

Centro Universitário SENAC

Projeto Integrador

Análise de Soluções de Tecnologia da Informação

FixIT

Caio Raphael Pereira dos Santos: 1142479100;

Guilherme Henrique Veloso Santos: 114231780;

Gustavo de Medeiros Sanchez: 1142733596;

Mariana Nery de Andrade Domingues: 1142717296.

São Paulo

2025

Introdução

No atual cenário corporativo, a tecnologia desempenha um papel fundamental na eficiência operacional das empresas, garantindo a continuidade dos serviços e aprimorando a experiência do cliente. No entanto, um dos grandes desafios enfrentados por muitas organizações é a gestão eficiente de chamados técnicos e suporte aos usuários. Problemas como demora na resolução de incidentes, falta de rastreabilidade das solicitações e comunicação ineficiente entre usuários e equipe de suporte são recorrentes e impactam diretamente a produtividade e a qualidade dos serviços prestados.

Diante desse contexto, este projeto propõe o desenvolvimento do FixIT, um sistema de suporte técnico inovador, projetado para oferecer uma solução robusta e eficiente para a gestão de chamados. O FixIT permitirá que empresas acompanhem e gerenciem suas solicitações de suporte de forma estruturada, garantindo mais transparência, agilidade e controle sobre as demandas.

A proposta visa a criação de uma plataforma intuitiva e acessível, capaz de oferecer funcionalidades avançadas, como categorização automática de chamados, sistema de notificações em tempo real, painéis de monitoramento com indicadores de desempenho e integração com diferentes canais de comunicação, como e-mail e chat interno. Além disso, a segurança da informação será uma prioridade, garantindo que apenas usuários autorizados possam acessar e manipular os dados do sistema.

Por meio da implementação do FixIT, as organizações poderão reduzir o tempo médio de resposta a chamados, otimizar o trabalho das equipes de suporte e melhorar a experiência dos usuários finais, promovendo um ambiente mais organizado e eficiente. Este documento apresenta a estrutura detalhada do projeto, abordando desde a concepção do problema até a arquitetura da solução, tecnologias empregadas e os requisitos necessários para a implementação bem-sucedida do sistema.

Descrição do Problema

As empresas frequentemente enfrentam desafios na gestão de chamados técnicos, impactando diretamente a eficiência da equipe de suporte e a satisfação dos usuários. A ausência de um sistema adequado para a organização e o acompanhamento das solicitações pode gerar diversos problemas, como a perda de informações cruciais, atrasos na resolução de chamados e falhas na comunicação entre técnicos e usuários.

Além disso, a falta de um mecanismo estruturado dificulta a priorização de chamados conforme sua criticidade, resultando em alocações inadequadas de recursos. Isso pode levar a longos tempos de espera, baixa produtividade dos técnicos e, consequentemente, impactos negativos no desempenho geral da empresa. A inexistência de um painel de monitoramento eficiente também compromete a tomada de decisões estratégicas, pois os gestores não possuem visibilidade em tempo real sobre a performance do suporte técnico.

Outro ponto crítico é a dificuldade de rastrear o histórico de chamados e soluções aplicadas, tornando o aprendizado contínuo mais desafiador e aumentando a repetição de problemas que poderiam ser evitados. Dessa forma, a implementação de um sistema robusto e intuitivo, como o FixIT, torna-se essencial para otimizar o fluxo de atendimento, reduzir tempos de resposta e proporcionar um ambiente mais eficiente e organizado para todas as partes envolvidas.

Justificativa

A implementação de um Sistema de Suporte Técnico eficiente visa otimizar o atendimento a chamados, reduzindo o tempo de resposta e aprimorando a gestão de solicitações de serviço. Com isso, melhora-se a experiência do cliente e a produtividade da equipe de suporte.

O FixIT trará maior controle e organização ao fluxo de suporte, garantindo que todas as solicitações sejam devidamente registradas, priorizadas e acompanhadas até sua resolução. A adoção de uma solução especializada reduzirá erros manuais, evitará a perda de informações importantes e permitirá um monitoramento detalhado da eficiência do suporte técnico. Com uma interface intuitiva e recursos automatizados, o sistema possibilitará uma comunicação mais clara entre usuários e técnicos, garantindo que cada chamado receba a devida atenção e seja resolvido de maneira eficiente.

Além disso, o FixIT permitirá que gestores obtenham relatórios detalhados sobre a performance da equipe de suporte, auxiliando na identificação de gargalos e na tomada de decisões estratégicas para aprimorar continuamente o serviço prestado. A implementação dessa solução não apenas melhorará a produtividade dos técnicos, como também proporcionará um ambiente mais estruturado e ágil para todas as partes envolvidas.

Descrição das Personas

Técnico de Suporte: Responsável por solucionar chamados.

Usuário Final: Cliente que reporta problemas e solicita suporte.

Administrador: Acompanha métricas e desempenho dos técnicos.

Macro Descrição da Proposta de Solução

1. Levantamento de Requisitos (24/02/2025 a 27/02/2025 - 4 dias) Definição do Plano de Projeto.

Estruturação do repositório no GitHub. Identificação das personas e regras de negócio.

Elaboração da Proposta Técnica.

2. Modelagem e Prototipagem (28/02/2025 a 20/03/2025 - 21 dias) Prototipação do produto.

Implementação inicial das funcionalidades. Estruturação do Plano de Teste.

Inicialização do DockerHub.

Implementação inicial (CRUD de login e registro). Gravação da prototipagem do login/registro.

3. Desenvolvimento Backend e Frontend (20/03/2025 a 15/05/2025 - 56 dias)

Sprint 01 (20/03 - 10/04, 21 dias): Desenvolvimento da base do sistema e ajustes no login/registro.

Sprint 02 (10/04 - 01/05, 21 dias): Implementação do painel de controle e integração para criação de chamados.

Sprint 03 (01/05 - 15/05, 14 dias): Implementação de notificações e melhorias.

4. Testes e Ajustes (15/05/2025 a 05/06/2025 - 21 dias) Testes funcionais e de usabilidade.

Correção de bugs e ajustes finais.

5. Implantação do MVP e Treinamento (15/05/2025 a 05/06/2025 - 21 dias) Gravação do vídeo de apresentação.

Disponibilização do MVP no DockerHub.

Escopo do Projeto

O projeto **FixIT** tem como objetivo o desenvolvimento de um sistema web responsivo voltado à gestão de chamados técnicos. A aplicação será acessível por navegadores modernos e dispositivos móveis, com uma interface intuitiva e adaptável aos diferentes perfis de usuário.

O sistema contemplará as seguintes funcionalidades:

- **Controle de acesso baseado em perfis de usuário**, com permissões específicas para:
 - **Usuário comum (cliente)**: poderá cadastrar chamados e acompanhar o andamento.
 - **Técnico de suporte**: terá acesso aos chamados disponíveis, podendo se atribuir a um chamado, alterar o status (aberto, em andamento, concluído) e responder ao solicitante.
 - **Administrador**: responsável por gerenciar os usuários e técnicos cadastrados, atribuir chamados, visualizar relatórios e realizar ações administrativas como a exclusão de chamados e usuários.
- **Cadastro e rastreamento de chamados**, incluindo campos como título, descrição, prioridade, categoria, status, data de criação e data de encerramento, com histórico acessível para técnicos e clientes.
- **Integração com banco de dados relacional** (H2 no ambiente de desenvolvimento), garantindo a persistência e integridade das informações.
- **Painel de controle adaptado ao perfil do usuário**, apresentando somente as funcionalidades permitidas para aquele tipo de acesso.
- **Validações de segurança e usabilidade**, impedindo o acesso indevido a funcionalidades restritas e proporcionando retorno claro ao usuário em caso de erro.
- **Automação de fluxos críticos**, como redirecionamento automático após o login conforme o perfil, validação de cadastro para impedir duplicidade de usuários e controle na atribuição de chamados.

Regras de Negócio

Todo chamado aberto deve ser classificado de acordo com sua **prioridade** (baixa, média ou alta), para que a equipe de suporte possa organizar o atendimento com base na criticidade da solicitação.

Apenas usuários autenticados têm permissão para acessar o sistema e realizar ações como abertura, visualização ou acompanhamento de chamados.

O processo de atendimento a chamados segue uma lógica definida por perfis:

- **Usuários comuns** podem apenas abrir chamados, visualizar seu histórico e acompanhar o status das solicitações em tempo real.
- **Técnicos de suporte** devem se atribuir manualmente a um chamado que ainda não tenha responsável, ficando impedidos de assumir chamados já atribuídos.
- Técnicos são obrigados a **atualizar o status do chamado** ao longo do processo de atendimento (de "Aberto" para "Em andamento" e posteriormente para "Concluído").

O sistema deve impedir:

- A **atribuição múltipla** de técnicos ao mesmo chamado.
- A **edição de chamados finalizados**, garantindo integridade do histórico.
- O **cadastro de usuários com login duplicado**.

Chamados podem ser **excluídos apenas por administradores**, desde que estejam em aberto ou não afetem o histórico técnico.

Todo chamado deve pertencer a uma **categoria** previamente cadastrada no sistema, garantindo organização temática e futura análise de dados.

Requisitos Funcionais

O sistema deve permitir a abertura de chamados com os seguintes campos: título, descrição detalhada, anexos, seleção de prioridade (baixa, média, alta) e categoria.

Deve possibilitar ao usuário acompanhar em tempo real o status do chamado (Aberto, Em andamento, Concluído), além de visualizar o histórico completo de interações, mensagens e atualizações relacionadas ao atendimento.

Requisitos Não Funcionais

O sistema deve suportar pelo menos 100 acessos simultâneos sem degradação de desempenho perceptível.

O tempo médio de resposta da interface para ações essenciais (como login, abertura de chamados e consultas) não deve ultrapassar 2 segundos.

A aplicação deve ser responsiva, garantindo compatibilidade e boa usabilidade em navegadores modernos e dispositivos móveis.

Diagramas

Diagrama de Classes

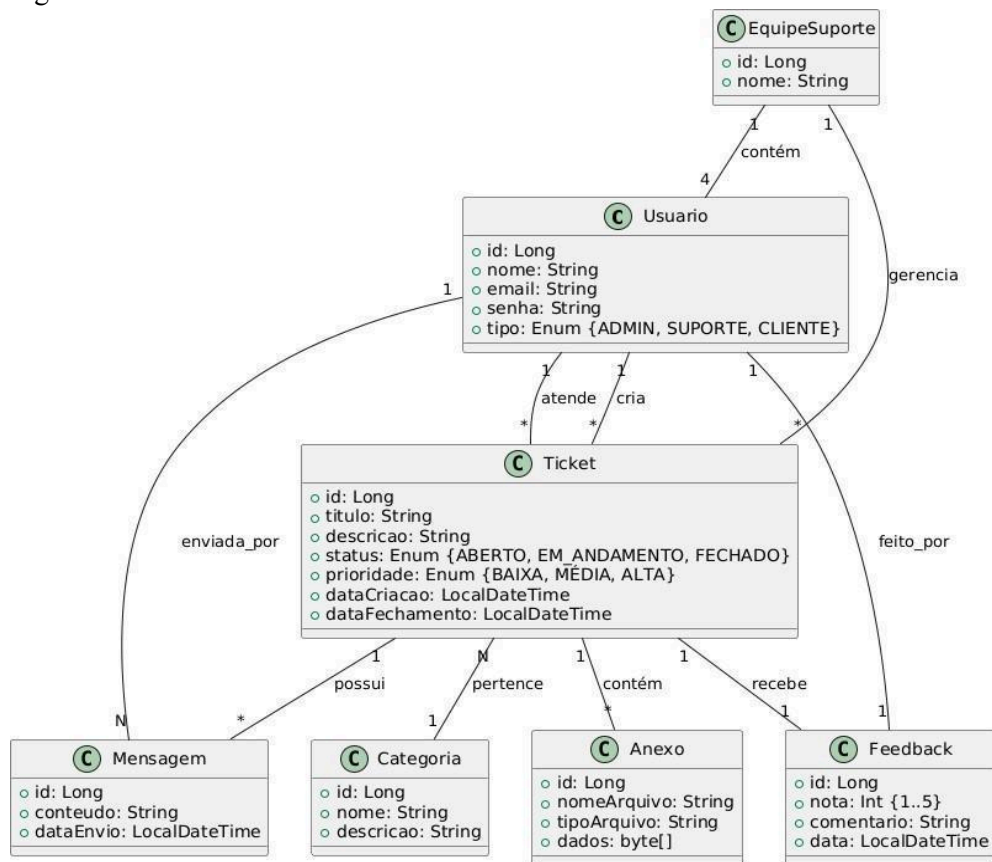
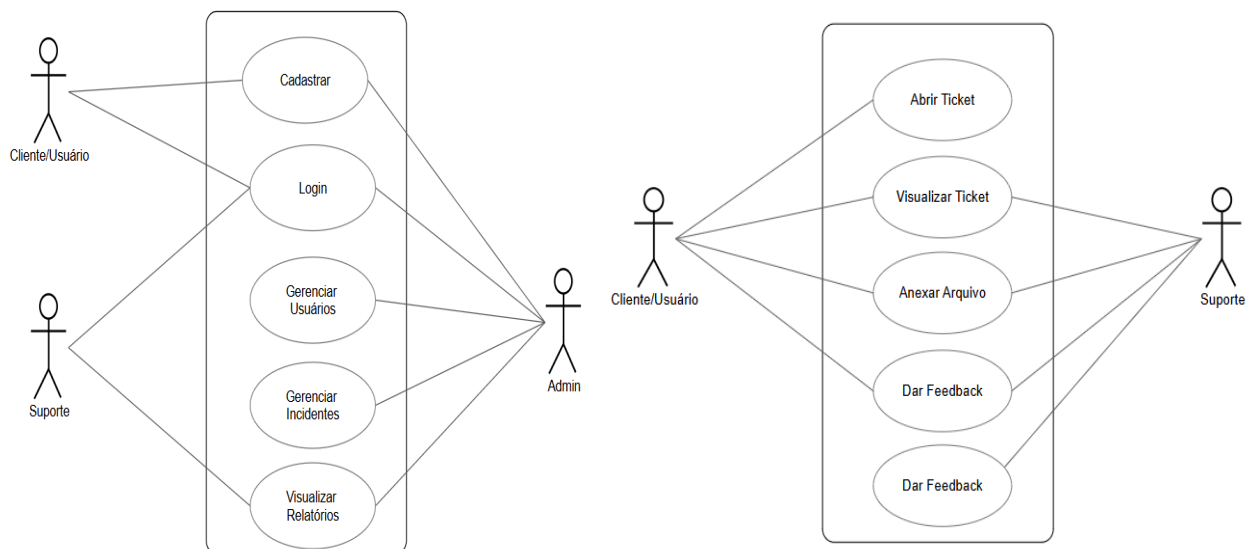


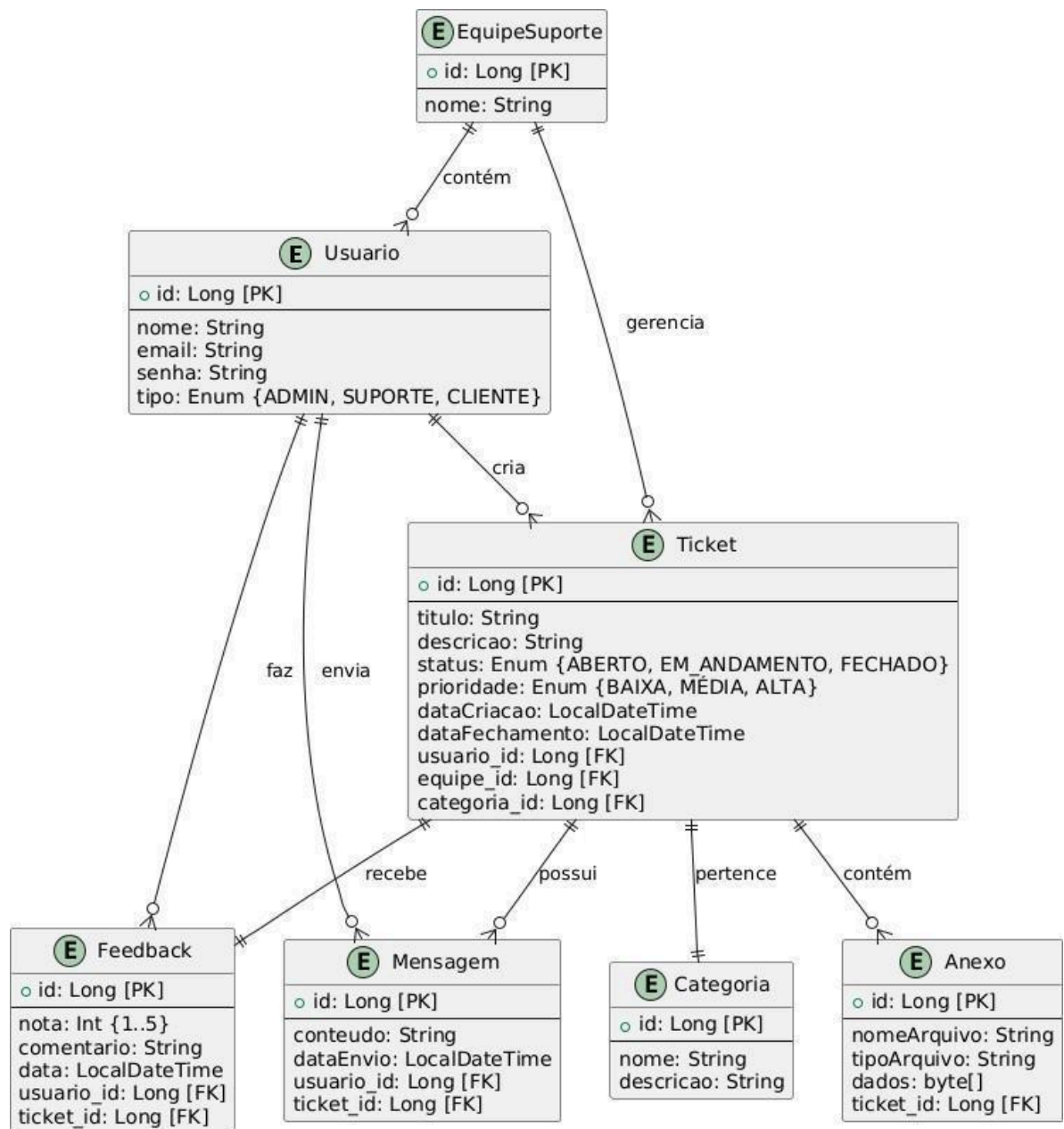
Diagrama de Caso de Uso



Modelos

Banco de Dados

MER/DER: (



Código - Diagrama de Classe:

@startuml

```
class Usuario
```

```
{
```

```
    +id: Long
```

```
    +nome: String
```

```
    +email: String
```

```
    +senha: String
```

```
    +tipo: Enum {ADMIN, SUPORTE, CLIENTE}
```

```
}
```

```
class EquipeSuporte {
```

```
    +id: Long
```

```
    +nome: String
```

```
}
```

```
class Ticket {
```

```
    +id: Long
```

```
    +titulo: String
```

```
    +descricao: String
```

```
    +status: Enum {ABERTO, EM_ANDAMENTO, FECHADO}
```

```
    +prioridade: Enum {BAIXA, MÉDIA, ALTA}
```

```
    +dataCriacao: LocalDateTime
```

```
    +dataFechamento: LocalDateTime
```

```
}
```

```
class Mensagem {  
  
    +id: Long  
  
    +conteudo: String  
  
    +dataEnvio: LocalDateTime  
  
}
```

```
class Categoria {  
  
    +id: Long  
  
    +nome: String  
  
    +descricao: String  
  
}
```

```
class Anexo {  
  
    +id: Long  
  
    +nomeArquivo: String  
  
    +tipoArquivo: String  
  
    +dados: byte[]  
  
}
```

```
class Feedback {  
  
    +id: Long  
  
    +nota: Int {1..5}  
  
    +comentario: String  
  
    +data: LocalDateTime  
  
}
```

Usuario "1" -- "*" Ticket : cria

Usuario "1" -- "*" Ticket : atende

EquipeSuporte "1" -- "4" Usuario :

contém EquipeSuporte "1" -- "*" Ticket :

gerencia Ticket "1" -- "*" Mensagem :

possui Ticket "1" -- "*" Anexo : contém

Ticket "N" -- "1" Categoria : pertence

Ticket "1" -- "1" Feedback : recebe

Mensagem "N" -- "1" Usuario :

enviada_por Feedback "1" -- "1" Usuario :

feito_por @enduml

Código - MERDER:

@startuml

entity Usuario {

+id: Long [PK]

--

nome: String

email: String

senha:

String

tipo: Enum {ADMIN, SUPORTE, CLIENTE}

}

entity EquipeSuporte {

+id: Long [PK]

--

nome: String

}

entity Ticket {

+id: Long [PK]

--

titulo: String

descricao:

String

status: Enum {ABERTO, EM_ANDAMENTO, FECHADO}

prioridade: Enum {BAIXA, MÉDIA,

ALTA} dataCriacao: LocalDateTime

```
dataFechamento:

LocalDateTime usuario_id:

Long [FK] equipe_id: Long

[FK] categoria_id: Long [FK]

}
```

```
entity Mensagem {

+id: Long [PK]

--

conteudo: String

dataEnvio:

LocalDateTime

usuario_id: Long [FK]

ticket_id: Long [FK]

}
```

```
entity Categoria {

+id: Long [PK]

--

nome: String

descricao:

String

}
```

```
entity Anexo {

+id: Long [PK]

--

nomeArquivo: String

}
```



```
    tipoArquivo: String
    dados: byte[]
    ticket_id: Long [FK]
}
```

```
entity Feedback {
    +id: Long [PK]
    --
    nota: Int {1..5}
    comentario: String
    data: LocalDateTime
    usuario_id: Long [FK]
    ticket_id: Long [FK]
}
```

```
Usuario ||--o{ Ticket : cria
Usuario ||--o{ Mensagem : envia
Usuario ||--o{ Feedback : faz
EquipeSuporte ||--o{ Usuario :
    contém EquipeSuporte ||--o{ Ticket :
        gerencia Ticket ||--o{ Mensagem :
            possui Ticket ||--o{ Anexo : contém
Ticket ||--|| Categoria : pertence
Ticket ||--|| Feedback : recebe
```

@enduml

Código - Caso de Uso:

@startuml

actor Cliente

actor Suporte

actor Admin

usecase "Abrir Ticket" as UC1 usecase

"Visualizar Tickets" as UC2 usecase

"Enviar Mensagem" as UC3 usecase

"Anexar Arquivo" as UC4 usecase

"Fechar Ticket" as UC5

usecase "Gerenciar Categorias" as UC6

usecase "Gerenciar Usuários" as UC7

usecase "Visualizar Relatórios" as UC8

usecase "Gerenciar Equipe de Suporte" as UC9

usecase "Dar Feedback" as UC10

Cliente --> UC1

Cliente --> UC2

Cliente --> UC3

Cliente --> UC4

Cliente --> UC10

Suporte --> UC2

Suporte --> UC3

Suporte --> UC4

Suporte --> UC5 Suporte --> UC9

Admin --> UC6

Admin --> UC7

Admin --> UC8

Admin --> UC9

@enduml

Especificação da Arquitetura de Software

Ecossistema da Solução

O sistema operará em um ambiente web, utilizando serviços em nuvem para armazenamento e segurança dos dados. Usuários interagiram com a plataforma via navegador ou dispositivos móveis.

Melhorias na Versão 2.0 do FixIT

A versão 2.0 do sistema FixIT trouxe melhorias significativas em relação à versão inicial, especialmente no que diz respeito à arquitetura, autenticação de usuários e controle de acesso baseado em papéis. Com a migração do uso de LocalStorage para persistência real via banco de dados H2 e autenticação controlada pelo backend (Spring Boot), as funcionalidades agora atendem requisitos reais de segurança, integridade e escalabilidade.

Entre os principais aprimoramentos da Sprint 02 estão:

- Implementação completa do fluxo de login com autenticação via backend.
- Controle de acesso com base em papéis (Admin, Técnico, Usuário Comum).
- Redirecionamento dinâmico de interface conforme o tipo de usuário logado.
- Validações de segurança para impedir que usuários não autorizados acessem funcionalidades restritas.
- Criação chamados por usuário e atribuição exclusiva para técnicos..
- Atualização de status de chamados com histórico visível para técnicos e usuários.
- Testes de fluxo técnico ampliados com foco em falhas e exceções controladas.

Essas melhorias foram validadas com uma ampliação no plano de testes, incluindo novos fluxos técnicos, testes de exceção e controle de permissões, garantindo mais robustez à solução.

Melhorias na Versão 3.0 do FixIT

A versão 3.0 do sistema FixIT representou um avanço expressivo em termos de segurança, inteligência de análise e eficiência na gestão de chamados. A principal ênfase desta sprint foi o aprimoramento da rastreabilidade de dados e a robustez das funcionalidades administrativas, aliadas à consolidação de uma base segura de autenticação.

Entre os principais aprimoramentos da Sprint 03 estão:

- Criptografia de senhas com o algoritmo BCrypt, assegurando maior proteção das credenciais dos usuários no banco de dados.
- Registro completo de chamados persistido no backend, com integração de status, prioridade, data e histórico de ações.
- Funcionalidade de atribuição de técnicos a chamados, realizada pelo administrador ou pelo próprio técnico, com validações para evitar múltiplas atribuições.
- Implementação de um painel de relatórios dinâmico, com indicadores visuais atualizados em tempo real sobre volume de chamados, prioridades, status e setores mais demandados.
- Ajustes de segurança nos endpoints e no fluxo de atendimento, fortalecendo a integridade dos dados e o controle de permissões.

Essas melhorias consolidam o FixIT como uma plataforma mais segura, escalável e orientada à análise estratégica, permitindo uma atuação proativa da equipe de suporte e uma tomada de decisão mais eficaz por parte dos gestores.

Melhorias na Versão 4.0 do FixIT

A Sprint 04 tem como foco a expansão da comunicação e colaboração dentro dos chamados, com a proposta de implementação de um **módulo de comentários**.

- Criação de um sistema de comentários nos chamados, permitindo que técnicos e usuários interajam diretamente dentro do chamado.
- Os comentários serão ordenados por data e associados ao autor, com identificação de perfil (usuário, técnico, admin).
- Funcionalidade de atribuição de técnicos a chamados, realizada pelo administrador ou pelo próprio técnico, com validações para evitar múltiplas atribuições.
- Possibilidade futura de anexar arquivos leves ou capturas dentro dos comentários (versão futura).

Essa funcionalidade visa tornar o atendimento mais transparente e colaborativo, evitando comunicações externas como e-mails ou mensagens soltas, além de manter todo o histórico registrado no chamado.

Definição das Tecnologias a serem Empregadas

Linguagem: Java, JavaScript

Banco de Dados: H2(desenvolvimento), PostgreSQL(produção)

Segurança: Autenticação e Criptografia de senhas com BCrypt

Repositórios: [GitHub](#), DockerHub

Provedores em Nuvem: AWS (Amazon Web Services) e Vercel

Recursos Utilizados no Projeto

Frameworks de Desenvolvimento: SpringBoot, React e Next.js.

Softwares de Apoio: Figma (prototipação), Jira (gerenciamento de tarefas).

Artefatos para Sprint 03

1. Link do protótipo do [Figma](#).
2. Plano de teste: anexado na entrega.
3. Link da Imagem em [Dockerhub](#) e testes direto no [deploy da aplicação](#).
4. Link da gravação do [vídeo](#).
5. Link do Backend do projeto: [GitHub](#).