

## **TCHÊFLUX – FERRAMENTA DE SERVICE DESK**

**LUCAS SILVA GARCIA<sup>1</sup>, MARCELO HENRIQUE EUZEBIO BATISTA<sup>2</sup>**

<sup>1</sup> Acadêmico em Análise e Desenvolvimento de Sistemas

<sup>2</sup> Titulação

*Faculdade Dom Bosco de Porto Alegre, Porto Alegre, Rio Grande do Sul*  
[22113001@faculdadedombosco.edu.br](mailto:22113001@faculdadedombosco.edu.br); [euzebio@faculdadedombosco.edu.br](mailto:euzebio@faculdadedombosco.edu.br)

### **RESUMO**

#### **Palavras-chave:**

## **TCHÊFLUX – SERVICE DESK TOOL**

### **ABSTRACT**

#### **Keywords:**

### **1 INTRODUÇÃO**

A Gestão de serviços de TI se apresenta como um desafio nas organizações de todos os portes. Em se tratando de empresas de menor porte, o custo de implantação inicial da gestão estratégica de TI através do modelo ITIL pode se tornar um impedimento. Neste contexto, emerge a necessidade de pensar em soluções que atendam de forma simplificada e com baixo custo a implantação de processos de controle e gestão.

Diversos são os problemas enfrentados por instituições que possuem gestão de TI em grau insuficiente. Incidentes recorrentes, falta de gestão de informações e insatisfação dos clientes são alguns dos problemas recorrentes dessa realidade, modelos de gestão permitem diminuir o impacto desses incidentes. Uma das ferramentas do modelo é o service desk virtual, uma evolução do helpdesk tradicional, tem como objetivo facilitar a comunicação entre solicitantes, usuários enfrentando problemas, com atendentes capazes de solucioná-los.

A partir disso, defini-se como objetivo do presente trabalho desenvolver uma plataforma de service desk baseada em um web-site. Possuindo a funcionalidade de tickets de atendimento e suas informações.

Isto posto este documento esta estruturado em 5 sessões, além desta introdução. Na 2 sessão apresenta o referencial teórico. Na sessão 3 demonstra a modelagem de dados realizada. Seguindo a 4 sessão apresenta os requisitos levantados e como irão ser implementados. Ao modo de finalização são apresentadas as consideracoes finais representado a reflexão ao após o termino deste.

## **2 REFERENCIAL TEÓRICO**

No contexto da gestão de serviços em ti, existem diversos modelos visando a padronização de processos em uma empresa, dentre estes, o ITILv4, desenvolvido pela AXELOS, destaca-se como líder na industria. Segundo AXELOS (2019), o modelo fornece sistemas flexíveis, coordenados e integrados, buscando uma gestão e governança efetiva para organizações.

Dentre as diversas ferramentas contempladas pelo modelo, a service desk é apresentada como um ponto de comunicação entre o provedor de serviços e o seus usuários. Possibilita o relato de problemas e solicitações, e que estas sejam reconhecidas, classificadas, atribuidas e atendidas pelos atores responsáveis. Dentre as diversas soluções de service desk virtual existentes no mercado, pode-se citar: ZohoDesk, Jira Service Management, FreshService. Nesse contexto, o presente trabalho implementa uma ferramenta de service desk baseada em um web-site. Dentro da plataforma, solicitantes e atendentes podem registrar e atender tickets respectivamente, mantendo o controle das ações e comunicações realizadas durante o ciclo de atendimento do ticket.

## **2 SOLUÇÃO IMPLEMENTA : ARQUITETURA**

Esta seção apresenta as tecnologias escolhidas para o desenvolvimento da ferramenta de software Tchêflux. A justificativa para tais escolhas se dá pelos padrões de mercado e pelo conhecimento e familiaridade do autor com estas.

### **2.1 FRONT-END**

#### **2.1.1 Bootstrap**

A tecnologia escolhida para o desenvolvimento do front-end do projeto é o bootstrap. Segundo Bootstrap Team (2025), é um framework web que utiliza-se de HTML, CSS e

JavaScript para formar um sistema de malha e componetes. Seu uso facilita o desenvolvimento de páginas web, permitindo reutilização e customizações de componentes.

## **2.2 BACK-END**

### **2.2.1 NodeJS**

A tecnologia escolhida para o desenvolvimento do back-end do projeto é o NodeJS. Segundo **OPENJSF (2025)**, é um ambiente de execução em tempo real multiplataforma de JavaScript. Utilizando o motor V8, compartilhado com o navegador Google Chrome, e utilizar a linguagem JavaScript, permite ser uma plataforma muito performática e fácil desenvolvimento.

## **2.3 BANCO DE DADOS**

### **2.3.1 PostgreSQL**

A tecnologia escolhida para o desenvolvimento do banco de dados do projeto é o PostgreSQL. Segundo The Postgresql Global Development Group (2025), é um sistema de gerenciamento de banco de dados relacional. Suporta funções como: pesquisas complexas, chaves estrangeiras, gatilhos, visualizações e integridade transacional.

## **3 DOCUMENTOS**

Aqui serão apresentados os documentos criados visando suportar o desenvolvimento da ferramenta de software. Estes documentos são importantes do ponto de vista da engenharia de software, segundo Sommerville (2018), tanto para agregar valor ao projeto quanto para garantir manutenibilidade futura.

### **3.1 MODELO DE DADOS**

Segundo Elmasri (2016), O processo de modelagem de dados é essencial dentro do contexto do desenvolvimento do software, contendo etapas como: modelagem conceitual, modelagem logica e modelagem fisica. Tem como objetivo representar a realidade do ambiente que o software irá abordar com qualidade, representando o mini-mundo da melhor maneira e definir como seus dados serão armazenados. Seu uso dentro do contexto de desenvolvimento resulta na melhor padronização de dados, garante a qualidade da solução desenvolvida, favorece a manutenibilidade de longo prazo e a melhor compreensão dos dados armazenados.

Durante o desenvolvimento do projeto tchêflux, foi realizado o processo de modelagem de dados. Nas sessões seguintes, serão descritas as etapas do processo, como foram implementadas dentro do contexto do projeto e os resultados obtidos, na forma do modelo de dados. Desta maneira, podemos melhor entender o processo de mapeamento de dados e como foi aplicado no projeto tchêflux.

### 3.1.1 Modelagem Conceitual

A modelagem conceitual consiste na etapa inicial do processo de mapeamento. Nesse período, são levantadas as entidades que compõem o modelo, quais atributos elas possuem e como se relacionam entre si. Ao finalizada, resulta em um documento que espelha a realidade do mini-mundo, servindo como base sólida para as próximas etapas da modelagem. Durante esse processo, no projeto, foram mapeadas as entidades usuário, ticket e departamento, cada entidade contendo seus atributos, e seus relacionamentos com outras entidades. O resultado produzido foram o diagrama entidade relacionamento e seu dicionário de dados listados abaixo.

#### Usuario

Atributo	Domínio	OBS
idUsuario	SERIAL	Identificador
nomeCompleto	VARCHAR(100)	Obrigatório
telefone	VARCHAR(15)	Obrigatório
email	VARCHAR(255)	Obrigatório
senhaHash	VARCHAR(255)	Obrigatório
tipo	TIPO	Obrigatório

#### Ticket

Atributo	Domínio	OBS
nro	SERIAL	Identificador
status	STATUS	Obrigatório ( 'Aberto', 'Em Andamento', 'Pendente Cliente', 'Resolvido', 'Fechado', 'Cancelado' )
descrição	TEXT	Obrigatório
titulo	VARCHAR(100)	Obrigatório
dataCriacao	TIMESTAMP	Obrigatório
dataAtualizacao	TIMESTAMP	Opcional

--	--	--

#### Departamento

Atributo	Domínio	OBS
codDepto	SERIAL	Identificador
areas	AREA	(RH,FN,MKT,OP,TI)

### 3.1.2 Modelagem Lógica

A modelagem lógica consiste na etapa intermediária do processo de mapeamento. Nesse período, a partir do diagrama ER e seu dicionário de dados, é construído as tabelas que irão pertencer ao banco de dados, contendo as entidades, atributos e relacionamento previamente levantados. Ao finalizada, resulta em um documento que define como as tabelas serão implementadas junto ao banco de dados, em conjunto de suas restrições, sendo base para a criação do script de DDL.

Durante esse processo no projeto, baseado no modelo ER e o seu dicionário de dados, foram mapeadas as seguintes tabelas: usuário, ticket e departamento. Cada tabela contém os atributos de sua respectiva entidade e os relacionamentos que ela possui, além das restrições presentes. O resultado produzido foram as tabelas listadas abaixo.

#### Usuario

(idUsuario, nomeCompleto, telefone, email, senhaHash, tipo, idAtendente, idSolicitante, codDepto, nro)

codDepto referencia Departamento

nro referencia Ticket

#### Ticket

(nro, status, descricao, titulo, dataInicio, dataAtualizacao, codDepto, idAtendente, idSolicitante)

idAtendente referencia Usuario

idSolicitante referencia Usuario

codDepto referencia Departamento

#### Departamento

(codDepto, areas)

### 3.1.3 Modelagem Física

A modelagem física consiste na etapa final do processo de mapeamento. Nesse período, a partir do diagrama ER, dicionário de dados e as tabelas definidas, é construído o script de DDL, contendo os passos necessários para a criação física do banco de dados pelo SGBD escolhido. Por fim, resulta em um schema de banco de dados completo, permitindo a utilização pelo software das devidas funcionalidades, como criar, ler, atualizar e deletar dados.

Durante esse processo no projeto, baseado no modelo ER, dicionário de dados e tabelas levantadas, foi desenvolvido o script abaixo. Como resultado, foi possível realizar a configuração e utilização do banco de dados necessário para o funcionamento da aplicação.

```
DROP TABLE IF EXISTS Ticket;  
DROP TABLE IF EXISTS Usuario;  
DROP TABLE IF EXISTS Departamento;  
DROP TYPE IF EXISTS tipo;  
DROP TYPE IF EXISTS status;  
DROP TYPE IF EXISTS area;
```

```
CREATE TYPE area AS ENUM('RH', 'FN', 'MKT', 'OP', 'TI');
```

```
CREATE TABLE Departamento (  
    codDepto SERIAL PRIMARY KEY,  
    areas area NOT NULL UNIQUE  
);
```

```
CREATE TYPE tipo AS ENUM('Solicitante', 'Atendente');
```

```
CREATE TABLE Usuario (  
    idUsuario SERIAL PRIMARY KEY,  
    nomeCompleto VARCHAR(100) NOT NULL,  
    telefone VARCHAR(15) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    senhaHash VARCHAR(255) NOT NULL,  
    tipo tipo NOT NULL,  
    codDepto INT,  
    FOREIGN KEY (codDepto) REFERENCES Departamento(codDepto),  
    CONSTRAINT chk_tipo_depto CHECK (  
        (tipo = 'Solicitante' AND codDepto IS NULL) OR  
        (tipo = 'Atendente' AND codDepto IS NOT NULL)  
    )  
);
```

```
CREATE TYPE status AS ENUM('Aberto', 'Em Andamento', 'Pendente Cliente', 'Resolvido',
```

'Fechado', 'Cancelado');

```
CREATE TABLE Ticket (  
    nro SERIAL PRIMARY KEY,  
    titulo VARCHAR(100) NOT NULL,  
    descricao TEXT NOT NULL,  
    status status NOT NULL DEFAULT 'Aberto',  
    dataInicio TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    dataAtualizacao TIMESTAMP,  
    idSolicitante INT NOT NULL,  
    idAtendente INT,  
    codDepto INT NOT NULL,  
    FOREIGN KEY (idSolicitante) REFERENCES Usuario(idUsuario),  
    FOREIGN KEY (idAtendente) REFERENCES Usuario(idUsuario),  
    FOREIGN KEY (codDepto) REFERENCES Departamento(codDepto)  
);
```

```
INSERT INTO departamento (areas) VALUES ('RH') ;
```

```
INSERT INTO departamento (areas) VALUES ('FN') ;
```

```
INSERT INTO departamento (areas) VALUES ('MKT');
```

```
INSERT INTO departamento (areas) VALUES ('OP');
```

```
INSERT INTO departamento (areas) VALUES ('TI')
```

#### **4.1 ESPECIFICAÇÃO FUNCIONAL**

Segundo SOMMERVILLE (2018), O processo de levantamento de requisitos compreende na descrição de como o sistema deve se comportar, suas restrições e seu funcionamento. Tem como documento primordial a especificação de requisitos, onde são definidos de maneira definitiva como o software irá ser implementado pelo time de desenvolvimento. Sua concepção permite que as diversas partes envolvidas dentro do processo estejam de acordo, e que eventuais mudanças sejam de fácil implementação e compreensão.

Durante o desenvolvimento do projeto tchêflux, foi realizado o processo de levantamento de requisitos. Nas sessões seguintes, serão descritas as etapas do processo, como foram implementadas dentro do contexto do projeto e os resultados obtidos, na forma da especificação funcional. Desta maneira, podemos melhor entender o processo de levantamento de requisitos e como foi aplicado no projeto tchêflux.

Durante o processo de especificação de requisitos de sistemas, foi levantado o requisito UC001 – Gerenciamento de Tickets. Composto esse requisito, foram elaborados o diagrama de caso de uso,

utilizando UML, e fluxo principal com BPMN e seus detalhes, como restrições. A união de todos os itens listados resultou na lista de requisitos do projeto, utilizada para guiar o desenvolvimento, em conjunto com o modelo de dados.

#### 4.1.1 UC001 – Gerenciamento de Tickets

A funcionalidade tem como objetivo permitir o usuário realizar o cadastro na plataforma como solicitante ou atendente de tickets. Durante o cadastro, o usuário deve preencher seus dados iniciais, como: nome completo, telefone, e-mail, senha de acesso e definir o tipo de conta que será criada. Por fim, deve ser gerado um cadastro válido que permita o usuário acessar o sistema e realizar e realizar suas atividades.

##### **Fluxo**

1. Usuário Acessa Cadastro
2. Usuário Preenche Campos Obrigatórios
3. Usuário Clica em registrar
4. Sistema valida se os dados são válidos
5. Sistema registra novo usuário no banco de dados
6. Sistema retorna o status da solicitação de cadastro
7. Sistema redireciona para a tela de log-in

##### **Detalhamento**

Resultado

1. O sistema deve permitir o cadastro completo do usuário

Campos obrigatórios

1. Nome Completo, telefone, e-mail, senha, tipo de usuário



# Interface

Tchefflux [Home](#) [Sobre Nós](#) [Login](#)

## Logo Tchefflux Cadastro de Novo Usuário

Nome Completo
Telefone
Endereço de e-mail
Senha
Tipo de Usuário Selecione o tipo de usuário
Registrar

## 5 CONSIDERAÇÕES FINAIS

Este trabalho teve por objetivo o desenvolvimento de um software de service desk, seguindo conceito presente dentro do modelo ITIL.

Durante este processo foram desenvolvidos os documentos de levantamento de requisitos, modelo de dados, dando suporte aos dados presentes no sistema, requisitos descrevendo as funcionalidades esperadas do projeto e como deveriam ser implementas, e o código da aplicação.

A ferramenta desenvolvida tem funcionalidades que representam a possibilidade de colocá-la em produção. Considerando aplicabilidade do projeto, considera-se como contribuição a possibilidade de implantação sem custo desta ferramenta em ambientes de produção.

## 5 REFERÊNCIAS

BOOTSTRAP TEAM (org.). **Get started with Bootstrap**. 2025. Disponível em: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. Acesso em: 07 jun. 2025.

OPENJSF. **Introduction-to-nodejs**. Disponível em: <https://nodejs.org/pt/learn/getting-started/introduction-to-nodejs>. Acesso em: 07 jun. 2025.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **What Is PostgreSQL**. Disponível em: <https://www.postgresql.org/docs/17/intro-what-is.html>. Acesso em: 07 jun. 2025.

AXELOS (org.). **ITIL Foundation: ITIL 4 Edition**. 4. ed. Londres, 2019.

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2018.

ELMASRI, Ramez. **Sistemas de Banco de Dados**. 7. ed. São Paulo: Pearson, 2016.