Senac Tech

Planejar e desenvolver websites utilizando o framework Django

**PROJETO SAC SENAC**

**PORTO ALEGRE, JULHO 2024**



Senac Tech

**PROJETO SISTEMA SAC SENAC**

Nome: Matheus Brizola

Turma: RSTI 140054

Professor: Lucas Matheus Peres

**PORTO ALEGRE, JULHO 2024**

Sumário

[1Introdução 3](#_Toc169694505)

[2 Objetivos 4](#_Toc169694506)

[2.1 Objetivo Principal 4](#_Toc169694507)

[2.2 Objetivos Específicos 4](#_Toc169694508)

[3 Regras de Negócio (RN) 5](#_Toc169694509)

[4 Requisitos 5](#_Toc169694510)

[4.1 Requisitos Funcionais (RF) 5](#_Toc169694511)

[4.2 Requisitos Não Funcionais (RNF) 6](#_Toc169694512)

[5 Diagramas de UML 7](#_Toc169694513)

[5.1 Diagramas de caso de uso 7](#_Toc169694515)

[5.2 Diagramas de classe 7](#_Toc169694516)

[6 Banco 9](#_Toc169694518)

[6.1 Modelo Físico 11](#_Toc169694521)

[7 Telas Web 12](#_Toc169694522)

[8 Conclusão 23](#_Toc169694523)

# 1Introdução

A implementação de um sistema de controle de atendimento de serviços de atendimento ao consumidor (SAC) é necessário para gerar uma boa comunicação entre o prestador de serviços e seus consumidores usando o processo de feedback e solicitações. Este sistema centraliza e organiza todo o contato dos estudantes com o administrativo da instituição de ensino além de salvar e poder visualizar as respostas que os funcionários enviaram. A solicitação de atendimento é o começo do processo. Todo atendimento contém o nome do solicitante, e-mail para contato, o a categoria do assunto e a mensagem do solicitante descrevendo o motivo de contato. Os colaboradores do setor administrativo por sua vez podem visualizar o chamado e atender o mais breve possível, deixando um resumo do atendimento salvo no sistema para futura consulta.

# 2 Objetivos

## 2.1 Objetivo Principal

O objetivo principal de um sistema de serviços de atendimento ao consumidor de um instituto de educação é **centralizar e gerenciar as interações e solicitações dos alunos, garantindo um atendimento eficiente e ágil. Ele facilita a comunicação entre alunos e a instituição, resolve problemas rapidamente e melhora a satisfação dos alunos**. Este sistema visa:

**Aumentar a eficiência e produtividade** da equipe administrativa, permitindo uma alocação mais eficaz dos recursos e priorização das demandas mais urgentes.

**Melhorar a satisfação dos alunos**, proporcionando uma comunicação clara e transparente sobre o status de suas solicitações e garantindo uma resolução rápida e eficaz dos problemas.

**Registrar e documentar todas as interações e soluções**, demonstrando transparência do processo de atendimento.

## 2.2 Objetivos Específicos

**1. Atender e resolver solicitações de forma eficiente:**

* **Agilizar o registro e a triagem de solicitações:** Permite que os usuários façam solicitações facilmente e categorize-os automaticamente para o assunto correto.
* **Facilitar a comunicação e colaboração:** Oferece ferramentas para que secretários colaborem entre si e com os solicitantes, compartilhem informações e atualizem o status das solicitações de forma transparente.

**2. Melhorar a experiência do usuário:**

* **Proporcionar atendimento:** Permita que os usuários abram e acompanhem solicitações.
* **Comunicar-se de forma proativa:** Mantenha os solicitantes informados sobre o andamento dos seus chamados por meio de notificações por e-mail ou dentro do próprio sistema.
* **Coletar feedback:** Solicite feedback dos usuários após a resolução dos chamados para identificar áreas de melhoria e aprimorar o atendimento.

**3. Reduzir custos e aumentar a produtividade:**

* **Proteger contra tempo de inatividade:** Priorize solicitações que impactam a instituição e resolva-os rapidamente para minimizar o tempo de inatividade.

**5. Obter insights acionáveis:**

* **Coletar e analisar dados de solicitações:** Armazene e analise dados de solicitações, como tempo de resolução e tipo de problema para identificar tendências e tomar decisões estratégicas.
* **Identificar oportunidades de melhoria:** Utilize a análise de dados para identificar áreas onde o serviço pode ser aprimorado e implementar mudanças estratégicas.
* **Tomar decisões baseadas em dados:** Utilize insights obtidos da análise de dados para tomar decisões mais informadas sobre alocação de recursos, priorização de projetos e investimentos em infraestrutura.

# 3 Regras de Negócio (RN)

Regras de negócio são diretrizes que definem e limitam operações e processos dentro de uma organização, assegurando consistência, eficiência e conformidade com políticas internas, leis e objetivos estratégicos. Elas cobrem aspectos como procedimentos operacionais, gestão de dados e atividades comerciais

#### 1. Abertura de Chamados

* **Título do Chamado**: Deve descrever brevemente o problema (ex.: “Funcionário não consegue acessar a internet”).
* **Descrição do Chamado**: Deve incluir detalhes sobre as tentativas de solução já realizadas (ex.: “O funcionário tentou reiniciar o computador e o roteador, mas o problema persiste”).
* **Prioridade**: Deve ser definida com base na urgência e impacto do problema (ex.: Alta, Média, Baixa).
* **Solicitante**: Nome do funcionário que está reportando o problema.
* **Departamento**: Nome do departamento ao qual o funcionário pertence.

#### 2. Atribuição de Chamados

* **Atribuição Automática**: O sistema deve atribuir automaticamente os chamados aos técnicos de TI mais qualificados, considerando a urgência, complexidade do problema e disponibilidade dos técnicos.

#### 3. Solução de Problemas

* **Verificação de Conexões**: O técnico deve verificar se o computador está conectado corretamente ao roteador.
* **Configurações de Rede**: O técnico deve verificar se as configurações de rede do computador estão corretas.
* **Reinício do Modem**: O técnico deve tentar reiniciar o modem.
* **Intervenção Presencial**: Se nenhuma das etapas anteriores resolver o problema, o técnico deve ir até a mesa do funcionário para diagnosticar e resolver o problema.

#### 4. Rastreamento e Acompanhamento

* **Status em Tempo Real**: O sistema deve permitir que usuários e técnicos acompanhem o status dos chamados em tempo real.
* **Atualizações e Comentários**: O sistema deve registrar todas as etapas de resolução, atualizações e comentários.

#### 5. Gerenciamento de Conhecimento

* **Base de Conhecimento**: O sistema deve facilitar a criação e o compartilhamento de uma base de conhecimento com soluções para problemas comuns.

#### 6. Relatórios e Análises

* **Geração de Relatórios**: O sistema deve gerar relatórios sobre o volume de chamados, tempo de resolução, satisfação dos usuários e outros indicadores importantes.
* **Análise de Dados**: O sistema deve permitir a análise dos dados para identificar áreas de melhoria e tomar decisões estratégicas.

# 4 Requisitos

São especificações documentadas que descrevem as necessidades, expectativas, características ou restrições que um sistema deve atender. Eles são fundamentais para o desenvolvimento de software e outros projetos, pois orientam o design, implementação e avaliação do sistema. Existem dois tipos principais de requisitos

## 4.1 Requisitos Funcionais (RF)

**1. Gerenciamento de Chamados:**

* **Criar e registrar atendimentos:** Os usuários devem poder registrar novas solicitações facilmente, fornecendo informações como nome do solicitante, e-mail para contato, categoria de assunto e descrição da solicitação.
* **Visualizar e acompanhar atendimentos:** Os usuários devem ter acesso a uma lista de suas solicitações, com a opção de ver a resposta recebida pelo administrativo.
* **Atualizar atendimentos:** O administrativo e deve poder atualizar as informações das solicitações como adicionar comentários e alterar o status.
* **Fechar atendimento:** O administrativo deve poder fechar solicitações quando forem solucionadas, e os usuários devem ser notificados sobre o fechamento e a resposta dada.

**2. Gerenciamento de Usuários:**

* **Criar e gerenciar usuários:** O administrador do sistema deve ter a capacidade de criar, editar e excluir usuários do sistema, definir suas permissões e atribuir funções.
* **Atribuir funções:** O administrador deve definir diferentes funções de usuário com diferentes permissões de acesso e funcionalidades (técnico, solicitante, supervisor, etc.).

**3. Categorização de Atendimentos:**

* **Criar e gerenciar categorias:** O administrador deve ter a capacidade de criar e gerenciar categorias de solicitações para facilitar a organização e a triagem.
* **Atribuir categorias:** Ao registrar uma solicitação, o usuário deve poder selecionar a categoria mais adequada ao seu problema.

**4. Fluxo de Trabalho:**

* **Visualizar o histórico:** Os usuários devem ter acesso ao histórico de uma solicitação, incluindo todas as interações, atualizações e resoluções.

**5. Relatórios:**

* **Gerar relatórios básicos:** O sistema deve gerar relatórios básicos sobre o número de chamados, tempo médio de resolução, tipo de problema mais frequente, etc.

## 4.2 Requisitos Não Funcionais (RNF)

**1. Desempenho:**

* **Tempo de resposta:** O sistema deve ter um tempo de resposta rápido para registro, triagem e atualização de chamados, minimizando o tempo de espera para os usuários.
* **Escalabilidade:** O sistema deve ser capaz de suportar um número crescente de usuários, chamados e dados sem comprometer o desempenho.
* **Disponibilidade:** O sistema deve estar disponível para uso a maior parte do tempo, com indisponibilidades mínimas para minimizar o impacto nas operações da empresa.

**2. Segurança:**

* **Autenticação e autorização:** O sistema deve ter mecanismos robustos de autenticação para garantir que apenas usuários autorizados acessem o sistema e suas funcionalidades.
* **Confidencialidade e integridade dos dados:** O sistema deve proteger os dados dos chamados contra acesso não autorizado, modificação ou perda.
* **Prevenção de ataques:** O sistema deve ter medidas de segurança para prevenir ataques cibernéticos, como invasão, malware e phishing.

**3. Usabilidade:**

* **Interface amigável:** O sistema deve ter uma interface amigável e intuitiva, fácil de usar para técnicos e solicitantes, mesmo para aqueles com pouca familiaridade com tecnologia.
* **Acessibilidade:** O sistema deve ser acessível a pessoas com deficiências, seguindo as diretrizes de acessibilidade web.
* **Ajuda e documentação:** O sistema deve fornecer ajuda e documentação claras e completas para auxiliar os usuários na utilização do sistema.

**4. Confiabilidade:**

* **Estabilidade:** O sistema deve ser estável e funcionar sem falhas frequentes ou travamentos.
* **Recuperação de desastres:** O sistema deve ter um plano de recuperação de desastres para garantir a continuidade das operações em caso de falhas de hardware ou software.
* **Monitoramento:** O sistema deve ser monitorado para identificar e solucionar problemas potenciais antes que causem indisponibilidade ou perda de dados.

**5. Manutenabilidade:**

* **Código modular:** O código do sistema deve ser modular, bem documentado e fácil de entender, para facilitar futuras modificações e manutenções.
* **Atualizações:** O sistema deve ser fácil de atualizar com novas funcionalidades e correções de bugs.
* **Suporte técnico:** O fornecedor do sistema deve oferecer suporte técnico de qualidade para auxiliar na resolução de problemas e dúvidas dos usuários.

**6. Portabilidade:**

* **Compatibilidade com diferentes plataformas:** O sistema deve ser compatível com diferentes navegadores web e sistemas operacionais para permitir o acesso de qualquer lugar.
* **Integração com outras ferramentas:** O sistema deve ser facilmente integrado com outras ferramentas de TI que você já utiliza.

**7. Adequação legal e regulatória:**

* **Conformidade com leis e regulamentações:** O sistema deve estar em conformidade com todas as leis e regulamentações aplicáveis ao seu setor e região, como leis de proteção de dados.

# 5 Diagramas de UML

**5.1 Diagrama de Casos de Uso:**

* **Atores:**
  + Solicitante: Usuário que registra e acompanha solicitações.
  + Técnico: Usuário que atende as solicitações.
  + Administrador: Usuário que gerencia o sistema e define configurações.
* **Casos de Uso:**
  + Registrar solicitação: O solicitante registra uma nova solicitação de atendimento, fornecendo informações como nome do solicitante, e-mail para contato, categoria do assunto e descrição da solicitação.
  + Visualizar solicitação: O solicitante e o técnico visualizam a lista de suas solicitações.
  + Atualizar solicitação: O técnico atualiza as informações do chamado como adicionar comentários e alterar o status.
  + Fechar solicitação: O técnico fecha a solicitação quando o problema for solucionado e o solicitante é notificado.
  + Gerenciar Usuários: O administrador cria, edita e exclui usuários, define permissões e atribui funções.
  + Gerar Relatórios: O administrador gera relatórios básicos sobre o número de solicitações.

**5.2. Diagrama de Classes:**

* **Classes:**
  + Solicitação: Representa uma solicitação individual, com atributos como nome do solicitante, e-mail para contato, categoria do assunto e descrição da solicitação.
  + Usuário: Representa um usuário do sistema, com atributos como login, email e senha.
  + Categoria: Representa uma categoria de assunto de uma solicitação, para facilitar a organização e triagem.
* **Relacionamentos:**
  + Uma solicitação é aberta por um solicitante.
  + Uma solicitação é atribuída a um técnico.
  + Uma solicitação pode ser categorizada em uma categoria.
  + Um solicitante pode abrir várias solicitações.
  + Um técnico pode ser responsável por vários chamados.
  + Um administrador gerencia os usuários e categorias.

# 6 Banco

## 6.1 Modelo Físico

CREATE DATABASE IF NOT EXISTS `senac\_pizzaria` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci;

USE `senac\_pizzaria`;

CREATE TABLE `contatos` (

`id` int(11) NOT NULL,

`nome` varchar(255) NOT NULL,

`email` varchar(255) NOT NULL,

`assunto` varchar(50) DEFAULT NULL,

`mensagem` text NOT NULL,

`situacao` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Despejando dados para a tabela `contatos`

--

INSERT INTO `contatos` (`id`, `nome`, `email`, `assunto`, `mensagem`, `situacao`) VALUES

(1, 'daniel', 'd@niel.com', 'RECLAMACOES', 'sou gay', 'Finalizado'),

(2, 'lucas', 'luc@s.com', 'ELOGIOS', 'teste', 'Finalizado'),

(3, 'teste', 'teste@t.com', 'SOLICITACOES', 'teste3', 'Finalizado'),

(4, 'mat', 'm@m.com', 'SUGESTOES', 'teste4', 'Em Atendimento'),

(5, 'a', 'asd@asd.com', 'SOLICITACOES', 'tedste', 'Não atendido'),

(6, 'b', 'b@b.com', 'SOLICITACOES', 'teste5', 'Não atendido'),

(7, 'c', 'c@c.com', 'SOLICITACOES', 'teste6', 'Não atendido'),

(8, 'd', 'd@d.com', 'DUVIDAS', 'teste7', 'Não atendido'),

(9, 'zequinha', 'nnnxxx@bbbb.com', 'RECLAMACOES', 'xxxxxxxxxxxxxxxxx', 'Finalizado');

CREATE TABLE `usuarios` (

`id` int(11) NOT NULL,

`nome` varchar(255) DEFAULT NULL,

`email` varchar(255) DEFAULT NULL,

`senha` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Despejando dados para a tabela `usuarios`

--

INSERT INTO `usuarios` (`id`, `nome`, `email`, `senha`) VALUES

(5, 'ADMIN', 'admin@a.com', 'admin123'),

(6, 'mat', 'm@t.com', 'qwerty'),

(8, 'asd', 'asd@asd.com', 'asd');

CREATE TABLE `usuario\_contato` (

`usuario\_id` int(11) NOT NULL,

`contato\_id` int(11) NOT NULL,

`situacao` varchar(255) NOT NULL,

`resposta` text DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Despejando dados para a tabela `usuario\_contato`

--

INSERT INTO `usuario\_contato` (`usuario\_id`, `contato\_id`, `situacao`, `resposta`) VALUES

(5, 2, 'Finalizado', 'Foi mandado email agradecendo pelo elogio'),

(5, 3, 'Finalizado', 'asdoijasdoiasd'),

(5, 4, 'Em Atendimento', ''),

(5, 9, 'Finalizado', 'Agradecemos sua colaboração, ela é muito importante para nós.');

# 7 Telas

## 7.1 Homepage

Interface gráfica do usuário, Texto, Aplicativo, Email, Site

Descrição gerada automaticamente

## 7.2 Tela de fale conosco

Interface gráfica do usuário, Site

Descrição gerada automaticamente

## 7.3 Tela de login

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

## 7.4 Tela de lista de solicitações de atendimentos

Tabela, Linha do tempo

Descrição gerada automaticamente

## 7.5 Tela de lista de meus atendimentos

Tabela, Linha do tempo

Descrição gerada automaticamente

## 7.6 Tela de finalização de atendimento

Linha do tempo

Descrição gerada automaticamente

## 7.7 Tela de visualização de resposta ao atendimento

Linha do tempo

Descrição gerada automaticamente

## 7.8 Tela de Lista de usuários administrativos cadastrados

Linha do tempo

Descrição gerada automaticamente

## 7.9 Tela de cadastro de novos usuários administrativos

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

## 7.10 Tela de edição de usuários administrativos

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

# 8 Conclusão