atividades dos técnicos. Esses relatórios ajudarão na análise de performance, acompanhamento de atividades e na tomada de decisões sobre a gestão de recursos.
* **Entrada**:
  + **Filtro de Relatório**: O administrador pode selecionar filtros para os relatórios, como período de tempo, tipo de chamado, status do chamado, técnico atribuído, etc.
  + **Tipo de Relatório**: Relatórios disponíveis podem incluir:
    - **Relatório de Chamados por Status**: Mostra o número de chamados em cada status (aberto, em andamento, concluído) dentro de um período específico.
    - **Relatório de Chamados por Tipo**: Exibe a distribuição dos chamados por tipo (manutenção, apoio técnico, etc.).
    - **Relatório de Atividades dos Técnicos**: Mostra os chamados atribuídos a cada técnico, o tempo médio de resolução e o status atual.
* **Saída**:
  + Relatórios em formato de **tabela** ou **gráfico** (ex: gráfico de barras ou pizza) para visualização dos dados.
  + O administrador pode exportar os relatórios em **formato CSV** ou **PDF** para análise e arquivamento.
* **Regras de Negócio**:
  + O sistema deve permitir que o administrador gere relatórios com base nos dados disponíveis no banco de dados, incluindo chamados e técnicos.
  + Relatórios podem ser gerados para qualquer período de tempo, com a possibilidade de filtrar por status, tipo de chamado, técnico, entre outros critérios.
  + O sistema deve garantir que apenas os administradores possam acessar a funcionalidade de geração de relatórios.

**4. Requisitos Não Funcionais**

### **4.1 Performance**

* O sistema deverá ser capaz de processar até **500 chamados** simultaneamente.
* O tempo de resposta para criação e atualização de chamados será de no máximo **2 segundos**.

### **4.2 Segurança**

* O sistema deve garantir a **autenticação** segura dos usuários, utilizando **JWT** para sessões de usuário.
* Todos os dados sensíveis, como informações pessoais e detalhes de chamados, devem ser **criptografados**.

### **4.3 Usabilidade**

* O sistema deve ser intuitivo e de fácil navegação para usuários com pouca experiência técnica.
* O layout deve ser responsivo, com adaptações para dispositivos móveis e desktops.

### **4.4 Disponibilidade**

* O sistema deve ter uma disponibilidade de **99%** durante o horário de funcionamento da escola (**segunda a sexta-feira, das 8h às 18h**).

## **5. Requisitos Técnicos**

### **5.1 Tecnologia**

O sistema será desenvolvido utilizando as seguintes tecnologias:

* **Frontend**: Next.js
* **Backend**: Node.js (Express)
* **Banco de Dados**: MySQL

### **5.2 Desempenho e Escalabilidade**

* O sistema deverá ser capaz de escalabilidade horizontal para lidar com picos de uso, garantindo a continuidade da operação durante o horário de maior tráfego.

## **6. Restrições**

### **6.1 Tecnológicas**

* O sistema deverá ser desenvolvido para **plataforma web** e deverá ser compatível com os navegadores **Google Chrome** e **Mozilla Firefox**.

### **6.3 Legais**

* O sistema deve estar em conformidade com a **LGPD** (Lei Geral de Proteção de Dados Pessoais).

## **7. Critérios de Aceitação**

### **7.1 Critérios de Aceitação de Funcionalidades**

Cada funcionalidade será considerada aceita se:

* A criação, acompanhamento e fechamento de chamados estiverem funcionando corretamente.
* O sistema apresentar desempenho adequado mesmo com o volume de dados previstos.
* A segurança e integridade dos dados forem mantidas.

### **7.2 Critérios de Aceitação do Sistema**

* O sistema será considerado aceito quando atender aos requisitos descritos neste documento e todas as funcionalidades estiverem operacionais sem falhas críticas.

## **8. Anexos**

### **8.1 Diagramas**

* **Diagrama de Arquitetura**: [Incluir diagrama aqui]

### **8.2 Glossário**

* **Patrimônio**: Código único que identifica um item da escola.
* **Chamado**: Solicitação de manutenção ou suporte relacionado ao patrimônio.