

**CENTRO PAULA SOUZA**

**Etec EURO ALBINO DE SOUZA**

**Ensino Técnico Integrado ao Médio em Desenvolvimento de  
Sistemas**

**Daniel Dias de Oliveira Júnior**

**Gabriel de Oliveira Cavenaghi**

**Leonardo Vilela da Silva**

**Matheus de Oliveira**

**Miguel Zimmermann Martins Silva**

**NEW T.E.**

**Mogi Guaçu – SP**

**2025**

**Daniel Dias de Oliveira Júnior**

**Gabriel de Oliveira Cavenaghi**

**Leonardo Vilela da Silva**

**Matheus de Oliveira**

**Miguel Zimmermann Martins Silva**

**NEW T.E.**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Ensino Médio Integrado ao Técnico em desenvolvimento de sistemas da Etec Euro Albino de Souza, orientado pelo Prof. Maércio Girardi Bisco e Pedro Ramires da Silva Amalfi Costa, como requisito parcial para a obtenção do título de técnico em Desenvolvimento de Sistemas.

**Mogi Guaçu – SP**

**2025**

O48n

Oliveira Júnior, Daniel Dias de; Cavenaghi, Gabriel de Oliveira; Silva Leonardo Vilela da; Oliveira, Matheus de; Silva, Miguel Zimmermann Martins.

NEW T.E. – Mogi Guaçu – SP. 2025.

121 p.

Trabalho de Conclusão do Curso de Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas da ETEC Euro Albino de Souza de Mogi Guaçu.

Orientadores: Professor Especialista Maércio Girardi Bisco e Professor Mestre Pedro Ramires da Silva Amalfi Costa

1. Onboarding. 2. Integração. 3. Gamificação. I. Bisco, Maércio Girardi; Costa, Pedro Ramires da Silva Amalfi (orientadores) II. Título.

CDD 621.381 5

Bibliotecário Guerreiro Ímpar Nunes Guerreiro – CRB-8/8137

## DEDICATÓRIA

Dedicamos esta obra para todos os formandos da ETEC ou de outras redes de ensino para um processo de estágio próspero e proveitoso.

## **AGRADECIMENTOS**

Agradecemos a Deus e a todos os professores da ETEC Euro Albino de Souza pela recepção calorosa durante a escrita do TCC e de todos os outros momentos da vida escolar do ensino médio.

## EPÍGRAFE

“Revisar o que foi aprendido para conhecer o novo, assim pode-se tornar um mestre.” – Confúcio

## RESUMO

O presente Trabalho de Conclusão de Curso propõe o desenvolvimento do sistema New T.E., uma solução digital para o processo de *onboarding* de novos funcionários em empresas. O projeto visa responder à crescente necessidade de integração eficiente, estruturada e tecnológica de colaboradores no ambiente corporativo. A proposta abrange funcionalidades como gamificação de tarefas, comunicação por *chat*, agendamento de atividades e supervisão em tempo real, promovendo maior engajamento e adaptação do colaborador à cultura da empresa. Com base em pesquisas de viabilidade operacional, técnica e econômica, o sistema foi desenvolvido utilizando tecnologias como Python, PyQt5 e banco de dados MySQL, hospedado remotamente pela Hostinger. Além disso, o trabalho aborda aspectos legais e contratuais fundamentais para a operação de um sistema SaaS, com foco em conformidade com a LGPD e outras legislações pertinentes. O projeto também passou por testes de usabilidade e heurísticas de Nielsen, a fim de garantir uma interface intuitiva e acessível. Por fim, é apresentada uma proposta concreta de monetização do produto e sua aplicabilidade real no mercado.

**Palavras-Chave:** *Onboarding*; Integração; Gamificação; Python.

## **ABSTRACT**

*The New T.E. system was developed as a digital solution to improve the onboarding process of new employees. It seeks to meet the growing demand for efficient, structured, and technology-based integration in companies. The system includes features such as task gamification, chat communication, class and material registration, and real-time supervision, helping employees engage and adapt to company culture. Developed with Python, PyQt5, and a MySQL database hosted by Hostinger, the project also discusses legal aspects of SaaS systems, with a focus on compliance with the Brazilian General Data Protection Law (LGPD). Usability tests and Nielsen's heuristics were applied to ensure a simple and user-friendly interface. Finally, the work presents possible business models and the potential of applying the system in the market.*

**Keywords:** *Onboarding; Integration; Gamification; Python.*

## LISTA DE ILUSTRAÇÕES

Figura 1 – Pergunta sobre conhecimentos a respeito do termo .....	55
Figura 2 - Especificação sobre o conhecimento de <i>onboarding</i> .....	56
Figura 3 - Realização do <i>onboarding</i> de forma digital .....	56
Figura 4 - Expectativas de funcionalidades do sistema .....	57
Figura 5 - Opinião acerca de processos de <i>onboarding</i> que já participaram .....	57
Figura 6 - Utilidade de um sistema que ajude com informações sobre o emprego ...	58
Figura 7 - Dificuldades em usar uma aplicação nova .....	58
Figura 8 - Expectativa dos usuários para os processos do sistema .....	59
Figura 9 - Pergunta sobre gamificação do sistema .....	59
Figura 10 - Pergunta sobre interatividade do processo .....	60
Figura 11 - Opinião sobre <i>feedbacks</i> na aplicação .....	60
Figura 12 - Pergunta sobre guia interativo durante uso do sistema .....	61
Figura 13 - Tela de <i>Login</i> (Protótipo).....	62
Figura 14 - <i>Dashboard</i> do Supervisor (Protótipo).....	63
Figura 15 – Tabela de funcionários (Protótipo) .....	64
Figura 16 - Tela de edição de funcionários (Protótipo).....	64
Figura 17 – Calendário (Protótipo) .....	65
Figura 18 - Tela de <i>Chat</i> (Protótipo).....	66
Figura 19 - Tela de Cadastro (Protótipo).....	66
Figura 20 - Perfil do supervisor (Protótipo).....	67
Figura 21 - Tela inicial do funcionário (Protótipo) .....	67
Figura 22 - Primeira heurística .....	68
Figura 23 - Segunda heurística .....	69
Figura 24 - Terceira heurística .....	69
Figura 25 - Quarta heurística.....	70
Figura 26 - Quinta heurística .....	71
Figura 27 - Sexta heurística .....	71
Figura 28 - Sétima heurística .....	72
Figura 29 - Oitava heurística .....	73
Figura 30 - Nona heurística .....	73
Figura 31 - Logo da I9Ação .....	80
Figura 32 - Logo da UserGuiding .....	81

Figura 33 - Logo da Pipefy .....	82
Figura 34 - Logo da Mambo.IO .....	83
Figura 35 - Logo da Wildgoose .....	84
Figura 36 - Tela de <i>Login</i> .....	97
Figura 37 - Código da Tela de Login 1 .....	98
Figura 38 - Código da Tela de Login 2 .....	99
Figura 39 - Tela de Recuperação de Senha .....	100
Figura 40 - Código da Tela de Recuperação de Senha 1 .....	101
Figura 41 - Código da Tela de Recuperação de Senha 2 .....	102
Figura 42 - Código da Tela de Recuperação de Senha 3 .....	103
Figura 43 - Tela de Atividades (Supervisor) .....	104
Figura 44 - Tela de Lição (Supervisor) .....	105
Figura 45 - Cadastro de Lições .....	106
Figura 46 - Visualização de Lições.....	107
Figura 47 - Atualização de Lições .....	108
Figura 48 - Exclusão de Lições .....	109
Figura 49 - Tela de Chat .....	110
Figura 50 - Trecho do Código do Chat .....	111
Figura 51 - Tela de Cadastro de Funcionário .....	112
Figura 52 - Trecho de Código do Cadastro de Funcionário .....	113

## **LISTA DE TABELAS**

Tabela 1 - Cronograma do desenvolvimento do TCC .....	40
Tabela 2 - Planejamento da viabilidade econômica do projeto .....	41

## **LISTA DE ABREVIATURAS E SIGLAS**

AI – Inteligência Artificial (*Artificial Intelligence*)

API – Interface de Programação de Aplicações (*Application Programming Interface*)

ARPANET – Rede da Agência de Projetos de Pesquisa Avançada (*Advanced Research Projects Agency Network*)

CERN – Organização Europeia para a Pesquisa Nuclear (*Conseil Européen pour la Recherche Nucléaire*)

CLT – Consolidação das Leis de Trabalho

CRMs – Sistemas de Gestão de Relacionamento com o Cliente (*Customer Relationship Management*)

CRUD – Cadastro, Visualização, Atualização, Exclusão (*Create, Read, Update, Delete*)

ENIAC – Computador Integrador Numérico Eletrônico (*Electronic Numerical Integrator and Computer*)

ERPs – Sistemas de Planejamento de Recursos Empresariais (*Enterprise Resource Planning*)

FTP – Protocolo de Transferência de Arquivos (*File Transfer Protocol*)

GCC – Coleção de Compiladores GNU (*GNU Compiler Collection*)

GIL – Bloqueio Global do Interpretador (*Global Interpreter Lock*)

GUI – Interface Gráfica do Usuário (*Graphical User Interface*)

HTML – Linguagem de Marcação de Hipertexto (*HyperText Markup Language*)

HTTP – Protocolo de Transferência de Hipertexto (*HyperText Transfer Protocol*)

HTTPS – Protocolo de Transferência de Hipertexto Seguro (*HyperText Transfer Protocol Secure*)

ICC – Compilador C++ da Intel (*Intel C++ Compiler*)

IoT – Internet das Coisas (*Internet of Things*)

JIT – Compilação Sob Demanda (*Just-In-Time*)

LAN – Rede Local (*Local Area Network*)

LGPD – Lei Geral de Proteção de Dados

MinGW – GNU Minimalista para Windows (*Minimalist GNU for Windows*)

ML – Aprendizado de Máquina (*Machine Learning*)

MOC – Compilador de Metaobjetos (*Meta-Object Compiler*)

MSVC – Compilador C++ da Microsoft (*Microsoft Visual C++*)

NFC – Comunicação por Campo de Proximidade (*Near Field Communication*)

OS – Sistema Operacional (*Operating System*)

PC – Computador Pessoal (*Personal Computer*)

PEP – Proposta de Aprimoramento do Python (*Python Enhancement Proposal*)

QML – Linguagem de Modelagem Qt (*Qt Modeling Language*)

QPL – Licença Pública Qt (*Qt Public License*)

RAM – Memória de Acesso Aleatório (*Random Access Memory*)

SMTP – Protocolo Simples de Transferência de Correio (*Simple Mail Transfer Protocol*)

SSD – Unidade de Estado Sólido (*Solid State Drive*)

UI – Interface do Usuário (*User Interface*)

URI – Identificador Uniforme de Recursos (*Uniform Resource Identifier*)

UX – Experiência de Usuário (*User Experience*)

WAN – Rede de Longa Distância (*Wide Area Network*)

WYSIWYG – O Que Você Vê é o Que Você Obtém (*What You See Is What You Get*)

# SUMÁRIO

INTRODUÇÃO .....	16
1 - TECNOLOGIAS .....	18
1.1 - LÓGICA DE PROGRAMAÇÃO .....	18
1.2 – LINGUAGEM DE PROGRAMAÇÃO .....	18
1.2.1 – <i>Desktop</i> .....	20
1.3 – SERVIDORES.....	21
1.3.1 – <i>Web</i> .....	22
1.4 – PYTHON .....	23
1.5 – QT .....	27
1.5.1 - Qt Widgets Designer.....	29
1.5.2 – PyQt5.....	29
1.6 – BANCO DE DADOS.....	33
1.6.1 – MySQL.....	34
1.6.2 - Modelagem.....	35
2 - ESTUDO DE CASO .....	37
2.1 – PESQUISA DE VIABILIDADE.....	38
2.1.1 – Viabilidade Operacional .....	38
2.1.2 – Viabilidade Técnica.....	39
2.1.3 – Viabilidade de Cronograma .....	40
2.1.4 – Viabilidade Econômica.....	41
2.2 – TÉCNICA DE LEVANTAMENTO DE REQUISITOS .....	42
2.2.1 – Entrevistas .....	42
2.2.2 – Formulário.....	53
2.2.3 Análise do Formulário .....	55
2.3 – PROTOTIPAGEM .....	61

2.3.1 – Interface do Supervisor .....	63
2.3.2 – Interface do Funcionário .....	67
2.3.3 – Heurísticas de Nielsen .....	68
2.4- CLIENTES .....	74
2.4.1 – Cliente Interno.....	74
2.4.2 – Cliente Externo .....	75
2.4.3 – Clientes Ativos .....	75
2.4.4 – Cliente Inativo .....	76
2.4.5 – Cliente Satisfeito .....	76
2.4.6 – Cliente Insatisfeito.....	77
2.5 – ÁREA DE ATUAÇÃO .....	77
2.5.1 – Microrregião.....	78
2.5.2 - Macrorregião.....	79
2.6 - CONCORRENTES .....	80
2.6.1 - Concorrentes Diretos.....	80
2.6.2 - Concorrentes Indiretos .....	83
2.7 - LEGISLAÇÃO E CONTRATUAL.....	84
2.7.1 - Legislação .....	85
2.7.2 – Contratual .....	88
2.7.2.6 - Cláusula 6 – Deveres do Cliente .....	90
2.7.3 – Contrato de Licença.....	92
3 - DOCUMENTAÇÃO .....	97
3.1 – TELAS E CÓDIGOS DA APLICAÇÃO EM PYTHON .....	97
3.1.1 – Tela de <i>Login</i> .....	97
3.1.2 – Código da Tela de <i>Login</i> .....	98
3.1.3 – Tela de Recuperação de Senha .....	100
3.1.4 – Código da Tela de Recuperação de Senha .....	101

3.1.5 – Tela de Atividades (Supervisor).....	104
3.1.6 – Tela de Lição Individual .....	105
3.1.7 – CRUD das Lições .....	106
3.1.8 – Telas e Código do <i>Chat</i> .....	110
3.1.9 – Tela e Código do Cadastro de Funcionários.....	112
CONSIDERAÇÕES FINAIS .....	115
REFERÊNCIAS BIBLIOGRÁFICAS .....	116
ANEXO.....	121

## INTRODUÇÃO

O avanço da tecnologia e da transformação digital nas últimas décadas impactou profundamente a forma como as empresas se organizam, treinam e integram seus colaboradores. Processos que antes dependiam quase exclusivamente de métodos presenciais e manuais passaram a contar com soluções digitais que oferecem maior eficiência, agilidade e praticidade. Nesse cenário, o *onboarding* de novos funcionários ganha destaque, visto que é um processo crucial para a adaptação e retenção de talentos em ambientes corporativos cada vez mais dinâmicos.

A integração mal estruturada pode gerar dificuldades de adaptação, baixa produtividade e até desligamentos precoces, aumentando os custos organizacionais. Por outro lado, quando bem conduzido, o *onboarding* garante que o colaborador se sinta acolhido, compreenda rapidamente suas responsabilidades e desenvolva um vínculo mais sólido com a cultura da empresa. Por isso, soluções digitais voltadas a essa área tornam-se um diferencial competitivo, sobretudo em setores que apresentam alta rotatividade de profissionais.

O presente Trabalho de Conclusão de Curso apresenta o desenvolvimento do sistema New T.E., uma plataforma digital cujo objetivo é apoiar o processo de integração de novos colaboradores. O sistema oferece recursos como o cadastro de funcionários, o cadastro de aulas e materiais de apoio ao aprendizado, a implementação de um chat para comunicação entre usuários e interfaces desenvolvidas em Python com PyQt5, integradas a um banco de dados MySQL hospedado remotamente. Além da implementação técnica, este estudo considera aspectos legais, como a Lei Geral de Proteção de Dados, e questões de ética profissional, fundamentais para o uso responsável de tecnologias no ambiente corporativo.

Metodologicamente, o projeto foi construído a partir de pesquisas exploratórias, entrevistas com estagiários e supervisores de empresas, além de análises de práticas de integração já existentes. Essa abordagem permitiu compreender melhor as dificuldades enfrentadas em ambientes de alta rotatividade e fundamentar as escolhas no desenvolvimento do sistema.

No que se refere à organização do trabalho, o **Capítulo 1** apresenta a fundamentação teórica, abordando conceitos relacionados à integração de novos colaboradores, ao papel da tecnologia na área de recursos humanos e às práticas de gamificação aplicadas ao ambiente corporativo. O **Capítulo 2** descreve os procedimentos metodológicos adotados, explicando o processo de levantamento de informações, definição de requisitos e planejamento do sistema. Por fim, o **Capítulo 3** detalha o desenvolvimento da aplicação, incluindo a modelagem de dados, a implementação das funcionalidades e a análise dos resultados obtidos a partir de testes de usabilidade.

Dessa forma, este TCC busca contribuir com uma solução prática e inovadora para o processo de *onboarding*, promovendo não apenas eficiência operacional, mas também maior engajamento e satisfação dos colaboradores, alinhando tecnologia e gestão de pessoas às demandas atuais do mercado.

## 1 - TECNOLOGIAS

Para melhor compreensão a respeito do projeto, esta seção é dedicada para apresentar as tecnologias utilizadas em seu desenvolvimento, desde os princípios da programação e suas bases, programas, servidores *Web* e bibliotecas inseridas, junto de sua história e versões até a escrita deste documento.

### 1.1 - LÓGICA DE PROGRAMAÇÃO

A lógica de programação refere-se ao conjunto de raciocínios e processos que os programadores utilizam ao desenvolver sistemas computacionais. A criação desses sistemas depende da organização lógica das etapas necessárias para atingir uma solução, estruturadas em sequências de instruções. Essas sequências são fundamentais para o desenvolvimento de *software*, orientando as ações dos programadores na construção de sistemas informatizados. De acordo com CARDOSO (2024) “Trata-se do conjunto de conceitos e técnicas que ajudam a resolver problema utilizando uma linguagem de programação ou uma pseudolinguagem.”

O conceito de algoritmo é fundamental na ciência da computação e na matemática, sendo uma abstração que descreve um conjunto finito de passos bem definidos e ordenados que levam à resolução de um problema específico. Esse termo tem uma história rica que remonta a muitos séculos, e um dos primeiros matemáticos a contribuir significativamente para o desenvolvimento desse conceito foi Muhammad ibn Musa al-Khwarizmi, um matemático persa que viveu durante o período islâmico entre os séculos VIII e IX. (SANTOS, 2023)

O processo da lógica da programação pode ser comparado à confecção de um bolo, quando um programador aplica os princípios de entrada, processamento e saída dos dados, seria como um confeitiro, colocando os ingredientes na forma ordenadamente, para o processamento no forno, e na saída, se recebe um delicioso bolo.

### 1.2 – LINGUAGEM DE PROGRAMAÇÃO

A linguagem de programação é uma linguagem formal e estruturada que define um conjunto de instruções e regras utilizadas para criar programas (*software*). Esses programas podem ser desenvolvidos para rodar em diferentes plataformas, como computadores, dispositivos móveis ou qualquer outro equipamento capaz de executar o código gerado. Existem várias linguagens e elas servem para muitos propósitos. Alguns óbvios, como criar um *software*, outros menos, como controlar um carro ou uma torradeira, (MONTEIRO, 2017).

Um *app* que roda no seu celular é um tipo de *software* criado com uma linguagem de programação. Um *game*, que roda em um *Personal Computer* – Computador Pessoal (PC) ou em outros equipamentos, também. Uma *TV smart*, que possui a funcionalidade de se conectar na *internet* e permitir ao consumidor usar um navegador, usa linguagens de programação tanto no *hardware* quanto no *software*, (\_\_\_\_, 2017).

Uma torradeira não tem uma interface digital de comunicação, mas possui um microcontrolador com um *software* que gerencia o equipamento. Esse tipo de *software*, embarcado em circuitos eletrônicos, é chamado de *firmware* e também é escrito usando uma linguagem de programação. A *internet* das coisas (IoT) demanda muitas soluções que usam esse tipo de tecnologia, (\_\_\_\_, 2017).

Com os diferentes tipos de linguagens existentes, pode-se citar as mais utilizadas no mundo em 2025, seguindo a ordem: Python, Java, Javascript, C++, C#. Existem hoje, no mundo, mais de 700 linguagens para se escolher e aprender, (BARRO, 2025).

Nos primórdios da programação, o principal método de linguagem eram os próprios 0s e 1s, praticada pelos pioneiros da computação, apesar de inovadora, era complicada de se realizar e muito fácil de se cometer erros, (MENDES, 2023).

Depois indo para a próxima geração há linguagens como o Assembly, que introduziu inovações como símbolos, entretanto, o desenvolvimento ainda era complexo, (\_\_\_\_, 2023).

Indo para a terceira geração surgem as linguagens mais conhecidas nos dias de hoje como C++, C# e Python, que tornaram a programação mais simplificada e acessível ao público. (\_\_\_\_, 2023)

Na quarta geração, surgem plataformas e ferramentas que permitem a programação com uma linguagem mais visual, com interfaces gráficas, permitindo um desenvolvimento de programas de forma mais acelerada. (\_\_\_\_, 2023)

E para a quinta e sexta geração, aparecem as linguagens baseadas ou que tem um suporte em inteligência artificial, com elas simplificando ainda mais os processos da programação, elas são atarefadas com processos complexos que seriam impossíveis para o ser humano como armazenamento de dados, permitindo que programadores físicos, tivessem um foco maior em outra área do desenvolvimento, (\_\_\_\_, 2023).

### 1.2.1 – *Desktop*

Refere-se ao processo de criação de *software* que é executado diretamente em computadores pessoais (*desktops* e *laptops*), em contraste com outros tipos de desenvolvimento de *software*, como desenvolvimento *web* ou desenvolvimento de aplicativos móveis. Este tipo de *software* é projetado para ser instalado e executado no sistema operacional do usuário, aproveitando ao máximo os recursos do *hardware* local. De forma mais simplificada, os programas instalados no seu computador são chamados de *softwares desktop*. Alguns desses *softwares* não precisam de conexão à *internet* para funcionar ou possuem versões *offline*; no entanto, também existem aqueles que são *online* e precisam de acesso à *internet*. Exemplos de *softwares offline* incluem o Word e o Excel, embora eles também tenham suas versões *online*. (R.S, 2024)

Segundo WK TECHNOLOGY (2022) “As aplicações *desktop*, *software* projetado para computadores *desktop* (como Windows ou Mac), funcionam em um computador de cada vez. Não está disponível para mais ninguém, a menos que o usuário o envie por *e-mail* ou por um *drive* externo”.

A evolução do desenvolvimento de *softwares* em *desktops* ou computadores começa na década de 1940 com o *Electronic Numerical Integrator Computer* – Computador Integrador Numérico eletrônico (ENIAC), que tinha como base de processamento válvulas e cartões com furos, a programação era feita com instruções manuais e era de baixo nível, (AWARI, 2023).

Alguns anos no futuro o avanço progrediu para o uso de transistores, que aumentava a velocidade de processamento, e desenvolveu novas linguagens e compiladores, (\_\_\_\_, 2023).

Nos anos 1970, são introduzidos os microprocessadores, junto à uma revolução nas tecnologias, como a linguagem C, (\_\_\_\_, 2023).

Na década seguinte surgem mais computadores pessoais com interfaces gráficas, como o Windows por exemplo, tornando o uso de *desktops* mais acessíveis ao público, (\_\_\_\_, 2023).

Nos anos 90, com a expansão da *internet* de forma global, o desenvolvimento de *softwares* ganha um novo significado, com novas linguagens, não só para aplicativos *desktops*, surgindo, (\_\_\_\_, 2023).

### 1.3 – SERVIDORES

Um servidor é um computador projetado para executar tarefas específicas, contando com um ou mais processadores, memória *Random Access Memory* – Memória de Acesso Aleatório (RAM), interfaces de comunicação e *softwares* especializados. Além disso, pode incluir sistemas de armazenamento, como discos rígidos internos ou unidades *Solid State Drive* – Unidade de Estado Sólido (SSD), (CONTROLE NET, 2025).

O termo também se refere a um componente dentro de um ambiente computacional mais amplo, responsável por processar aplicações, fornecer serviços e armazenar informações. Esses servidores podem ser físicos ou virtuais, operando localmente ou em ambientes remotos. Em geral, são altamente escaláveis e possuem grande capacidade de processamento, (\_\_\_\_, 2025).

De acordo com \_\_\_\_ (2025) “Capazes de executar um conjunto específico de programas ou protocolos para fornecer serviços para outras máquinas ou clientes, servidores são equipamentos dedicados a executar aplicações e serviços dentro de uma rede LAN<sup>1</sup> ou WAN<sup>2</sup>”.

Os primeiros servidores surgiram logo no início da era dos computadores, na metade do século XX e, assim como a tecnologia evoluiu transformando os equipamentos ao longo dos anos, o seu uso também foi adaptado para atender melhor às necessidades dos negócios conforme a Transformação Digital, até chegarmos ao modo como utilizamos hoje, (\_\_\_\_, 2023).

Um dos primeiros registros que se tem é da “*Advanced Research Projects Agency Network*”, a ARPANET, ou em português “Agência para Projetos de Pesquisa Avançadas”, uma rede de computadores construída em 1969, um servidor que antecedeu e ajudou a dar início à *internet*, que era utilizado para a transmissão de dados militares sigilosos e interligação de departamentos de pesquisas nos Estados Unidos da América, (\_\_\_\_, 2023).

Se o objetivo antigamente era apenas a comunicação, com o passar dos anos ele adquiriu outros usos, como a transmissão e recepção de fax pela *internet*, servidor de impressão conectado às impressoras, servidores de *web* para armazenar páginas de *site*, servidores de *e-mail* para armazenar o envio e recebimento de correio eletrônico, servidor de arquivos, de banco de dados, e muitos outros, (\_\_\_\_, 2023).

<sup>1</sup> Local Area Network (Rede Local)

<sup>2</sup> Wide Area Network (Rede de Longa Distância)

### 1.3.1 – Web

De maneira simplificada, um servidor *web* é um computador responsável pelo armazenamento, processamento e entrega dos arquivos dos *sites* para os navegadores, (\_\_\_\_, 2023).

Através deste processo, os servidores *web* carregam e entregam a página solicitada para o navegador do usuário — como o Google Chrome, por exemplo. Os servidores *web* também utilizam o *Simple Mail Transfer Protocol* (SMTP)<sup>1</sup> e *File Transfer Protocol* (FTP)<sup>2</sup> para processar arquivos para *e-mails* e armazenamento, (\_\_\_\_, 2023).

A *Web* funciona ao conectar documentos e recursos na *internet* por meio de *hiperlinks*, permitindo aos usuários navegar entre páginas e acessar as informações de forma interativa. Esta estrutura facilita a exploração e a integração de conteúdos, tornando a navegação mais eficiente e dinâmica, (TAVARES, 2025).

Ao acessar um *site web*, o navegador solicita informações aos servidores, utilizando os *Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertextos (HTTP) ou *Hypertext Transfer Protocol Secure* – Protocolo de Transferência de Hipertextos Seguro (HTTPS). Estes protocolos garantem uma comunicação eficiente entre o cliente e o servidor, sendo que o HTTPS adiciona uma camada de segurança através da criptografia, (\_\_\_\_, 2025).

Os servidores enviam os dados solicitados e o navegador interpreta-os e exibe-os como páginas visíveis. Este processo ocorre em segundos, proporcionando uma navegação fluida, (\_\_\_\_, 2025).

A origem da *Web* remonta a 1980, no CERN (Organização Europeia para a Pesquisa Nuclear), onde Tim Berners-Lee criou o ENQUIRE, um sistema para facilitar o compartilhamento de documentos de pesquisa. Esse sistema continha conceitos fundamentais da *Web* moderna, incluindo ideias da *Web* semântica, (\_\_\_\_, 2025).

Em 1989, Berners-Lee desenvolveu uma proposta de gerenciamento de informações, expandindo a ideia do ENQUIRE para um sistema mais complexo. Com o apoio de Robert Cailliau, ele formalizou a proposta da *World Wide Web* no final de 1990, (\_\_\_\_, 2025).

Usando um computador NeXTcube, Berners-Lee escreveu o primeiro navegador, chamado WorldWideWeb, e também o primeiro servidor da *Web*. Ele criou as primeiras páginas, que descreviam o projeto. Em 6 de agosto de 1991, a *Web* foi finalmente apresentada ao público no grupo de notícias alt.hypertext, (\_\_\_\_, 2025).

O conceito central da *Web* estava no uso do hipertexto, que surgiu em projetos anteriores, como o Xanadu, mas foi Berners-Lee que uniu o hipertexto com a *Internet*. Ele também desenvolveu o URI (*Uniform Resource Identifier*)<sup>3</sup>, essencial para a identificação de recursos na *Web*, (\_\_\_\_, 2025).

Ao contrário de sistemas anteriores, a *Web* utilizava ligações unidimensionais, facilitando a navegação sem necessidade de intervenção do autor do conteúdo. Essa inovação foi crucial para o crescimento exponencial da *Web*, (\_\_\_\_, 2025).

<sup>1</sup> SMTP (Simple Mail Transfer Protocol – Protocolo Simples de Transferência de Correio)

<sup>2</sup> FTP (*File Transfer Protocol* – Protocolo de Transferência de Arquivos)

<sup>3</sup> URI (*Uniform Resource Identifier* – Identificador Uniforme de Recursos)

Em 1993, a CERN anunciou que a *Web* seria livre para todos, impulsionando sua popularização. O lançamento do navegador gráfico Mosaic, desenvolvido na Universidade de Illinois, foi outro marco importante. Ele trouxe gráficos para a *Web*, tornando-a mais acessível e interativa, (\_\_\_\_, 2025). Em 1994, Berners-Lee fundou o World Wide Web Consortium (W3C)<sup>1</sup> para padronizar e expandir a *Web*. Esse movimento consolidou a evolução da *Web*, promovendo sua transformação em uma plataforma global, (\_\_\_\_, 2025).

## 1.4 – PYTHON

O Python é uma linguagem de programação amplamente usada em aplicações da *Web*, desenvolvimento de *software*, ciência de dados e *Machine Learning* (ML)<sup>2</sup>. Os desenvolvedores usam o Python porque é eficiente e fácil de aprender e pode ser executado em muitas plataformas diferentes. O *software* Python pode ser baixado gratuitamente, integra-se bem a todos os tipos de sistema e agiliza o desenvolvimento (AMAZON WEB SERVICES, 2025)

O Python é útil para escrever código do lado do servidor, porque oferece muitas bibliotecas, que consistem em código pré-escrito para funções complexas de *backend*. Os desenvolvedores também usam uma ampla variedade de *frameworks* Python que fornecem todas as ferramentas necessárias para criar aplicações da *Web* com mais rapidez e facilidade. Por exemplo, os desenvolvedores podem criar o esqueleto da aplicação *Web* em segundos, porque não precisam escrever do zero. Eles podem testar usando as ferramentas de testes do *framework* sem depender de ferramentas externas. (\_\_\_\_, 2025)

A origem do Python remonta a dezembro de 1989, quando o programador holandês Guido van Rossum iniciou seu desenvolvimento como um projeto independente. Seu propósito era criar uma linguagem acessível, de fácil compreensão e com uma sintaxe intuitiva. Em fevereiro de 1991, foi lançada a versão 0.9.0, já contendo elementos essenciais que definiriam a linguagem, como tratamento de exceções, funções e suporte a módulos, (SOUSA, 2023).

O crescimento do Python está ligado à sua filosofia de *design*, baseada em clareza, simplicidade e legibilidade do código. Esses princípios ajudaram a construir uma comunidade ativa e colaborativa. A linguagem passou por diversas evoluções ao longo dos anos, sendo a versão 2.0, lançada em 2000, um marco importante com a introdução de *list comprehensions* e um sistema de coleta de lixo automatizada. No entanto, a transição para o Python 3.0, em 2008, foi ainda mais significativa, pois reformulou aspectos da sintaxe, eliminou redundâncias e tornou a linguagem mais coesa. Apesar da resistência inicial da comunidade, essa mudança foi consolidada ao longo do tempo, (\_\_\_\_, 2023).

<sup>1</sup> W3C (World Wide Web Consortium – Consórcio da World Wide Web)

<sup>2</sup> *Machine Learning* (Aprendizado de Máquina)

Nos últimos anos, o Python experimentou um crescimento exponencial em diversas áreas. Sua vasta biblioteca e ecossistema versátil fizeram dele uma escolha popular para desenvolvimento *web*, automação, análise de dados, aprendizado de máquina e inteligência artificial. Tecnologias como Django e Flask facilitaram a criação de aplicações *web*, enquanto ferramentas como NumPy impulsionaram a análise e visualização de dados, (\_\_\_\_, 2023).

### **Versões:**

Python 0.9.0 (1991): Primeira versão pública, com recursos como exceções, funções e módulos, que definiram a estrutura básica da linguagem, (PYTHON SOFTWARE FOUNDATION, 2025).

Python 1.0 (1994): Introdução de recursos como a manipulação de *strings* e a implementação do módulo "os", além de melhorias na gestão de erros e exceções, (\_\_\_\_, 2025).

Python 1.5 (1997): Inclusão de recursos importantes como o módulo *re* (expressões regulares), e melhorias na sintaxe e no gerenciamento de memória, (\_\_\_\_, 2025).

Python 2.0 (2000): Lançamento de recursos como *garbage collection* automático e o início do suporte a Unicode, que facilitaria o trabalho com textos em diferentes idiomas, (\_\_\_\_, 2025).

Python 2.1 (2001): Introdução de classes com *metaclasses* e suporte a *list comprehensions*, uma das características que tornam Python mais expressivo e conciso, (\_\_\_\_, 2025).

Python 2.2 (2001): Apresentou novas funcionalidades significativas, (\_\_\_\_, 2025).

Python 2.3 (2003): Focou em estabilidade e desempenho, com várias correções de *bugs* e melhorias na biblioteca padrão, (\_\_\_\_, 2025).

Python 2.4 (2004): Introduziu novos recursos, incluindo tipos e funções adicionais, (\_\_\_\_, 2025).

Python 2.5 (2006): Apresentou melhorias na confiabilidade e novas funcionalidades, (\_\_\_\_, 2025).

Python 2.6 (2008): Serviu como uma ponte para a transição para a série 3.x, incorporando recursos do Python 3.0 para facilitar a migração, (\_\_\_\_, 2025).

Python 2.7 (2010): Última versão 2.x, com melhorias como a biblioteca *argparse* e a introdução de otimizações de desempenho, mantendo a compatibilidade com Python 2 por mais alguns anos antes de seu fim de vida em 2020, (\_\_\_\_, 2025).

Python 3.0 (2008): Grande reformulação da linguagem, com mudanças incompatíveis com o Python 2, como a alteração na forma de tratamento de *strings* e a divisão entre inteiros (divisão real por padrão), (\_\_\_\_, 2025).

Python 3.1 (2009): Introdução de melhorias no desempenho, como a implementação do *garbage collector* de referência contada e a otimização da execução do código, (\_\_\_\_, 2025).

Python 3.2 (2011): Melhorias na biblioteca padrão e a adição de novos módulos, como o *asyncio* para programação assíncrona (em versão experimental), (\_\_\_\_, 2025).

Python 3.3 (2012): Inclusão de recursos como a nova sintaxe *yield from* e a implementação do PEP 420, que introduziu pacotes *namespace*, além de melhorias na biblioteca padrão, (\_\_\_\_, 2025).

Python 3.4 (2014): Introdução de módulos como *asyncio* para programação assíncrona e melhorias no desempenho, como a implementação de otimizações no interpretador, (\_\_\_\_, 2025).

Python 3.5 (2015): Novos recursos como a introdução de "*type hints*" (anotações de tipo) e a sintaxe *async/await* para programação assíncrona, além de melhorias de desempenho, (\_\_\_\_, 2025).

Python 3.6 (2016): Adoção das *f-strings* para formatação de *strings* e a introdução de outros recursos como a palavra-chave *async* e a ordenação de dicionários, (\_\_\_\_, 2025).

Python 3.7 (2018): Novas funcionalidades como a inserção de variáveis de tipo em funções, melhorias na biblioteca padrão, e o início do modo "*deterministic*" para a execução de código, (\_\_\_\_, 2025).

Python 3.8 (2019): Introdução das expressões de atribuição com *walrus operator*, além de melhorias no sistema de tipos e mais otimizações, (\_\_\_\_, 2025)

Python 3.9 (2020): Remoção de recursos legados, melhorias em dicionários e coleções, e novas funcionalidades como o operador `|` para união de tipos, (\_\_\_\_, 2025).

Python 3.10 (2021): Introdução de melhorias significativas na sintaxe, como o *pattern matching* (semelhante ao *switch/case*), e novas ferramentas de tipagem estática, (\_\_\_\_, 2025).

Python 3.11 (2022): Aumento do desempenho, com melhorias significativas na execução do código e otimização do interpretador, além de novos recursos de depuração, (\_\_\_\_, 2025).

Python 3.12 (2023): Melhor desempenho e novos recursos, como melhorias no sistema de tipos e refinamentos na sintaxe, além da continuidade no aprimoramento da performance da linguagem, (\_\_\_\_, 2025).

Python 3.13.0 (Outubro de 2024): Introduziu interpretador interativo aprimorado, suporte experimental a execução sem *Global Interpreter Lock* – Bloqueio Global do Interpretador (GIL), e avanços em compilação *Just-In-Time* – Sob Demanda (JIT), (\_\_\_\_, 2025).

Python 3.13.1 (Dezembro de 2024): Atualização de manutenção com quase 400 correções de *bugs* e melhorias, (\_\_\_\_, 2025).

Python 3.13.2 (Fevereiro de 2025): Segunda atualização de manutenção da série 3.13, trazendo quase 250 correções adicionais, melhorias de desempenho e atualizações de segurança. Essa versão resolve diversos problemas encontrados nas versões anteriores e aprimora a estabilidade geral do Python, tornando-o mais confiável para desenvolvedores e usuários. A atualização também inclui ajustes em bibliotecas padrão e melhorias em compatibilidade com plataformas e sistemas operacionais mais recentes. Versão mais recente, (\_\_\_\_, 2025),

## 1.5 – QT

O Qt é uma estrutura de desenvolvimento de aplicativos multiplataforma para *desktop*, incorporado e móvel. As plataformas suportadas incluem Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS e outros. Qt não é uma linguagem de programação por conta própria. É uma estrutura escrita em C++. Um pré-processador, o MOC (*Meta-Object Compiler*)<sup>1</sup>, é usado para ampliar o idioma C++ com recursos como *Signals* e *Slots*. Antes da etapa de compilação, o MOC analisa os arquivos de origem escritos em C++ estendido e gera fontes C++ compatíveis com o padrão deles. Assim, o próprio *framework* e as aplicações / bibliotecas que o utilizam podem ser compiladas por qualquer compilador C++ compatível com padrões como *Clang*, *GCC*<sup>2</sup>, *ICC*<sup>3</sup>, *MinGW*<sup>4</sup> e *MSVC*<sup>5</sup>, (QT COMPANY, 2025)

O desenvolvimento do Qt começou em 1990 pelos programadores noruegueses Eirik Chambe-Eng e Haavard Nord. A sua empresa, a Trolltech, que vendeu as licenças do Qt e forneceu suporte, foi alvo de várias aquisições ao longo dos anos. Atualmente, a antiga Trolltech é conhecida como The Qt Company e é uma subsidiária integral da Digia Plc, na Finlândia, (\_\_\_\_, 2025).

Embora a Qt Company seja a principal fonte do Qt, este é atualmente desenvolvido por uma aliança mais vasta: o projeto Qt. Esta aliança é constituída por muitas empresas e indivíduos de todo o mundo e segue um modelo de governação meritocrático, (\_\_\_\_, 2025).

### Versões:

Qt 1.0 (24 de setembro de 1996): Primeira versão estável do *framework*, estabelecendo as bases para o desenvolvimento de aplicações multiplataforma, (\_\_\_\_, 2025).

Qt 2.0 (26 de junho de 1999): Introduziu a licença *Q Public License* – Licença Pública Qt (QPL), tornando o Qt disponível como *software* livre, (\_\_\_\_, 2025)

<sup>1</sup> MOC – *Meta-Object Compiler*

Pré-processador do Qt que estende o C++ com funcionalidades como sinais e *slots*, essenciais para a programação orientada a eventos no *framework*.

<sup>2</sup> GCC – *GNU Compiler Collection*

Conjunto de compiladores mantido pelo projeto GNU, amplamente usado para compilar programas em C, C++ e outras linguagens.

<sup>3</sup> ICC – *Intel C++ Compiler*

Compilador da Intel otimizado para alto desempenho em processadores da própria marca, com foco em aplicações exigentes como computação científica.

<sup>4</sup> MinGW – *Minimalist GNU for Windows*

Versão do GCC adaptada para Windows, fornecendo ferramentas GNU para desenvolvimento nativo nesse sistema operacional.

<sup>5</sup> MSVC – *Microsoft Visual C++*

Compilador C/C++ da Microsoft, parte integrante do ambiente de desenvolvimento Visual Studio.

Qt 3.0 (16 de outubro de 2001): Melhorou o suporte a fontes estrangeiras, permitindo a combinação de fontes para preencher lacunas onde a fonte principal não possuía glifos, facilitando a renderização de textos em diferentes idiomas, (\_\_\_\_, 2025).

Qt 4.0 (28 de junho de 2005): Introduziu novas tecnologias, incluindo: Tulip, Conjunto de classes de contêiner de *templates*; Interview, Arquitetura *model-view* para exibição de itens; Radek P; Arthur, Novo *framework* de pintura 2D; Scribe, Renderizador de texto Unicode com *Application Programming Interface* - Interface de Programação de Aplicações (API) pública para *layout* de texto de baixo nível; MainWindow, Arquitetura moderna baseada em ações para janelas principais, barras de ferramentas, menus e acoplamento; Novo Qt Designer, Ferramenta de *design* de interface de usuário, (\_\_\_\_, 2025).

Qt 5.0 (19 de dezembro de 2012): Focou em melhorias de desempenho e facilidade de desenvolvimento de interfaces, com destaque para: Suporte aprimorado a JavaScript e *Qt Modeling Language* – Linguagem de modelagem Qt (QML); QTHub; Introdução do Qt Quick para desenvolvimento rápido de interfaces; Integração com o Chromium através do Qt WebEngine, (\_\_\_\_, 2025).

Qt 6.0 (8 de dezembro de 2020): Revisão significativa na arquitetura do *framework*, notando: Adoção do C++17; Nova arquitetura gráfica com suporte aprimorado a APIs modernas como Vulkan; Unificação de 2D e 3D no Qt Quick; Transição para o sistema de build CMake, (\_\_\_\_, 2025).

Qt 6.9 (2 de abril de 2025): A última versão estável disponível traz inovações como a renderização de emojis atualizada, que melhora a sua exibição nas aplicações, a transparência independente da ordem no Qt Quick 3D, que melhora a renderização de objetos 3D transparentes, e melhorias no suporte ao OAuth2, que responde aos requisitos modernos de autenticação. Há também novos recursos de desempenho em todas as plataformas, que beneficiam os desenvolvedores de aplicações e criadores de dispositivos, (\_\_\_\_, 2025)

### 1.5.1 - Qt Widgets Designer

Qt Widgets Designer é a ferramenta Qt para projetar e construir *Graphic User Interfaces* - Interfaces Gráficas de Usuário (GUIs). É possível compor e personalizar suas janelas ou diálogos de uma maneira *What You See Is What You Get* – O Que Você Vê É O Que Você Obtém (*WYSIWYG*) e testá-los usando diferentes estilos e resoluções. *Widgets* e formulários criados com o Qt Widgets Designer integram-se perfeitamente com o código programado, usando o mecanismo de sinais e *slots* do Qt, para que se possa facilmente atribuir comportamento a elementos gráficos, (QT GROUP, 2025).

Todas as propriedades definidas no Qt Widgets Designer podem ser alteradas dinamicamente dentro do código. Além disso, recursos como promoção de *widgets* e *plugins* personalizados permitem que o usuário use seus próprios componentes com o Qt Widgets Designer, (\_\_\_\_, 2025).

O Qt Designer é uma Qt que fornece uma interface de usuário *WYSIWYG*. Com esta ferramenta, você cria componentes gráficos arrastando e soltando *widgets* em um formulário vazio. Depois disso, você pode organizá-los em uma interface coerente usando diferentes gerenciadores de *layout*. O Qt Designer também permite que você visualize suas interfaces e componentes usando diferentes estilos e resoluções, conecte sinais e slots, crie menus e barras de ferramentas e muito mais. O Qt Designer é independente de plataforma e linguagem de programação. Ele não produz código em nenhuma linguagem de programação específica, mas cria arquivos (.ui). Você pode traduzir o conteúdo de um arquivo .ui em código Python com *pyuic6* para PyQt6 ou *pyside6-uic* para o PySide6, que é uma ferramenta de linha de comando que vem com o PyQt6/PySide6. Então você pode usar este código Python para construir suas aplicações com interface. Você também pode ler .ui diretamente e carregar seu conteúdo para gerar a interface associada. (ZEMBRUZKI, 2024).

### 1.5.2 – PyQt5

Qt é um conjunto de bibliotecas C++ multiplataforma que implementam APIs de alto nível para acessar muitos aspectos de sistemas *desktop* e móveis modernos. Isso inclui serviços de localização e posicionamento, multimídia, conectividade NFC<sup>1</sup> e Bluetooth, um navegador da *web* baseado em Chromium, bem como desenvolvimento de UI<sup>2</sup> tradicional, (RIVERBANK COMPUTING LIMITED, 2024).

PyQt5 é um conjunto abrangente de ligações Python para Qt v5. Ele é implementado como mais de 35 módulos de extensão e permite que Python seja usado como uma linguagem de desenvolvimento de aplicativos alternativa para C++ em todas as plataformas suportadas, incluindo iOS e Android, (\_\_\_\_, 2024).

<sup>1</sup> NFC – *Near Field Communication* (Comunicação por Campo de Proximidade)

<sup>2</sup> UI – *User Interface* (Interface de Usuário)

“Existem duas *bindings* para Python disponíveis para o *framework* Qt, PyQt e PySide.PyQt foram lançadas em 1998, mas requerem um licença para projeto comerciais.”, (JARMUL; LAWSON, 2017, p.105).

De acordo com (RIVERBANK COMPUTING LIMITED, 2024) “O PyQt5 também pode ser incorporado em aplicativos baseados em C++ para permitir que os usuários desses aplicativos configurem ou aprimorem a funcionalidade desses aplicativos”.

Apesar da disponibilidade de uma versão mais recente, a PyQt6, opta-se por utilizar a PyQt5 devido à sua maior variedade de opções e à compatibilidade com versões anteriores de tecnologias relacionadas. Permitindo maior flexibilidade e estabilidade para muitos projetos que dependem de versões específicas de bibliotecas e *frameworks*.

### **Versões:**

PyQt5 5.0 (16 de junho de 2013): A primeira versão do PyQt5 introduz suporte inicial para o Qt 5, (DEBIAN, 2025).

PyQt5 5.1 (2 de outubro de 2013): Adicionou suporte completo ao Qt 5.1, incluindo os módulos QtSensors e QtSerialPort, bem como um conjunto quase completo de *bindings* para OpenGL v2.0 e OpenGL ES/2, (\_\_\_\_, 2025).

PyQt5 5.2 (8 de janeiro de 2014): Foi introduzido suporte ao Qt 5.2, que incorpora os módulos QtBluetooth, QtPositioning, QtMacExtras, QtWinExtras e QtX11Extras. A conversão de objetos Python para tipos de coleção do Qt foi melhorada e o depurador QML ficou habilitado automaticamente ao usar a opção *--debug*, (\_\_\_\_, 2025).

PyQt5 5.3 (26 de maio de 2014): Adicionou suporte ao Qt 5.3, incluindo os módulos QtQuickWidgets e QtWebSockets. Foram implementadas as opções *--target-py-version*, *--sysroot* e *--no-tools* no *configure.py*, tendo sido também oferecido suporte à compilação cruzada, especificamente para iOS e Android, (\_\_\_\_, 2025).

PyQt5 5.4 (25 de dezembro de 2014): Forneceu suporte completo ao Qt 5.4, adicionando os módulos QtWebChannel, QtWebEngineWidgets e QtXml., (\_\_\_\_, 2025).

PyQt5 5.5 (17 de julho de 2015): Introduziu suporte completo ao Qt 5.5, incluindo os módulos QtLocation e QtNfc. Adicionou o QProxyStyle ao QtWidgets e o QMacToolBar e o QMacToolBarItem ao QtMacExtras, (\_\_\_\_, 2025).

PyQt5 5.6 (24 de abril de 2016): Ofereceu suporte completo ao Qt 5.6, adicionando o módulo QtWebEngineCore. Implementou, por predefinição, os ficheiros-tipo *Python Enhancement Proposal* – Proposta de Aprimoramento do Python (PEP) 484 e introduziu diversas opções no ficheiro configure.py, como "--import-from", "--no-stubs" e "--stubsdir", (\_\_\_\_, 2025).

PyQt5 5.7 (25 de julho de 2016): Adicionou suporte ao Qt 5.7.0, (\_\_\_\_, 2025).

PyQt5 5.9 (3 de julho de 2017): Forneceu suporte ao Qt 5.9.0 e 5.9.1, melhorou a deteção da destruição de instâncias baseadas em QObject em C++ e tornou as instâncias de QFlags hasháveis, (\_\_\_\_, 2025).

PyQt5 5.10 (23 de janeiro de 2018): Foram introduzidas funcionalidades de suporte ao Qt 5.10.0 e foi adicionado o módulo QtNetworkAuth, (\_\_\_\_, 2025).

PyQt5 5.11 (23 de junho de 2018): Adicionou suporte ao Qt 5.11.0 e 5.11.1, implementou QtCore.Q\_ENUM() e QtCore.Q\_FLAG() e permitiu o uso de enums Python com Q\_ENUM(), (\_\_\_\_, 2025).

PyQt5 5.12 (10 de fevereiro de 2019): Foi ainda fornecido suporte ao Qt 5.12.0 e 5.12.1, incluindo o módulo QtRemoteObjects. Os módulos QtWebEngine, QtWebEngineCore e QtWebEngineWidgets foram movidos para um pacote separado, designado PyQtWebEngine, (\_\_\_\_, 2025).

PyQt5 5.13 (5 de julho de 2019): Adicionou suporte ao Qt 5.13.0 e suporte para SIP v5. , (\_\_\_\_, 2025).

PyQt5 5.14 (18 de dezembro de 2019): Introduziu suporte ao Qt 5.14.0. , (\_\_\_\_, 2025).

PyQt5 5.15 (1 de junho de 2020): Forneceu suporte ao Qt 5.15.0, adicionando o módulo QtQuick3D e suporte para Python v3.8, (\_\_\_\_, 2025).

PyQt5 5.15.1 (12 de setembro de 2020): Foi adicionado o módulo QtTextToSpeech e melhorada a interpretação de None como um QJsonValue nulo, (\_\_\_\_, 2025).

PyQt5 5.15.2 (23 de novembro de 2020): Tornou os sinais vinculados hasháveis e configurou automaticamente a variável de ambiente `QT_MAC_WANTS_LAYER` no macOS, (\_\_\_\_, 2025).

PyQt5 5.15.3 (22 de Fevereiro de 2021): Foram efetuadas melhorias estruturais no projeto, nomeadamente em termos de ajustes nos ficheiros de configuração e organização dos diretórios. Foram efetuados ajustes no `pyproject.toml` e na definição de produtos, estabelecendo uma base para versões futuras do build, (RIVERBANK COMPUTING LIMITED, 2025)

PyQt5 5.15.4 (5 de março de 2021): A `pylupdate5` passou a assumir o codec UTF-8 por padrão e a adotar a versão 2.1 do formato `.ts`. Também foi adicionada a sobrecarga ausente do método `QPdfWriter.setPageSize()`, (\_\_\_\_, 2025).

PyQt5 5.15.5 (12 de outubro de 2021): A geração de código pelo `pyuic` melhorou ao lidar com `QFontDatabase` e removeu `QObject.__getattr__()` da tipagem para evitar o mascaramento de erros em atributos inexistentes, (\_\_\_\_, 2025).

PyQt5 5.15.6 (29 de outubro de 2021): Foi adicionado suporte ao Python 3.10 no processo de serialização (*pickling*) de `QByteArray`. A tipagem também foi ajustada para facilitar a deteção de erros em chamadas incorretas de construtores, (\_\_\_\_, 2025).

PyQt5 5.15.7 (18 de junho de 2022): Os métodos de desenho do `QPainter` foram fatorados para tirar proveito do `sip.array`, melhorando o desempenho e a eficiência no uso de estruturas gráficas do Qt. Adicionou suporte ao membro `QEvent.Type.NativeGesture`, (\_\_\_\_, 2025)

PyQt5 5.15.8 (22 de janeiro de 2023): Atualizou elementos do Qt v5.15 patch-levels e reorganizou a ordem de construção dos módulos, garantindo que os módulos compostos só fossem construídos após as respetivas dependências, (\_\_\_\_, 2025)

PyQt5 5.15.9 (2 de fevereiro de 2023): Revisou o sistema de construção para corrigir a ordem das dependências e melhorou a integração com o sistema `pyproject.toml`, (\_\_\_\_, 2025)

PyQt5 5.15.10 (10 de outubro de 2023): Foram efetuadas atualizações internas para melhorar a compatibilidade com o Python 3.12, incluindo correções na API `PyType_GetDict()` e no manuseamento de dicionários internos para menus personalizados, (\_\_\_\_, 2025)

PyQt5 5.15.11 (13 de julho de 2024): A partir de agora, é necessário o SIP v6.8.6 e o PyQt5-sip v12.15 para garantir a compatibilidade com o Python 3.13 e o Linux ARM. Também foi atualizado o suporte à anotação de tipos (*type hints*) para dar resposta ao PEP 585 e foram efetuadas melhorias nos métodos de rede, como o `QNetworkAccessManager`, (\_\_\_\_, 2025)

## 1.6 – BANCO DE DADOS

Com a evolução da tecnologia e circulação de informações pelo planeta, centenas de milhares de computadores, celulares e outros dispositivos eletrônicos enviando dados um para o outro, surge-se uma necessidade de organizá-los para serem gerenciados, convertidos, analisados e manipulados, com isso nascem os bancos de dados, que são sistemas feitos especialmente com esse propósito.

Claro que nesse primeiro momento de digitalização, os Bancos de Dados não foram criados logo de imediato. A princípio, as fichas (os registros) eram armazenados de maneira bem arcaica. Basicamente, o que acontecia era pegar uma dessas fichas que havia sido digitalizada e colocá-la uma após a outra, dentro de um arquivo sequencial. Isso porque os arquivos eram guardados em cartões magnéticos, ou cartões perfurados, e esses meios de armazenamento eram sequenciais, por exemplo: caso quiséssemos ler o quinto cartão de um arquivo específico, tínhamos que percorrer pelos quatro cartões anteriores que estavam anexados nesse mesmo cartão. As fitas de papel, ou as fitas magnéticas funcionam da mesma maneira, onde se quisermos ler o meio de uma fita, temos que rebobinar ela toda até chegar no conteúdo desejado. Então, os arquivos iniciais não poderiam ser diferentes. Para encontrar um registro específico, era necessário varrer do primeiro registro até o registro desejado. Por conta dessa característica das sequências de registros, esse tipo de arquivo ficou conhecido como arquivos sequenciais, (RIBAS, 2022).

Os bancos de dados tem sua evolução em paralelo com os avanços de tecnologia de armazenamento de informações num geral, a princípio, essas informações eram armazenadas em fichas de papel, documentos entre outras mídias físicas, além de serem organizadas manualmente, (RICARDO, 2006).

Na década de 1970, Edgar F. Codd, um trabalhador da empresa de tecnologia IBM, deu partida para um modelo relacional de dados, que estruturava-os em tabelas interligadas, que facilitava imensamente o acesso às informações. Esse mesmo modelo, no futuro, serviu de base para sistemas gerenciadores de bancos de dados, como Oracle, MySQL e o PostgreSQL, (\_\_\_\_, 2006).

Na década seguintes, os sistemas gerenciadores se consolidaram no mercado, com diversas companhias utilizando deles por serem mais práticos, nos anos 90 com a expansão da rede de *internet*, a demanda por armazenamento de informações disparou, dando surgimento à banco de dados orientados a objetos e bancos de dados NoSQL, que indicam serem capazes de lidar com grandes volumes de dados que não estão organizados como estariam em um sistema de *Structured Query Language* – Linguagem de Consulta Estruturada (SQL), (\_\_\_\_, 2006).

Nos dias de hoje, bancos de dados ainda progredem em sua evolução, incorporando as mais modernas tecnologias como Inteligências Artificiais (IAs) e *Machine Learning* para uma maior otimização, (\_\_\_\_, 2006).

### 1.6.1 – MySQL

O MySQL é um Sistema de Gestão de Bases de Dados (SGBD) relacional de código aberto que utiliza a linguagem SQL para organizar, gerir e manipular dados de forma eficiente. É amplamente utilizado por empresas e programadores em todo o mundo, graças à sua fiabilidade, desempenho e licença aberta, (DIWAN, 2021).

A sua história começou em 1981, quando Michael "Monty" Widenius criou o UNIREG, uma ferramenta que mais tarde serviria de base para o MySQL. Em 1995, foi fundada a empresa MySQL AB na Suécia por Widenius, David Axmark e Allan Larsson, tendo o MySQL sido oficialmente lançado nesse ano. Cinco anos depois, em 2000, o sistema tornou-se de código aberto, o que impulsionou a sua popularidade a nível global, (\_\_\_\_, 2021).

À medida que a empresa crescia, Mårten Mickos assumiu o cargo de CEO em 2001 e, em 2002, a MySQL AB abriu uma sede também nos Estados Unidos. Em 2003, a empresa estabeleceu uma parceria com a SAP, o que levou à criação de funcionalidades específicas para a integração com os sistemas da empresa alemã, (\_\_\_\_, 2021).

Em 2005, foi lançado o MySQL Network e a Oracle adquiriu a Innobase, responsável pelo InnoDB — o mecanismo de armazenamento padrão do MySQL. Três anos depois, em 2008, a Sun Microsystems comprou a MySQL AB. Por fim, em 2009, a Oracle Corporation anunciou a aquisição da Sun Microsystems, tendo o MySQL passado a fazer parte do portfólio da Oracle, (\_\_\_\_, 2021).

### 1.6.2 - Modelagem

A modelagem de dados é uma prática essencial no desenvolvimento de sistemas de informação, na medida em que define como os dados serão organizados, armazenados e acedidos. Existem três tipos principais de modelação de dados: relacional, dimensional e Entidade-Relacionamento (E-R). Cada tipo possui uma estrutura lógica distinta que influencia diretamente a forma como os dados são guardados e manipulados, (SAP, 2025).

O modelo relacional é o mais comum e tradicional. Organiza os dados em tabelas com linhas (tuplas) e colunas (atributos), estabelecendo relações por meio de chaves. As tabelas podem estar ligadas por diferentes tipos de relacionamento, como um para um ou muitos para muitos. Neste modelo, os dados podem ser classificados como medidas (valores numéricos utilizados em cálculos) ou dimensões (informações qualitativas ou categóricas). Esta abordagem proporciona consistência e integridade aos dados, sendo amplamente utilizada em sistemas de bases de dados transacionais, (\_\_\_\_, 2025).

O modelo dimensional é mais flexível e orientado para a análise de dados em contextos empresariais, como as bases de dados. Organiza os dados em torno de uma tabela de factos, que contém dados numéricos, acompanhada de dimensões que fornecem contexto, como a data ou a categoria. Embora este modelo permita uma recuperação mais rápida de informações, a sua estrutura menos rigorosa pode dificultar a integração com outros sistemas, (\_\_\_\_, 2025).

Já o modelo Entidade-Relacionamento representa os dados graficamente, com entidades (representadas por retângulos) e respetivos relacionamentos (linhas que as conectam). Este modelo é útil para mapear conceitos do mundo real e transformá-los numa estrutura lógica que servirá de base para a criação de uma base de dados relacional. As entidades possuem atributos e os relacionamentos podem ter diferentes graus de complexidade, (\_\_\_\_, 2025).

Quanto aos níveis de abstracção da modelação, existem três: conceitual, lógico e físico. O modelo conceitual é uma visão de alto nível que define os principais conjuntos de dados e respetivos fluxos dentro da organização, sem se preocupar com detalhes técnicos. O modelo lógico é mais detalhado que o modelo conceitual, adicionando pormenores como os tipos de dados e as suas relações, mas sem

estabelecer uma ligação direta com tecnologias específicas. Por fim, o modelo físico descreve a implementação concreta do modelo lógico num sistema de base de dados específico, com pormenores técnicos como índices, tabelas e partições, (\_\_\_\_, 2025).

## 2 - ESTUDO DE CASO

Este capítulo tratará a respeito dos estudos realizados pelo grupo para questões em um contexto mais profissional, englobando viabilidades, estudo de casos, entrevistas, formulários e protótipos. No caso do New T.E., o projeto aborda a área de recursos humanos (RH) no mercado de trabalho.

Um processo de integração feito de forma apressada e desorganizada pode deixar de contemplar etapas essenciais, gerando falhas no conhecimento operacional e causando confusão. Os colaboradores acabam sobrecarregados e incertos sobre como suas funções se alinham ao todo da organização. Esse cenário não representa um bom começo, (EDUVEM, 2022).

Cerca de 40% dos empregadores afirmam que a falta de um processo de integração estruturado trouxe impactos negativos para seus negócios, (\_\_\_\_, 2022).

Antigamente, acreditava-se que ao adentrar uma empresa, o funcionário deveria adquirir os seus conhecimentos “na prática”, ao longo da realização das suas atividades. Entretanto, já foi notado pelas grandes companhias que o processo de treinamento e acolhimento de novos empregados possui a capacidade de reduzir custos e cativar o trabalhador. Isso acontece pois é a partir deste pilar que o trabalhador vai conhecer quais são as principais relações culturais da sua empresa, tornando a adaptação ao meio muito mais simples, (LUZ, 2022).

Caso contrário, o funcionário encontrará maiores dificuldades em entender e sincronizar os seus hábitos ao padrão cultural promovido pela empresa. Alguns anos atrás, era comum visualizar esse período de adaptação como algo trabalhoso e desgastante. Porém, todos nós já aprendemos a importância da integração e da realização correta do *onboarding* para os empregadores, principalmente quando estamos nos referindo a custos dos processos de admissão e a evasão de funcionários, (\_\_\_\_, 2022)

O *onboarding* digital trouxe avanços, mas também desafios, como a falta de socialização e interação presencial, o que pode gerar isolamento e dificultar a adaptação à cultura organizacional. Problemas técnicos e a baixa interação impactam o engajamento, tornando o processo impessoal. Treinamentos *online* podem ser difíceis para quem não está familiarizado com tecnologia, e a comunicação assíncrona pode gerar dúvidas. Sem um acompanhamento adequado, há risco de aumento na rotatividade, o que exige um planejamento eficaz para otimizar a transição.

Para solucionar este problema, a proposta do projeto do New T.E. apresenta uma plataforma que abrange esses problemas, com um *chat* e interfaces simples que estabelecem conexões entre os supervisores e os estagiários, permitindo uma maior interação, até mesmo com usuários não familiarizados com um ambiente digital.

## 2.1 – PESQUISA DE VIABILIDADE

A pesquisa de viabilidade se concentra na aquisição de informações pelos membros do grupo para ver a capacidade e possibilidade do projeto em um âmbito mais profissional. Ela aborda quatro tópicos, sendo eles: Operacional, onde são expostos os conhecimentos necessários para o desenvolvimento; Técnica, englobando elementos mais físicos e práticos, como computadores, servidores e *softwares* que serão utilizados; Cronograma, onde é verificada a possibilidade da conclusão do TCC dentro do prazo estipulado e Econômica, em que foram realizados cálculos hipotéticos para caso o projeto se torne uma empresa real.

### 2.1.1 – Viabilidade Operacional

No desenvolvimento do TCC New T.E., a equipe deve adquirir os seguintes conhecimentos para poder realizar o projeto:

Compreensão das metas de um processo de *onboarding* que as empresas usam, assim, será possível adaptar o sistema a um padrão razoável para a clientela, no quesito conteúdo. Alguns artigos, já deixam exemplificar um bom procedimento e quais são as etapas da integração, além de boas atitudes a serem tomadas durante o processo.

Junto disso, há conhecimentos mais técnicos:

**Lógica de Programação:** Utilização do raciocínio lógico para desafios de problemas computacionais por meio de algoritmos e estruturas de controle.

**Uso de Bibliotecas:** Aplicação de recursos prontos guardados nas bibliotecas com o intuito de diminuir tempo de desenvolvimento e ampliar funcionalidades do código.

**Programação Orientada a Objetos:** Paradigma de programação baseado em objetos, que incentiva modularidade, reutilização e estruturação do código.

**Python:** Aprendizado da linguagem para utilização dela e das bibliotecas como o PyQt.

**QtDesigner:** Ferramenta utilizada para a construção da interface do *software*.

**GitHub:** Uma plataforma de hospedagem de código fonte e controle de versões *online*.

### 2.1.2 – Viabilidade Técnica

Para a funcionalidade do projeto, o New T.E. contará com um servidor remoto, hospedado por meio dos serviços da Hostinger. Como o projeto pretende ter um banco de dados que guarde informações de dezenas de funcionários e de empresas, é previsto que ele fique bem pesado, portanto, o armazenamento em um servidor local não seria ideal.

O desenvolvimento será feito pela equipe de programação designada durante as etapas de organização da equipe do TCC.

**Computador ou *Laptop* com acesso à *internet*:** Essencial para rodar o *software* e permitir a comunicação com o banco de dados remoto.

**Python:** Linguagem de programação escolhida para o desenvolvimento.

**PyQt5 e QtDesigner:** Ferramentas utilizadas para criar a interface gráfica do *software*.

**Servidor da Hostinger:** Banco de dados que armazenará as informações do sistema.

Naturalmente, o projeto pode sofrer diversos problemas durante seu funcionamento, e por isso, foram planejadas soluções para alguns deles, que incluem:

**Backups automáticos no servidor Hostinger:** Configurar *backups* regulares no painel e *dashboard* da Hostinger.

**Controle de acesso:** Implementar autenticação forte e limitar permissões do banco para o usuário utilizado pelo *app*.

**Instabilidade no servidor:** Quando se utiliza um banco de dados hospedado na nuvem, a conexão pode sofrer interferências em caso de instabilidade. Nesses casos, são aplicados no código alguns sistemas que permitem a criação rápida de um banco de dados local para evitar a perda de dados, um sistema para reconexão automática e, adicionalmente, um mecanismo que sincroniza o banco local com o da nuvem, garantindo a integridade das informações.

**Instalação acelerada:** É possível transformar a aplicação em um arquivo executável (.exe), que instala o sistema direto da *internet* ou do próprio GitHub

**Sincronização com resolução de conflitos:** Implementar lógica que identifica conflitos entre dados locais e remotos e resolve automaticamente ou via intervenção do usuário.

**Uso de *timestamps*:** Manter registros de quando os dados foram modificados para controlar a versão mais atual.

### 2.1.3 – Viabilidade de Cronograma

O projeto foi iniciado em Março de 2025, com a parte escrita começando em 8 de Abril do mesmo ano.

Durante as entregas das partes além das apresentações de etapas como a revisão sistemática e as pesquisas de viabilidade, o cronograma criado pelo líder do projeto tem sido atualizado constantemente.

Durante os meses de Abril e Maio, antes do servidor da Hostinger ser adquirido, os membros responsáveis pelo desenvolvimento da aplicação já estiveram trabalhando em aspectos como a modelagem do banco de dados, a prototipagem e o *design* da interface inicial do *software*.

Outros membros responsáveis pela programação estiveram realizando pesquisas e estudos a respeito da linguagem Python e de outras tecnologias vitais para a conclusão do projeto.

Tabela 1 - Cronograma do desenvolvimento do TCC

	A	B	C	D	E
1					
2					
3					
4					
	Atividade	Responsável	Estado	Data Início	Data Término
5	Definição Linguagem de Programação	Leo e Gabriel	Concluído	2/7/2025	2/14/2025
6	Estrutura Base Banco de Dados	Gabriel	Concluído	2/7/2025	2/21/2025
7	Entrega do Levantamento de Dados	Todos	Concluído	2/14/2025	2/28/2025
8	Páginas iniciais da Aplicação	Léo	Concluído	2/14/2025	3/4/2025
9	Pesquisa para Slide (Apresentação)	Todos	Concluído	2/14/2025	3/4/2025
10	Entrega dos Iniciais do Slide	Matheus	Concluído	2/14/2025	3/4/2025
11	Revisão das Tarefas	Daniel	Concluído	2/14/2025	3/7/2025
12	Slide para Entrega	Matheus	Concluído	3/7/2025	3/7/2025
13	Protótipo Baixa Fidelidade	Matheus	Concluído	3/7/2025	3/7/2025
14	Início do Relatório	Miguel	Concluído	3/7/2025	3/7/2025
15	Protótipo Média Fidelidade	Matheus	Concluído	3/7/2025	3/14/2025
16	Protótipo Real	Matheus	Concluído	3/7/2025	3/21/2025
17	Personas Relatório	Miguel e Gabriel	Concluído	3/21/2025	4/3/2025
18	Slide para Entrega (Personas)	Matheus	Concluído	3/21/2025	4/4/2025
19	Entrevistas Protótipo	Todos	Concluído	4/25/2025	5/15/2025
20	Forms	Todos	Concluído	4/25/2025	5/15/2025
21	Slide para Entrega (Protótipo/forms)	Matheus	Concluído	4/25/2025	5/16/2025
22	Relatório de Clientes	Gabriel e Léo	Andamento	5/21/2025	6/6/2025
23	Slide para Entrega (Clientes)	Matheus	Andamento	5/21/2025	6/6/2025
24	Revisão das Tarefas	Daniel	Andamento	2/14/2025	6/6/2025
25	Capítulo 1	Miguel	Concluído	2/14/2025	4/11/2025
26	Capítulo 2	Miguel	Andamento	2/14/2025	6/27/2025
27	Capítulo 3	Miguel	Andamento	2/14/2025	9/26/2025
28	Apresentação	Todos	Andamento	2/14/2025	11/21/2025

Fonte: (AUTORIA PRÓPRIA, 2025)

Visível na Tabela 1, respectivamente, da esquerda para a direita, há a coluna Atividade, que indica a etapa a ser concluída. Responsável, onde é exibido o membro do grupo que deverá realizar a tarefa referente à coluna anterior, seguido pela coluna Estado, que indica se está concluído, em andamento, ou atraso. E por último, as colunas de Início e Término das Atividades.

Ao analisar a tabela, verifica-se um bom estado de atividades concluídas, com nenhuma em atraso, deixando o grupo em uma posição confortável para o desenvolvimento e conclusão do TCC sem muitos problemas.

#### 2.1.4 – Viabilidade Econômica

Tabela 2 - Planejamento da viabilidade econômica do projeto

CUSTOS	VALOR MENS	VALOR ANUAL	profissão	cargos	salario	total
<b>CUSTO FIXO</b>	<b>VALOR MENSAL</b>	<b>VALOR ANUAL</b>				
Salário	R\$ 41.500,00	R\$ 498.000,00	Programador Dev	1	R\$ 8.000,00	R\$ 8.000,00
Aluguel	R\$ 1.200,00	R\$ 14.400,00	Programador BD	1	R\$ 7.500,00	R\$ 7.500,00
Internet	R\$ 250,00	R\$ 3.000,00	Desingner	1	R\$ 6.000,00	R\$ 6.000,00
Domínio	R\$ 29,99	R\$ 359,88	Administrador	2	R\$ 10.000,00	R\$ 20.000,00
Telefone	R\$ 1.000,00	R\$ 12.000,00				
<b>CUSTO VARIÁVEL</b>	<b>VALOR MENSAL</b>	<b>VALOR ANUAL</b>				
Energia	R\$ 250,00	R\$ 3.000,00	Energia Mensal (kWh)	500		
Água	R\$ 225,00	R\$ 2.700,00	Água Mensal (m³)	15		
Mercado	R\$ 500,00	R\$ 6.000,00	Água Mensal Preço (M³)	R\$ 15,00		
Marketing	R\$ 5.000,00	R\$ 60.000,00	Mercado	R\$ 500,00		
			Marketing Mensal	R\$ 5.000,00		
<b>CUSTO ÚNICO</b>						
Móveis	R\$ 2.500,00	R\$ 2.500,00				
Eletrônicos	R\$ 35.000,00	R\$ 35.000,00				
<b>TOTAL ÚNICO</b>	<b>TOTAL MENSAL</b>	<b>TOTAL ANUAL</b>				
R\$ 37.500,00	R\$ 49.954,99	R\$ 599.459,88				

Fonte: (AUTORIA PRÓPRIA, 2025)

Na Tabela 2, há a viabilidade econômica do projeto da New T.E. para *onboarding online* de empresas e fábricas é essencial para seu sucesso. Esta seção avalia custos e receitas.

Os custos do projeto abrangem diferentes categorias e foram estimados com base nas necessidades operacionais e de desenvolvimento. No desenvolvimento, incluem infraestrutura tecnológica, *software*, licenciamento e testes necessários para garantir qualidade e segurança.

Na operação, envolvem hospedagem de servidores, suporte técnico, atualizações do sistema e estratégias de *marketing* para aquisição de clientes. A estrutura de custos fixa conta com salários, aluguel, *internet*, domínio, *marketing* e telefone, totalizando aproximadamente R\$ 599.459,88 anuais.

Entre os custos variáveis, estão gastos com energia projetados em 500kWh mensais, consumo de água estimado em 15.000 litros mensais e despesas de mercado. Além disso, há custos únicos como aquisição de móveis e eletrônicos.

A monetização do projeto se baseia em três principais modelos. O primeiro é a assinatura, oferecendo planos mensais ou anuais para as empresas acessarem a plataforma sendo esse o mais comum entre todos. O segundo modelo utiliza a taxa por usuário, cobrando conforme a quantidade de funcionários integrados, sendo eficaz para empresas menores. O terceiro se baseia na oferta de serviços extras, como customizações específicas e suporte *premium*, atendendo a demandas personalizadas dos clientes, utilizando da gamificação para melhores resultados.

Com uma estratégia adequada de controle de despesas, o projeto demonstra viabilidade econômica e potencial lucro, garantindo crescimento e competitividade no mercado.

## **2.2 – TÉCNICA DE LEVANTAMENTO DE REQUISITOS**

A etapa de levantamento de requisitos do projeto servirá para traçar as necessidades dos clientes e usuários do sistema, além da oferta que o projeto propõe.

As técnicas utilizadas aqui foram de entrevistas e formulários / questionários.

### **2.2.1 – Entrevistas**

As entrevistas realizadas pelos membros do grupo abrangem diversos tipos de pessoas, conseqüentemente, uma base variada de respostas pode ser adquirida de empresários, recepcionistas ou estagiários.

Para algumas das perguntas, foi requisitado um exemplo de uma empresa em que o entrevistado trabalhou para coletar a opinião dele ou dela.

As entrevistas sempre são iniciadas com o entrevistador questionando o entrevistado à cerca de seu nome e ocupação, que, caso não forem revelados, estarão como “Anônimo” nas páginas seguintes.

A seguir, se apresentam as perguntas realizadas, acompanhadas da justificativa por trás delas.

**Perguntas realizadas:**

**Você sabe o que é um processo de *onboarding* ou integração de funcionários?**

Busca-se avaliar o conhecimento dos entrevistados sobre o *onboarding* e entender a importância que eles atribuem a esse processo.

**Você já participou de um processo de *onboarding* ou integração em alguma empresa em que trabalhou ou trabalha?**

Identificar quantas pessoas já passaram pelo processo e se as empresas utilizam métodos manuais ou digitais.

**Durante o *onboarding*, você sentiu que absorveu todas as informações necessárias e que não surgiram dúvidas após o treinamento?**

Verificar se o processo foi efetivo no aprendizado e se as dúvidas foram sanadas, considerando que a aplicação oferecerá suporte contínuo mesmo após o *onboarding*.

**Houve o acompanhamento de um gestor ou mentor durante seu período de adaptação?**

Descobrir se a empresa oferece suporte ativo aos novos funcionários ou se estes precisam se adaptar por conta própria.

**A empresa realizou algum tipo de teste ou avaliação para medir o que você aprendeu durante o *onboarding*?**

Avaliar se a empresa monitora o progresso dos novos funcionários e se há mecanismos para identificar pontos de melhoria no processo.

**Em quanto tempo, aproximadamente, você se sentiu totalmente adaptado(a) à empresa?**

Obter um dado quantitativo sobre o tempo médio de adaptação, permitindo comparar a eficiência do processo atual com a proposta de uma solução digital.

**Você acredita que um processo de *onboarding* digital facilitaria sua adaptação? Por quê?**

Medir a aceitação de um sistema digital de *onboarding* e entender se os funcionários percebem benefícios nesse modelo.

**Durante os primeiros meses, você já chegou a pensar em deixar o emprego por não se sentir bem integrado(a)?**

Investigar se a ausência de um *onboarding* eficaz pode gerar insatisfação e alta rotatividade.

**Você acha que uma plataforma *online*, com explicações detalhadas sobre a empresa e suas funções, teria facilitado esse processo de adaptação no exemplo que citou?**

Medir a percepção dos funcionários sobre a utilidade de uma solução digital, mesmo para aqueles que nunca tiveram contato com esse modelo.

**Logo no início, você recebeu uma explicação clara sobre a empresa, suas regras e sua cultura?**

Avaliar se a empresa possui um processo estruturado para apresentar seus valores, regras e cultura aos novos funcionários.

**Houve situações em que você precisou aprender algo por conta própria por não ter recebido treinamento adequado?**

Medir a eficácia do *onboarding* e identificar se há lacunas no treinamento oferecido, evidenciando a necessidade de suporte adicional.

**Primeira Entrevista:**

Entrevistado(a): Valéria

Ocupação: Empresária

**Você sabe o que é um processo de *onboarding* ou integração de funcionários?**

“Eu acho fundamental, né, esse sistema. É. Já trabalhei em algumas empresas maiores, que era exigido esse processo de integração.”

**Você já participou de um processo de *onboarding* ou integração em alguma empresa em que trabalhou ou trabalha?**

“Eu já participei das 2 formas, quando entrei, fiz a integração, e depois, como gestora de pessoas, acompanhei a integração de funcionários e de colaboradores.”

**Durante o *onboarding*, você sentiu que absorveu todas as informações necessárias e que não surgiram dúvidas após o treinamento?**

“Treinamento de qualidade foi, mas é óbvio que na prática surgiu outras dúvidas. É, ajuda bastante, mas o treinamento tem que ser contínuo, né?”

**Houve o acompanhamento de um gestor ou mentor durante seu período de adaptação?**

“Em algumas empresas, sim.”

**A empresa realizou algum tipo de teste ou avaliação para medir o que você aprendeu durante o *onboarding*?**

“Sim, sim, não sei se de forma estruturada e organizada, mas sim, com certeza.”

**Em quanto tempo, aproximadamente, você se sentiu totalmente adaptado(a) à empresa?**

“Eu demorei quase um mês nesse processo de integração e treinamento.”

**Você acredita que um processo de *onboarding* digital facilitaria sua adaptação? Por quê?**

“Nesse exemplo especificamente, não. Porque foi um treinamento de campo, né? A atividade que eu iria exercer, uma atividade de campo, embora eu tivesse alguns cursos que foram digitais, *online*. Mas se fosse só assim, não seria suficiente.”

**Durante os primeiros meses, você já chegou a pensar em deixar o emprego por não se sentir bem integrado(a)?**

“Sim, já teve empresa que isso aconteceu. “

**Você acha que uma plataforma *online*, com explicações detalhadas sobre a empresa e suas funções, teria facilitado esse processo de adaptação no exemplo que citou?**

“Sim, com certeza. “

**Logo no início, você recebeu uma explicação clara sobre a empresa, suas regras e sua cultura?**

“Sim”

**Houve situações em que você precisou aprender algo por conta própria por não ter recebido treinamento adequado?**

“Infelizmente, sim. “

**Segunda Entrevista:**

Entrevistado(a): Iasmin

Ocupação: Recepcionista

**Você sabe o que é um processo de *onboarding* ou integração de funcionários?**

“Sim, são treinamentos.”

**Você já participou de um processo de *onboarding* ou integração em alguma empresa em que trabalhou ou trabalha?**

“Já sim.”

**Durante o *onboarding*, você sentiu que absorveu todas as informações necessárias e que não surgiram dúvidas após o treinamento?**

“Não.”

**Houve o acompanhamento de um gestor ou mentor durante seu período de adaptação?**

“Sim, foi a minha supervisora, Cíntia.”

**A empresa realizou algum tipo de teste ou avaliação para medir o que você aprendeu durante o *onboarding*?**

“Não.”

**Em quanto tempo, aproximadamente, você se sentiu totalmente adaptada à empresa?**

“Mais ou menos uns 6 meses.”

**Você acredita que um processo de *onboarding* digital facilitaria sua adaptação? Por quê?**

“Sim, porque eu sou recepcionista, então acho que sim, porque eu trabalho com computador.”

**Durante os primeiros meses, você já chegou a pensar em deixar o emprego por não se sentir bem integrada?**

“Sim, já.”

**Você acha que uma plataforma *online*, com explicações detalhadas sobre a empresa e suas funções, teria facilitado esse processo de adaptação no exemplo que citou?**

“Sim, com certeza.”

**Logo no início, você recebeu uma explicação clara sobre a empresa, suas regras e sua cultura?**

“Não.”

**Houve situações em que você precisou aprender algo por conta própria por não ter recebido treinamento adequado?**

“Sim.”

### **Terceira Entrevista**

Entrevistado(a): [Anônimo]

Ocupação: [Anônima]

**Você sabe o que é um processo de *onboarding* ou integração de funcionários?**

“Sei, sim.”

**Você já participou de um processo de *onboarding* ou integração em alguma empresa em que trabalhou ou trabalha?**

“Sim, já participei.”

**Durante o *onboarding*, você sentiu que absorveu todas as informações necessárias e que não surgiram dúvidas após o treinamento?**

“Não.”

**Houve o acompanhamento de um gestor ou mentor durante seu período de adaptação?**

“Sim. Houve.”

**A empresa realizou algum tipo de teste ou avaliação para medir o que você aprendeu durante o *onboarding*?**

“Sim.”

**Em quanto tempo, aproximadamente, você se sentiu totalmente adaptado(a) à empresa?**

“Depois de três experiências de adaptação, aproximadamente, 2 a 3 meses, pois não era passado tudo às claras conforme deveria ser em cada reunião *online*.”

**Você acredita que um processo de *onboarding* digital facilitaria sua adaptação? Por quê?**

“Não. Porque na prática é totalmente diferente. Durante o que é explicado em reunião *online*, fica muito vago. Tem coisas no sistema que, na prática, são totalmente diferentes do que é explicado *online*.”

**Durante os primeiros meses, você já chegou a pensar em deixar o emprego por não se sentir bem integrado(a)?**

“Sim.”

**Você acha que uma plataforma *online*, com explicações detalhadas sobre a empresa e suas funções, teria facilitado esse processo de adaptação no exemplo que citou?**

“Não.”

**Logo no início, você recebeu uma explicação clara sobre a empresa, suas regras e sua cultura?**

“Sim.”

**Houve situações em que você precisou aprender algo por conta própria por não ter recebido treinamento adequado?**

“Sim.”

**Quarta Entrevista:**

Entrevistado(a): Camila

Ocupação: Coordenadora de Horários

**Você sabe o que é um processo de *onboarding* ou integração de funcionários?**

“Pra ser bem sincero, não sabia o que era com esse nome. Agora que você explicou, faz sentido. Acho importante sim, porque quando você entra numa empresa sem ninguém te orientar, é meio desesperador. Fica tudo confuso no começo.”

**Você já participou de um processo de *onboarding* ou integração em alguma empresa em que trabalhou ou trabalha?**

“Acho que não. O mais próximo disso foi quando me mostraram a mesa e me deram um computador. O resto eu fui aprendendo na marra, perguntando para as pessoas.”

**Durante o *onboarding*, você sentiu que absorveu todas as informações necessárias e que não surgiram dúvidas após o treinamento?**

[Esta pergunta não se aplica a este caso, pois não houve um processo formal]

**Houve o acompanhamento de um gestor ou mentor durante seu período de adaptação?**

“Como eu disse, só me mostraram a mesa, então foi meio difícil.”

**A empresa realizou algum tipo de teste ou avaliação para medir o que você aprendeu durante o *onboarding*?**

“Não.”

**Em quanto tempo, aproximadamente, você se sentiu totalmente adaptada à empresa?**

“Ah, foram vários meses. Sem acompanhamento, dificulta.”

**Você acredita que um processo de *onboarding* digital facilitaria sua adaptação? Por quê?**

“Ah, sem dúvida. Eu teria aprendido muito mais rápido. Se tivesse uma plataforma com tudo explicadinho, economizava dias tentando entender sozinho.”

**Durante os primeiros meses, você já chegou a pensar em deixar o emprego por não se sentir bem integrada?**

“Sim. Fiquei bem desanimada nas primeiras semanas, porque parecia que ninguém ligava se eu tava aprendendo ou não. Pensei sim em sair.”

**Você acha que uma plataforma *online*, com explicações detalhadas sobre a empresa e suas funções, teria facilitado esse processo de adaptação no exemplo que citou?**

“Sim, com certeza. Seria como ter um manual da empresa, né? Ia me dar mais segurança, principalmente nos primeiros dias.”

**Logo no início, você recebeu uma explicação clara sobre a empresa, suas regras e sua cultura?**

“Nada muito estruturado. Fui descobrindo aos poucos, observando os outros e cometendo alguns erros até entender como as coisas funcionavam.”

**Houve situações em que você precisou aprender algo por conta própria por não ter recebido treinamento adequado?**

“Bastante. A maioria das coisas eu tive que descobrir sozinho ou pedir ajuda pra alguém. Foi bem frustrante às vezes.”

**Quinta Entrevista:**

Entrevistado(a): Lucas

Ocupação: Supervisor de Contratações

**Você sabe o que é um processo de *onboarding* ou integração de funcionários?**

“Sim, sei sim! O *onboarding* é aquele processo de integração, né? Quando a empresa te apresenta o ambiente, a cultura, as ferramentas... Eu acho essencial, porque ajuda a pessoa a se sentir parte da equipe logo no começo, reduz muita insegurança.”

**Você já participou de um processo de *onboarding* ou integração em alguma empresa em que trabalhou ou trabalha?**

“Já sim, na empresa onde eu tô agora. Foi bem estruturado, com cronograma, treinamento, apresentações, e até um mentor designado pra me acompanhar nas primeiras semanas.”

**Durante o *onboarding*, você sentiu que absorveu todas as informações necessárias e que não surgiram dúvidas após o treinamento?**

“No geral, foi bem informativo, mas sim, surgiram dúvidas depois. Nada muito grave, mas seria legal ter um lugar pra consultar essas coisas depois que o treinamento acaba.”

**Houve o acompanhamento de um gestor ou mentor durante seu período de adaptação?**

“Teve sim. Me colocaram em contato com uma mentora, que me acompanhou por algumas semanas. Era ótimo saber que podia contar com ela, principalmente quando batiam aquelas inseguranças.”

**A empresa realizou algum tipo de teste ou avaliação para medir o que você aprendeu durante o *onboarding*?**

“Sim. Alguns treinamentos tinham *quiz* no final e também tive uma conversa com o RH, pra eles entenderem como eu tava me sentindo e o que tava funcionando.”

**Em quanto tempo, aproximadamente, você se sentiu totalmente adaptado à empresa?**

“Em uns quatro meses.”

**Você acredita que um processo de *onboarding* digital facilitaria sua adaptação? Por quê?**

“Com certeza! Ter tudo num só lugar, treinamentos, informações, contato com mentor... facilitaria muito. Às vezes você esquece algo do treinamento e precisa de um lugar confiável pra consultar de novo.”

**Durante os primeiros meses, você já chegou a pensar em deixar o emprego por não se sentir bem integrado?**

“Não, pelo contrário. Me senti bem recebida, e isso aumentou minha confiança logo no começo.”

**Você acha que uma plataforma *online*, com explicações detalhadas sobre a empresa e suas funções, teria facilitado esse processo de adaptação no exemplo que citou?**

“Facilitaria bastante. Mesmo com um bom *onboarding*, ter um material digital pra consulta contínua é ótimo. Evita muita dúvida desnecessária.”

**Logo no início, você recebeu uma explicação clara sobre a empresa, suas regras e sua cultura?**

“Recebi sim. Tiveram apresentações, documentos, vídeos... foi bem claro desde o começo.”

**Houve situações em que você precisou aprender algo por conta própria por não ter recebido treinamento adequado?**

“Algumas vezes sim, mas eram casos específicos. No geral, tive material de apoio e pessoas disponíveis pra ajudar.”

### **Resultados das entrevistas**

Os entrevistados em sua maioria apresentaram respostas parecidas, com algumas divergências quando se trataram dos métodos de avaliações e também sobre as informações que receberam no começo de suas integrações.

É seguro de concluir que, por mais que todos participaram de um processo de *onboarding*, o sistema de integração e o suporte aguardado não foi o suficiente para

conseguirem trabalhar de forma produtiva e inspiradora na empresa sem ao menos pesquisar por conta própria e/ou terminarem um período de até 6 meses.

Com os dados adquiridos, o grupo agora possui uma base de opiniões e afirmações confiáveis acerca dos processos de *onboarding* atuais e como proceder para o desenvolvimento de um projeto e sistema dentro das expectativas dos usuários e clientes.

### 2.2.2 – Formulário

Junto das entrevistas, foi realizado um questionário pelos membros do projeto para adquirir mais informações de forma rápida, com dados quantitativos, em contraste com os obtidos da entrevista.

Com os dados do formulário, será possível traçar exatamente tanto o que os usuários quanto os clientes que comprariam o sistema podem achar.

Houve uma mesclagem de perguntas realizadas na entrevista no formulário, com outras sendo direcionadas para o desenvolvimento do sistema no futuro.

16 pessoas responderam ao formulário, que continha as seguintes 12 perguntas:

Algo a se notar é que as perguntas do formulário começam da numeração 4, isso se deve pelas três perguntas anteriores serem direcionadas a uma permissão para armazenamento de nome e *e-mail* dos que responderam.

**“Você já ouviu falar sobre o termo ‘*onboarding*’ antes de começar esse processo?”**

Esta pergunta busca entender o conhecimento das pessoas acerca do termo *onboarding*, como há certas pessoas que participam de processos de integração, mas não sabem o que é *onboarding*, ainda foi feita uma segunda pergunta, por precaução.

**“Se sim, você sabe o que significa ‘*onboarding*’?”**

A segunda pergunta foi feita com o intuito de expandir a gama de respostas da questão anterior, nesta, busca-se saber se as pessoas têm conhecimento total sobre o processo, apenas uma ideia vaga, ou não sabem nada a respeito.

**“Você já participou de algum tipo de tutorial ou orientação para usar algum aplicativo, site ou plataforma digital para a execução do ‘onboarding’?”**

A terceira pergunta tem o objetivo de traçar quantas das pessoas já se depararam com algum método de integração e *onboarding* semelhante à proposta do projeto.

**“O que você espera de um processo que te ajude a entender como funciona sua nova empresa?”**

Esta foi a pergunta servirá para ver qual o método de aprendizagem preferido pelas pessoas que responderam, para ver como o sistema deve atuar na integração dos usuários.

**“Você sentiu que os processos de *onboarding* (do qual fez parte) foram fáceis de entender?”**

Para refinar a ideia do projeto, a pergunta foi feita para ver o quão eficiente são os processos de *onboarding* atuais.

**“Você acredita que um aplicativo contendo tais informações teria sido útil no seu processo de *onboarding*?”**

Pergunta feita para entender a opinião a respeito de um processo de *onboarding* digital e sua utilidade, retomando o exemplo da dificuldade de compreensão do processo.

**“Qual parte de aprender a usar uma aplicativo novo você acha mais difícil?”**

Aquisição de dados para o desenvolvimento, com um foco em entender as dificuldades das pessoas ao utilizar o programa no futuro.

**“Como você imagina que seria um processo de *onboarding* digital?”**

Ver a tendência atual das expectativas das pessoas ao se tratar de sua introdução aos processos de integração por plataformas digitais.

**“Você acha que ganhar pontos ou recompensas enquanto aprende a usar algo novo ajudaria a te motivar?”**

A pergunta foi feita para saber a opinião das pessoas sobre a possibilidade de gamificação no sistema, obtendo dados sobre a compatibilidade da ideia do TCC com as necessidades dos usuários.

**“Você acha que se o processo de *onboarding* fosse mais interativo e ‘divertido’, você aprenderia mais rápido?”**

Pergunta realizada para refinar a proposta da aplicação de um *onboarding* gameificado, verificando se é atrativo aos clientes e usuários que utilizarão do sistema.

**“Você gostaria de receber *feedbacks* (mensagens ou avisos) sobre o seu progresso durante o *onboarding*?”**

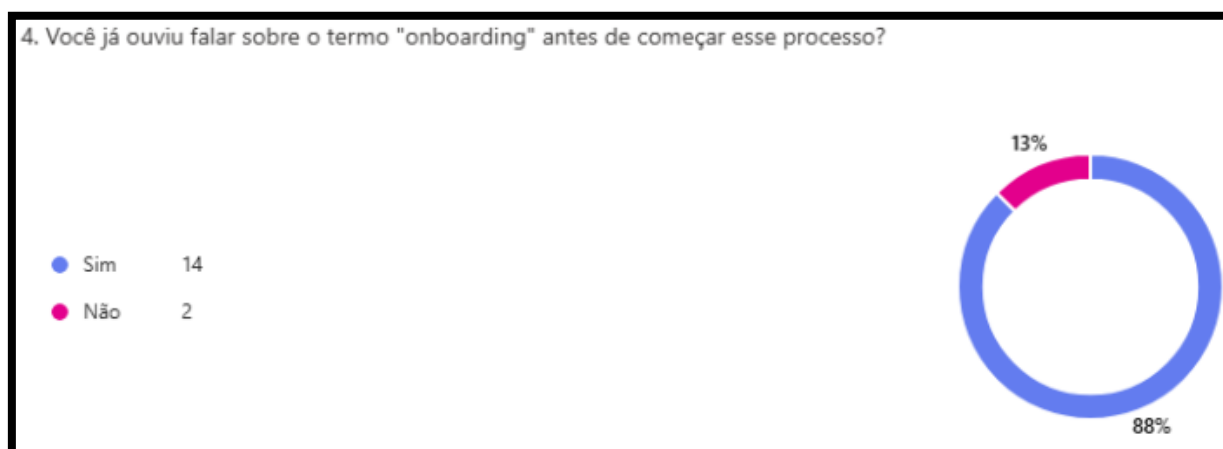
Saber a opinião das respostas quando se trata do rastreamento de seu progresso, num contexto de gamificação ou de integração e aprendizado com as informações de seu cargo em específico.

**“Você acredita que um guia interativo (onde você faz as ações enquanto aprende) seria mais útil do que um guia com apenas textos e vídeos?”**

Pergunta feita com o intuito de balancear a interatividade e a burocracia do sistema, verificando se a ideia está muito difícil, enjoativa, ou desinteressante de um modo geral.

### 2.2.3 Análise do Formulário

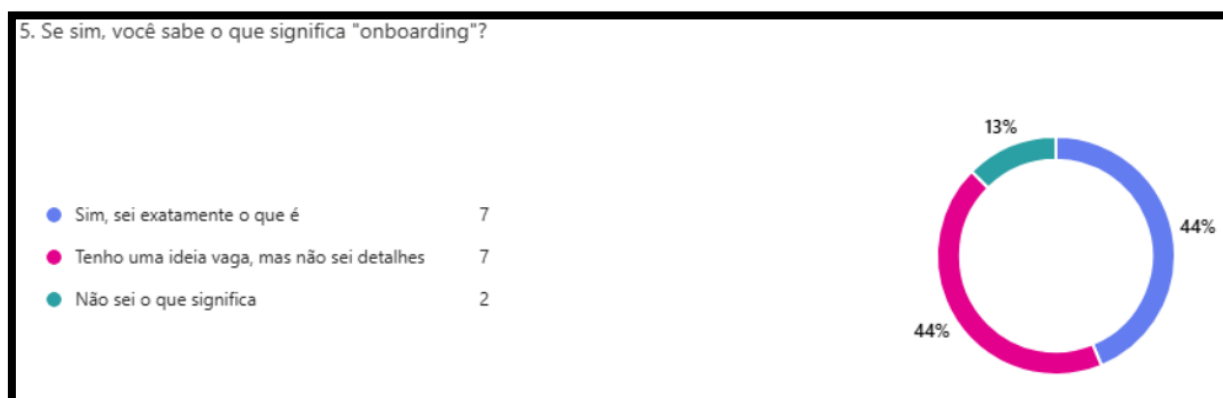
Figura 1 – Pergunta sobre conhecimentos a respeito do termo



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 1 mostra que as respostas aqui tiveram resultados esperados, muitos dos que responderam ao formulário já se encontram no meio profissional, consequentemente, já estão familiarizados com a terminologia.

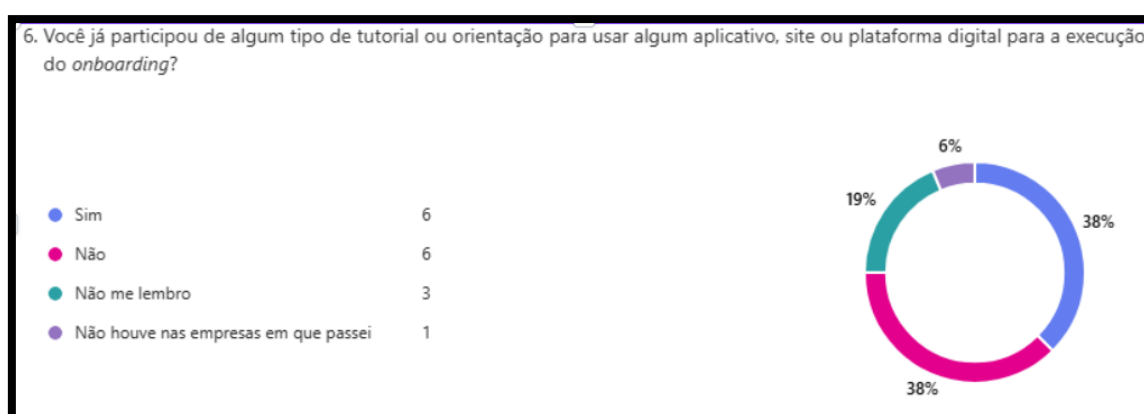
Figura 2 - Especificação sobre o conhecimento de *onboarding*



Fonte: (AUTORIA PRÓPRIA, 2025)

Aqui, na Figura 2, descobre-se que apesar da maioria das pessoas estão familiarizadas com o termo, elas ainda não sabem completamente do que se trata o *onboarding* e integração.

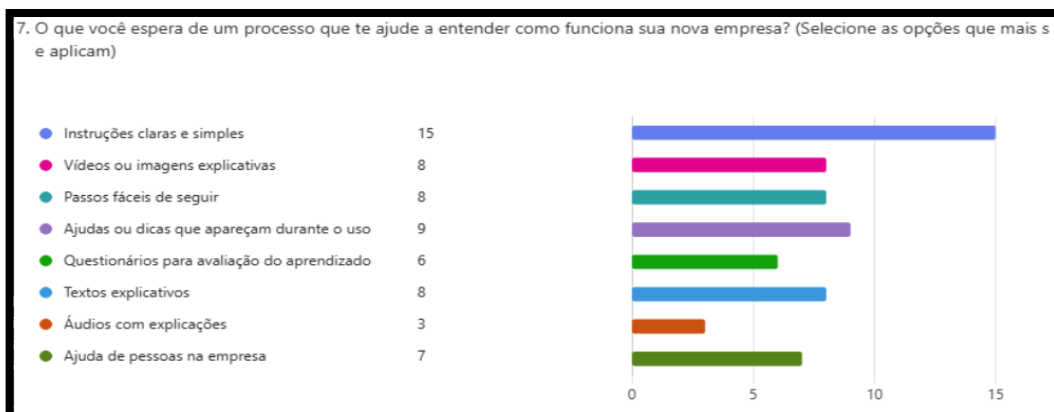
Figura 3 - Realização do *onboarding* de forma digital



Fonte: (AUTORIA PRÓPRIA, 2025)

As respostas estão bem mistas como indica a Figura 3, desconsiderando os que não passaram por processos de *onboarding* de uma forma geral, 38% das pessoas fizeram uma integração por plataforma enquanto outros 38% não fizeram. 19% das pessoas não se lembram.

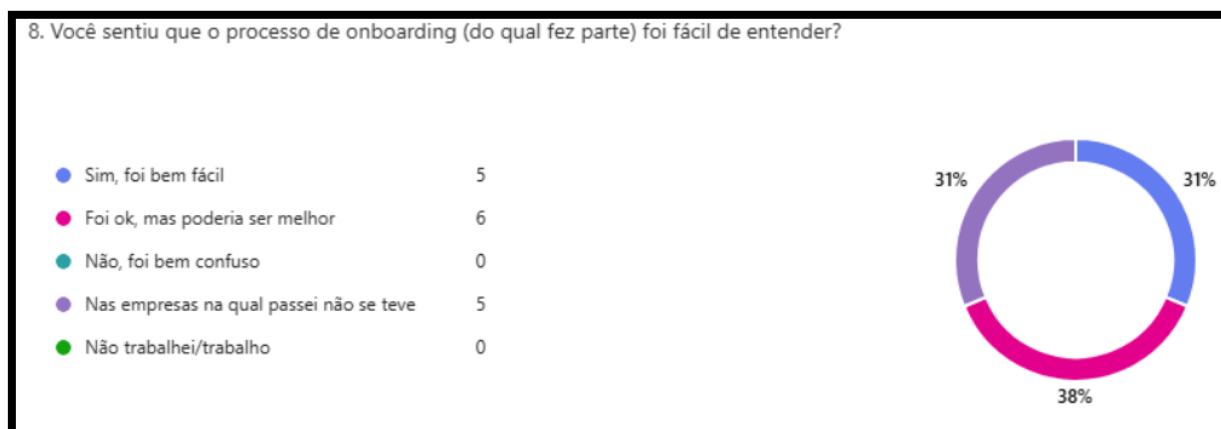
Figura 4 - Expectativas de funcionalidades do sistema



Fonte: (AUTORIA PRÓPRIA, 2025)

Na Figura 4, verifica-se que a maioria das pessoas que responderam ao formulário preferem Instruções claras e simples, desconsiderando aspectos complexos de um sistema como tutoriais longos.

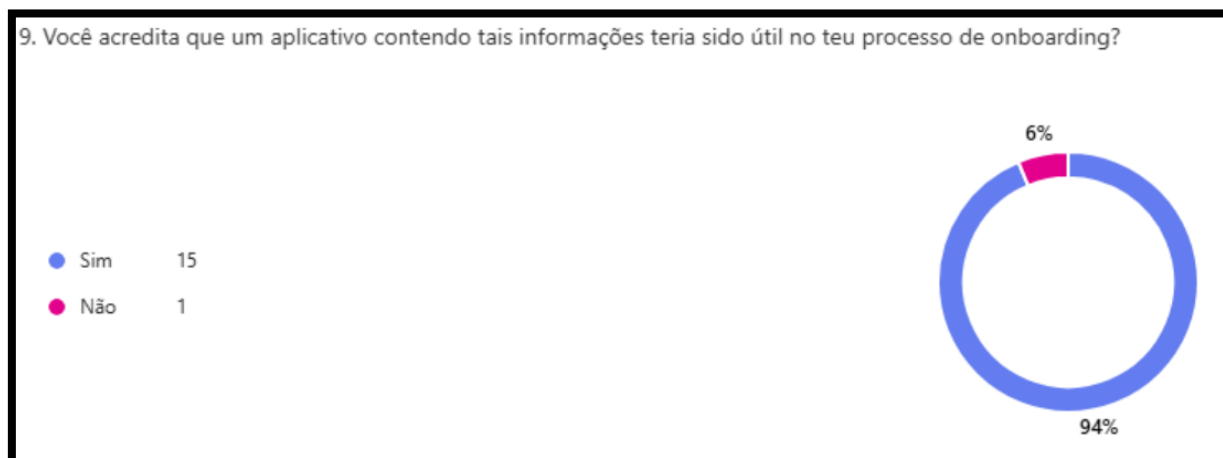
Outras preferências incluem dicas que apareçam durante o uso, como *pop-ups*, textos explicativos, vídeos e imagens.

Figura 5 - Opinião acerca de processos de *onboarding* que já participaram

Fonte: (AUTORIA PRÓPRIA, 2025)

Em uma divisão quase igual, presente na Figura 5, nota-se que 31% ficou satisfeito com o processo, 31% não passou por um processo formal, ou não se lembra e 38% acha que poderia ter sido melhor.

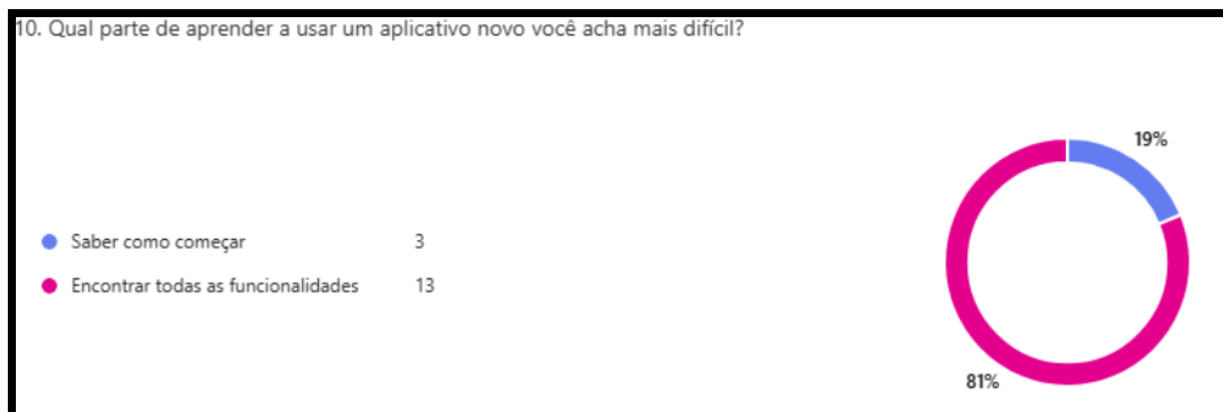
Figura 6 - Utilidade de um sistema que ajude com informações sobre o emprego



Fonte: (AUTORIA PRÓPRIA, 2025)

Na Figura 6, a maioria esmagadora das pessoas que responderam o formulário concorda que um meio digital com as ferramentas para sua integração teria ajudado em seu *onboarding* da pergunta anterior.

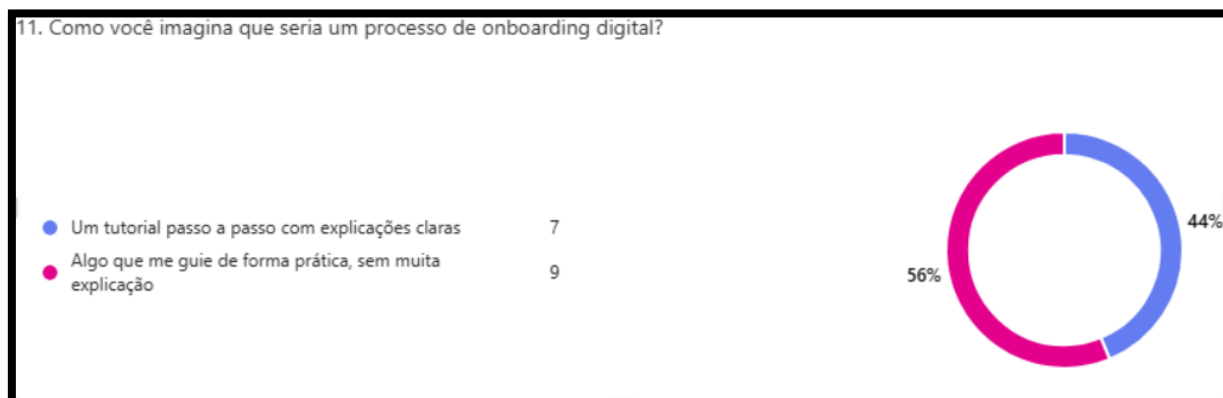
Figura 7 - Dificuldades em usar uma aplicação nova



Fonte: (AUTORIA PRÓPRIA, 2025)

Como mostra a Figura 7, os dados dessa questão serão muito úteis para o desenvolvimento do projeto, já que ajudarão os membros a identificar os principais tipos de usuários. Caso eles já estejam, ou não, familiarizados com o meio digital.

Figura 8 - Expectativa dos usuários para os processos do sistema

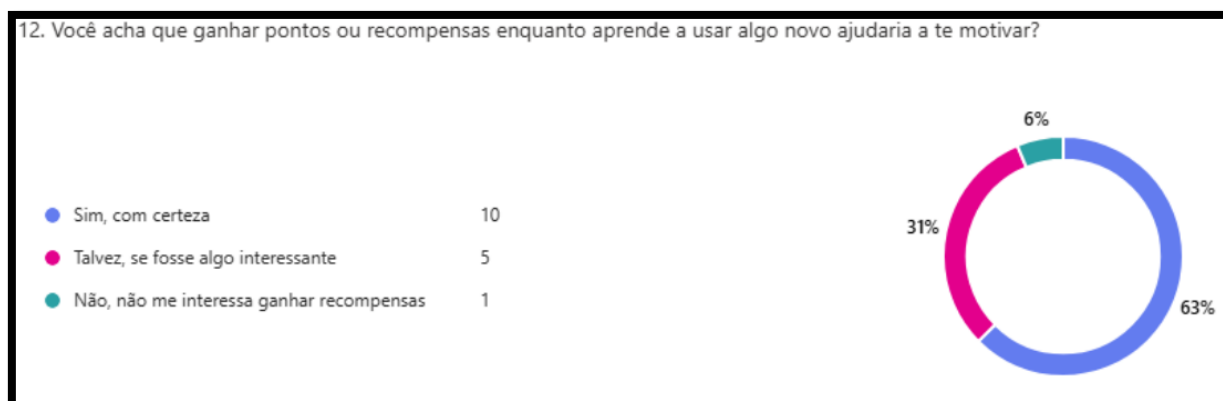


Fonte: (AUTORIA PRÓPRIA, 2025)

O mesmo vale para a questão exposta na Figura 8, neste caso, as respostas estão mais polarizadas, com a maioria aderindo a um processo mais intuitivo e prático, sem muito texto.

Isso retoma ao gráfico de linhas da Figura 4, onde a maioria das pessoas preferem um processo de integração mais fácil de digerir.

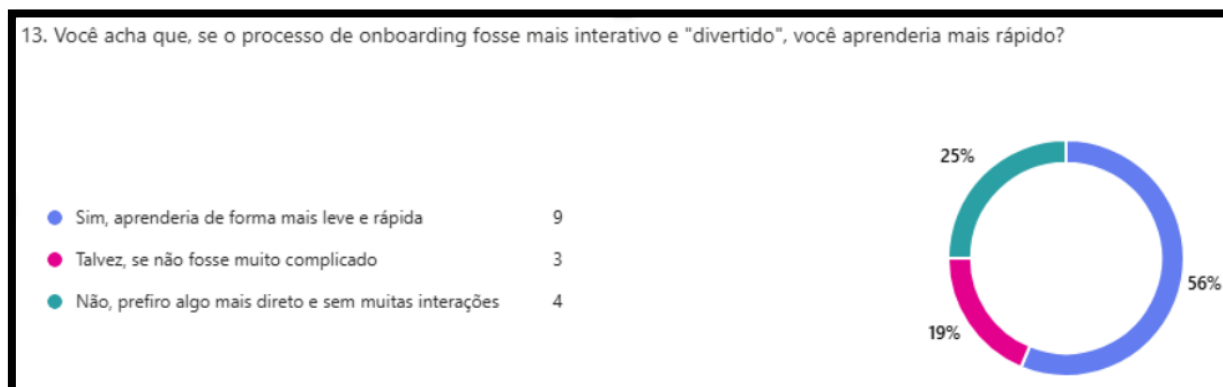
Figura 9 - Pergunta sobre gamificação do sistema



Fonte: (AUTORIA PRÓPRIA, 2025)

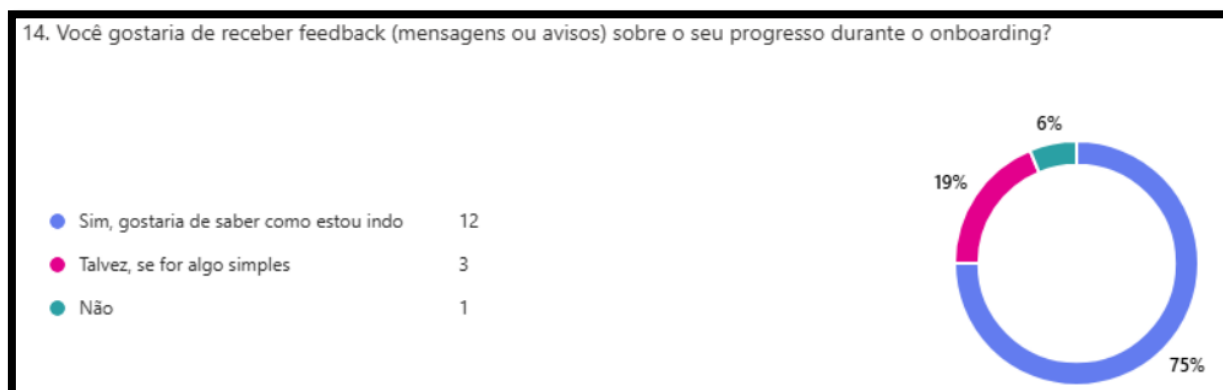
A pergunta presente na Figuras 9 e a próxima estão relacionadas à proposta de gamificação do *onboarding* do projeto, onde a maioria das pessoas concorda que uma compensação pelo estudo do material da empresa e o aprendizado é algo satisfatório.

Figura 10 - Pergunta sobre interatividade do processo



**Fonte:** (AUTORIA PRÓPRIA, 2025)

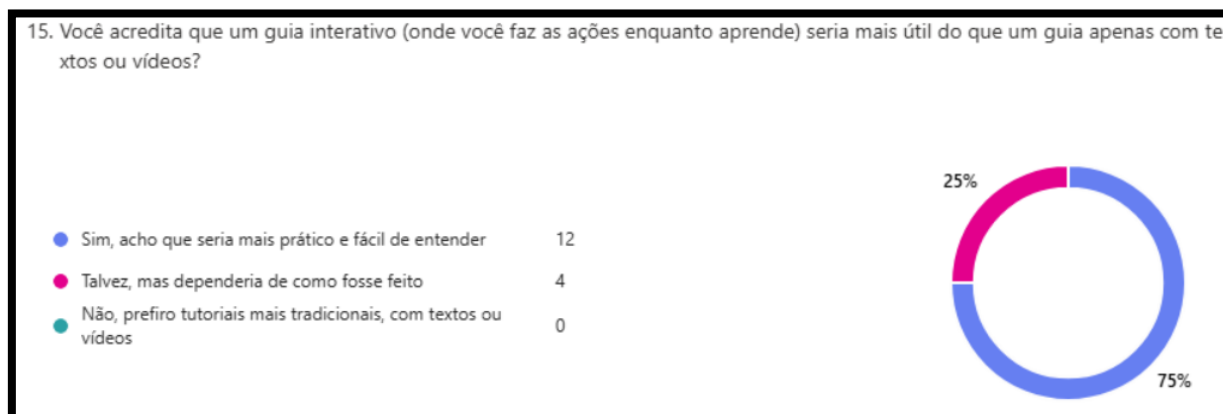
Na Figura 10, com as propostas de questionários e funcionalidades que instigam o funcionário a proceder com seu processo de integração, mais da metade das respostas são positivas, refletindo o interesse de colaboradores e empresários quando esta etapa de contratação deixa de ser tão “burocrática”.

Figura 11 - Opinião sobre *feedbacks* na aplicação

**Fonte:** (AUTORIA PRÓPRIA, 2025)

As respostas na Figura 11 indicam um interesse de receber avisos e notificações para terem um melhor controle de como estão indo, seja um *feedback* eletrônico e padronizado ou uma mensagem de um supervisor.

Figura 12 - Pergunta sobre guia interativo durante uso do sistema



**Fonte:** (AUTORIA PRÓPRIA, 2025)

Aqui, na Figura 12, há uma pesquisa de opinião, onde a maioria concorda em inserir mais meios interativos na aplicação, enquanto 25% acredita que será melhor ver para ter uma resposta concreta.

### **Análise dos formulários**

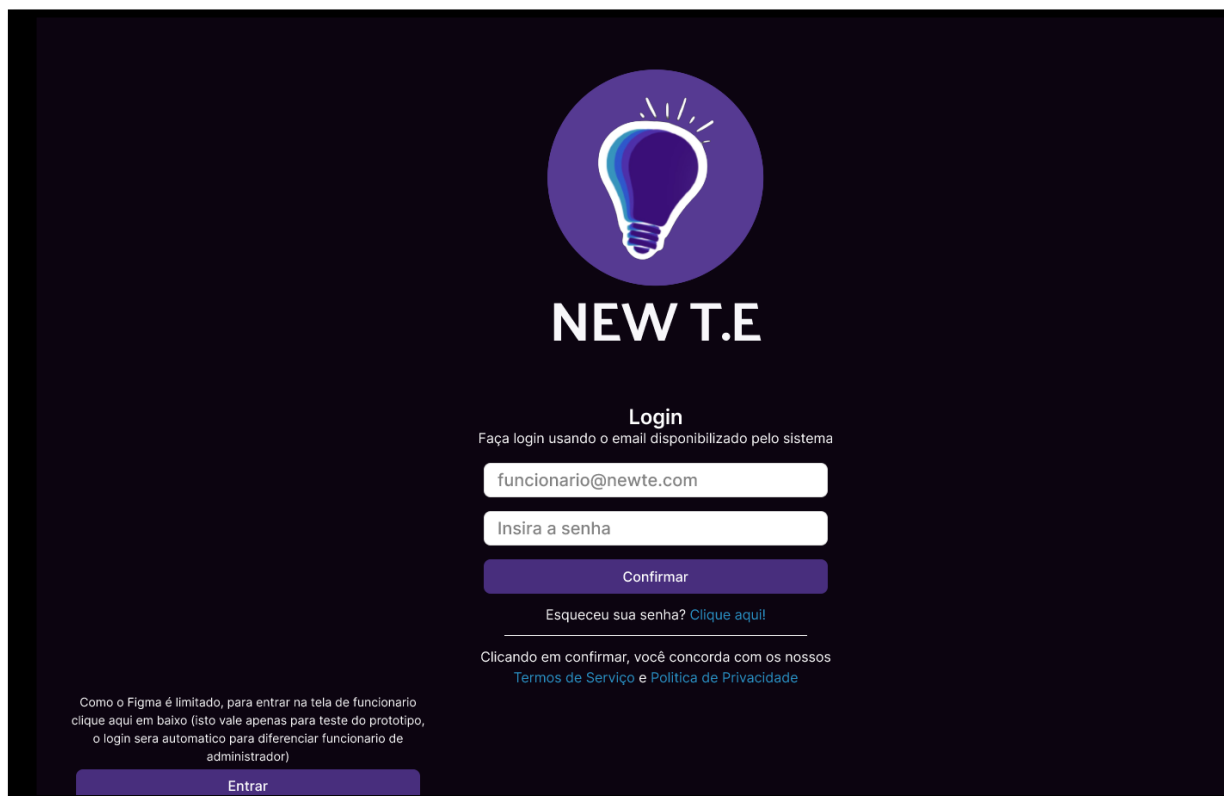
Ao tomar em consideração as respostas do formulário, os membros do projeto conseguiram adquirir um melhor entendimento da opinião de seus usuários sobre as medidas e funcionalidades que devem ser implementadas no sistema, junto de o quão interativo e “gameficado” o sistema deve ser, sem que se torne muito burocrático, entediante ou pouco profissional.

Analisando as respostas como um todo, observa-se um desejo por parte dos usuários de uma plataforma simples e direta, com métodos que gameificação mais simplistas e que evitem a complexidade desnecessária. Além de demonstrarem interesse no projeto, acreditando na melhoria de sua própria integração ao se tomar em conta a proposta do TCC.

### **2.3 – PROTOTIPAGEM**

Para a próxima etapa do projeto, foi desenvolvido um protótipo, que conta com boa parte das funcionalidades e conceitos pesquisados e estudados pelos membros do grupo. As figuras a seguir servem como exemplos de algumas partes do protótipo.

Também foram realizadas outras pesquisas, incluindo um estudo sobre *User Experience* – Experiência de Usuário (UX), e as Heurísticas de Nielsen.

Figura 13 - Tela de *Login* (Protótipo)

**Fonte:** (AUTORIA PRÓPRIA, 2025)

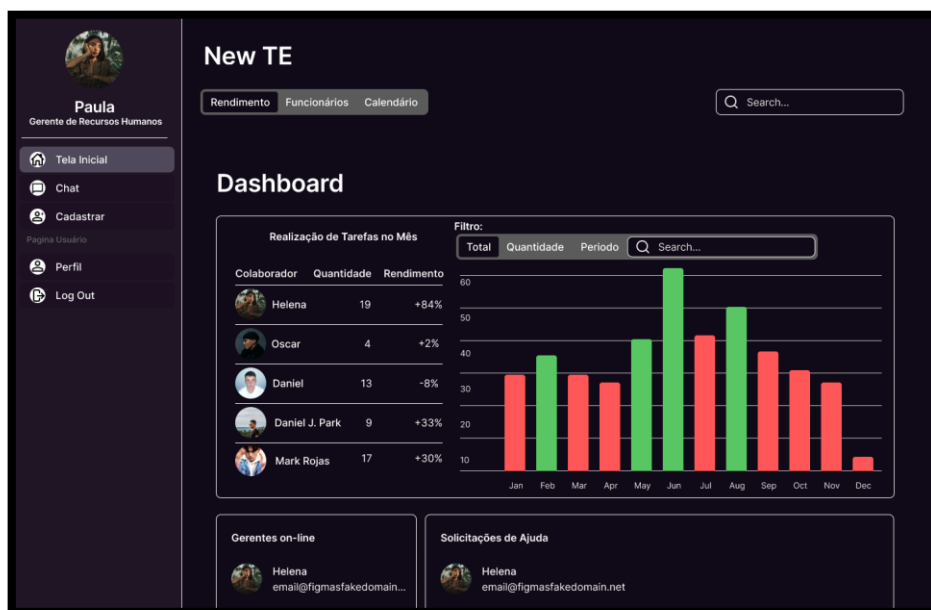
A Figura 13 apresenta a tela de *login* do sistema, a qual é utilizada tanto por funcionários e estagiários quanto pelos supervisores, que também terão acesso completo à plataforma. Esta tela é o ponto de entrada para o uso do sistema e foi desenvolvida com foco na simplicidade e na clareza.

Nela, é possível visualizar duas caixas de texto, destinadas à inserção do *e-mail* e da senha do usuário. Abaixo desses campos, há um botão com texto em azul, posicionado de forma discreta, que direciona o usuário para uma nova tela voltada à recuperação de senha, caso ele tenha esquecido suas credenciais.

Também está presente outros dois textos azuis, que fazem referência aos Termos de Serviço e Política de Privacidade, que serão explorados com mais cuidado nos capítulos seguintes desse TCC.

### 2.3.1 – Interface do Supervisor

Figura 14 - *Dashboard* do Supervisor (Protótipo)



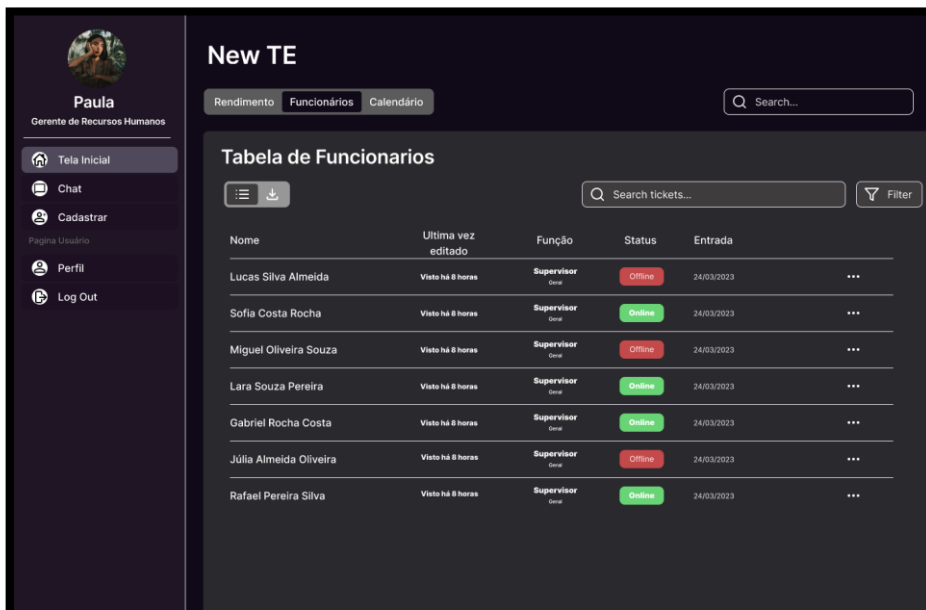
Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 14 mostra a tela inicial, ou *Dashboard* do supervisor. A tela conta com uma barra lateral à esquerda, mostrando outros menus do sistema, como o *Chat*, o local de Cadastro, o Perfil do usuário e outro botão para fazer *Logout*.

Abaixo da palavra “*Dashboard*” é possível ver algumas divisões da tela, que incluem uma tabela da realização das tarefas (relacionadas à gamificação do sistema), com uma lista dos colaboradores da empresa e um filtro para o período em meses e a quantidade de tarefas realizadas. Os gerentes *Online* e as solicitações de ajuda.

No topo da tela também há uma subdivisão da *Dashboard*, que se diverge para outras duas telas. Possuindo outras funcionalidades acessíveis ao supervisor, as telas são para gerenciamento dos colaboradores e funcionários, além de outra para gerenciamento de calendário.

Figura 15 – Tabela de funcionários (Protótipo)



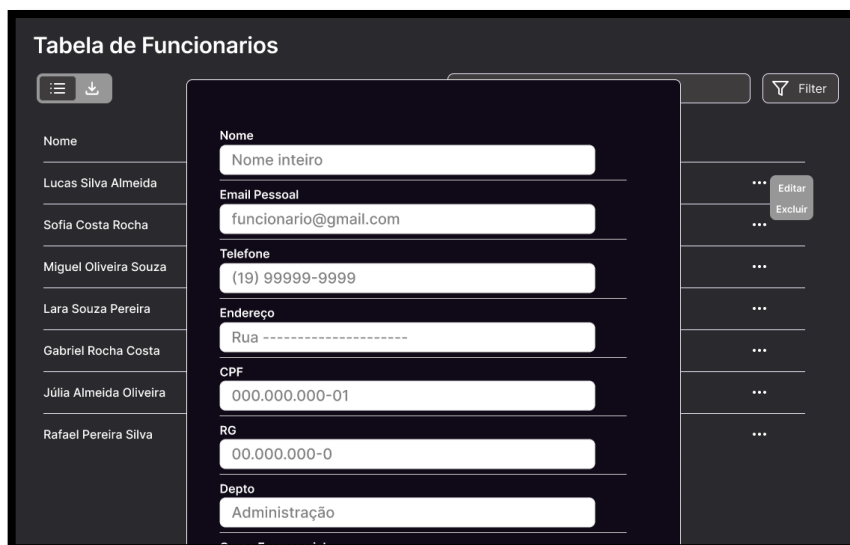
The screenshot shows a user interface for a Human Resources system. The user is Paula, Gerente de Recursos Humanos. The main content area is titled 'New TE' and contains a 'Tabela de Funcionários' (Employee Table). The table has columns for Name, Última vez editado (Last edited), Função (Role), Status, and Entrada (Entry). The data is as follows:

Nome	Última vez editado	Função	Status	Entrada
Lucas Silva Almeida	Visto há 8 horas	Supervisor	Offline	24/03/2023
Sofia Costa Rocha	Visto há 8 horas	Supervisor	Online	24/03/2023
Miguel Oliveira Souza	Visto há 8 horas	Supervisor	Offline	24/03/2023
Lara Souza Pereira	Visto há 8 horas	Supervisor	Online	24/03/2023
Gabriel Rocha Costa	Visto há 8 horas	Supervisor	Online	24/03/2023
Júlia Almeida Oliveira	Visto há 8 horas	Supervisor	Offline	24/03/2023
Rafael Pereira Silva	Visto há 8 horas	Supervisor	Online	24/03/2023

**Fonte:** (AUTORIA PRÓPRIA, 2025)

Nesta seção, mostrada na Figura 15, há uma tabela que mostra os funcionários da empresa, com seu nome, última vez que seu perfil foi editado por um supervisor, a função do colaborador, o status dele no sistema (*Online* e *Offline*), a data de incursão no sistema e um símbolo de três pontos, onde ao serem clicados, abrem as telas de edição e exclusão de dados, como mostrado na figura a seguir.

Figura 16 - Tela de edição de funcionários (Protótipo)



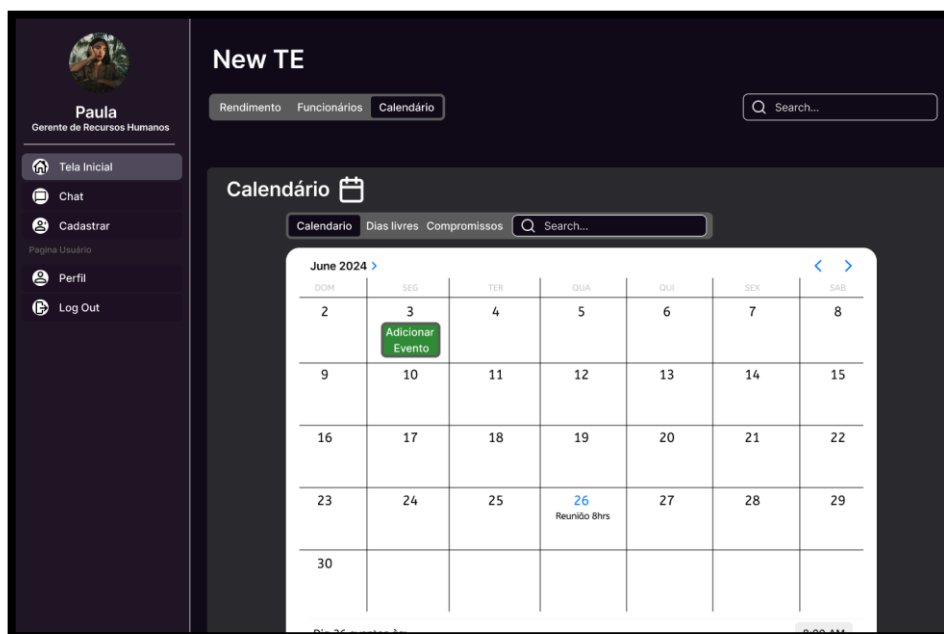
The screenshot shows the 'Tabela de Funcionarios' interface with a modal form for editing employee details. The form includes the following fields:

- Nome: Nome inteiro
- Email Pessoal: funcionario@gmail.com
- Telefone: (19) 99999-9999
- Endereço: Rua -----
- CPF: 000.000.000-01
- RG: 00.000.000-0
- Depto: Administração

**Fonte:** (AUTORIA PRÓPRIA, 2025)

A Figura 16 mostra a tela de edição dos funcionários, onde o supervisor será capaz de atualizar e manipular algumas informações dos usuários com níveis de permissões inferiores, como Nome, *E-mail*, Telefone e seu cargo na empresa perante o sistema.

Figura 17 – Calendário (Protótipo)

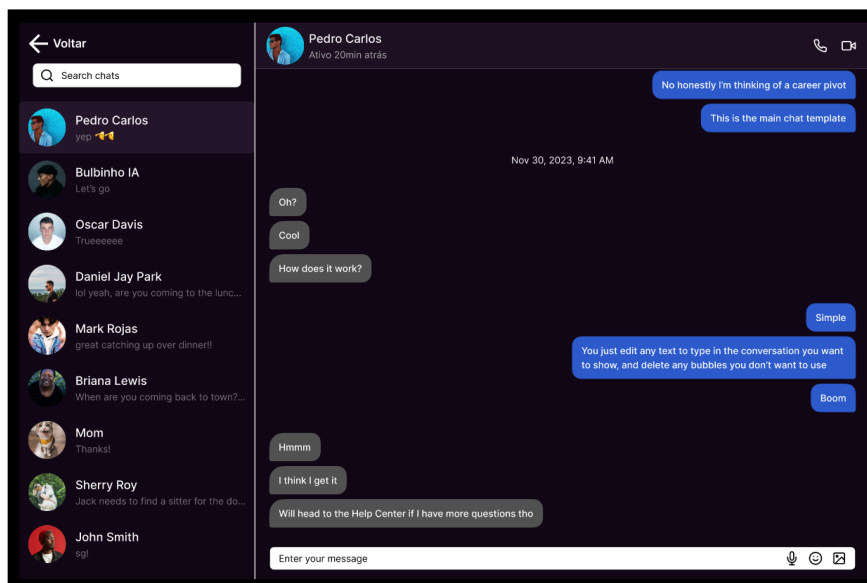


**Fonte:** (AUTORIA PRÓPRIA, 2025)

A Figura 17 ilustra a interface do módulo de calendário do sistema, uma funcionalidade pensada para facilitar o gerenciamento de compromissos e eventos por parte dos supervisores e gerentes. Por meio dessa tela, os usuários conseguem visualizar de forma clara e organizada todos os dias do mês, com destaque para eventos previamente cadastrados, como por exemplo a "Reunião 8hrs" marcada no dia 26 de junho.

Uma funcionalidade prática e intuitiva permite que o usuário adicione novos eventos simplesmente posicionando o cursor sobre a data desejada, o que torna o processo de agendamento mais ágil e eficiente. Além disso, há opções de filtragem na parte superior do calendário, divididas em três categorias: Calendário, Dias livres e Compromissos, possibilitando ao usuário uma visualização mais específica de acordo com sua necessidade no momento.

Figura 18 - Tela de Chat (Protótipo)



Fonte: (AUTORIA PRÓPRIA, 2025)

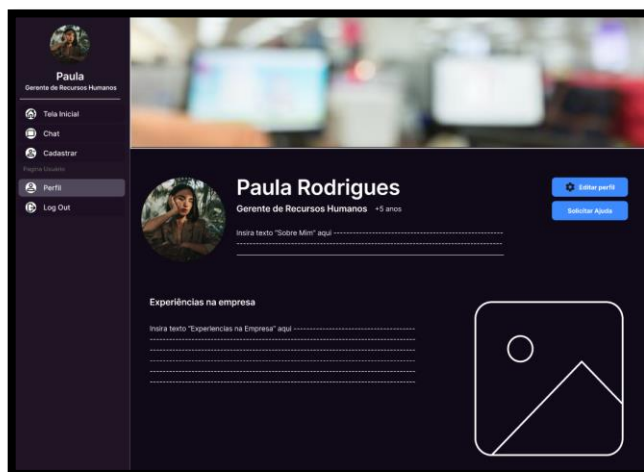
A figura 18 mostra uma tela de *chat* que estará presente tanto no perfil do supervisor quanto do funcionário.

Figura 19 - Tela de Cadastro (Protótipo)

Fonte: (AUTORIA PRÓPRIA, 2025)

A tela de cadastro de funcionários na Figura 19 serve para que novos usuários com nível de permissão de funcionário sejam cadastrados pelos seus supervisores, dando uma maior independência ao sistema do cliente.

Figura 20 - Perfil do supervisor (Protótipo)



Fonte: (AUTORIA PRÓPRIA, 2025)

Aqui, na Figura 20 está o perfil do supervisor, ele conta com o nome, cargo, tempo na empresa, uma seção de “Experiências na empresa” como uma espécie de biografia. E dois botões, um para a edição do perfil, e outro com o botão de solicitar ajuda, que mostrará uma notificação na aba de “solicitações de ajuda”, visível na Figura 14.

### 2.3.2 – Interface do Funcionário

Figura 21 - Tela inicial do funcionário (Protótipo)



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 21 mostra a tela inicial do funcionário, aqui, ele terá acesso às aulas da gameificação, que será expandida para além de vídeos, como mostra a figura. O funcionário poderá baixar e assistir às aulas que serão disponibilizadas pelo sistema.

O funcionário também terá acesso às telas de *chat* e de perfil, que não se diferem das Figuras 18 e 20, respectivamente.

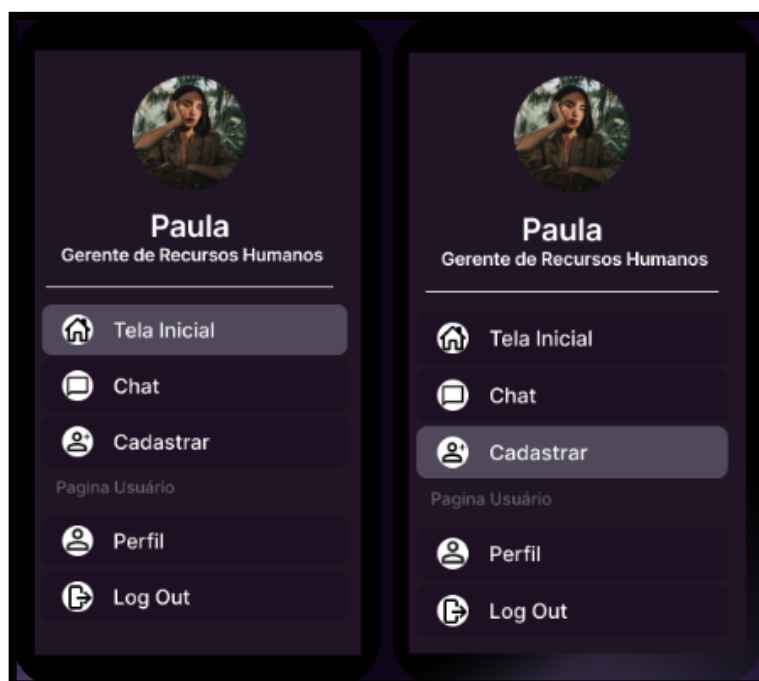
### 2.3.3 – Heurísticas de Nielsen

Neste TCC, foi utilizada a metodologia das Heurísticas de Usabilidade de Jakob Nielsen, um conjunto de princípios que ajudam a identificar falhas comuns em sistemas e aplicativos. A partir da análise de uma interface real, o trabalho propõe melhorias que podem facilitar o uso e tornar a navegação mais intuitiva para todos os tipos de usuários.

A seguir, serão exemplificadas as Heurísticas no protótipo, e como elas funcionam.

#### 2.3.3.1 – Visibilidade do Status do Sistema

Figura 22 - Primeira heurística

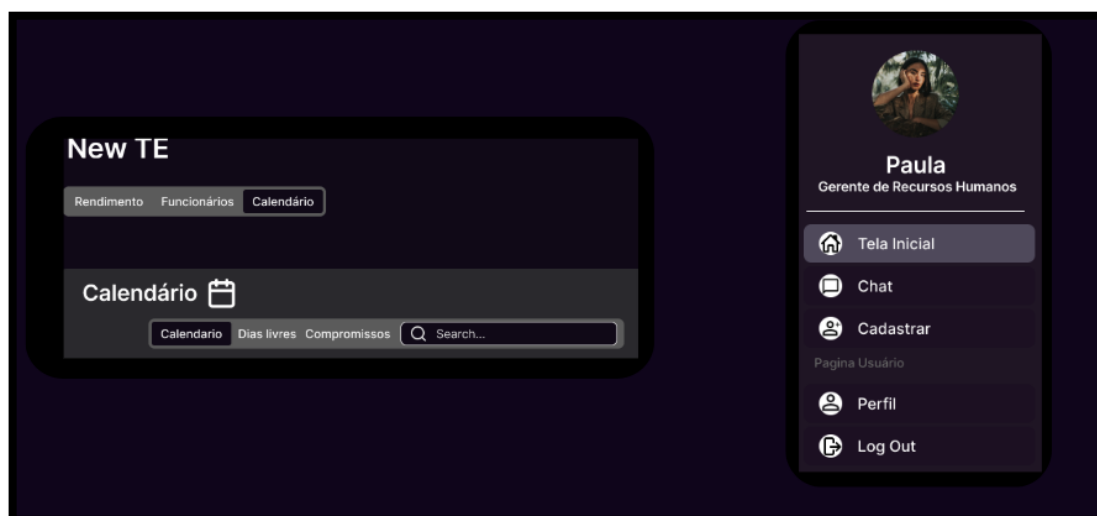


**Fonte:** (AUTORIA PRÓPRIA, 2025)

Como mostra a Figura 22, quando o usuário muda de tela, a categoria a qual ele reside brilha de uma cor mais clara.

### 2.3.3.2 – Compatibilidade do sistema com o mundo real

Figura 23 - Segunda heurística

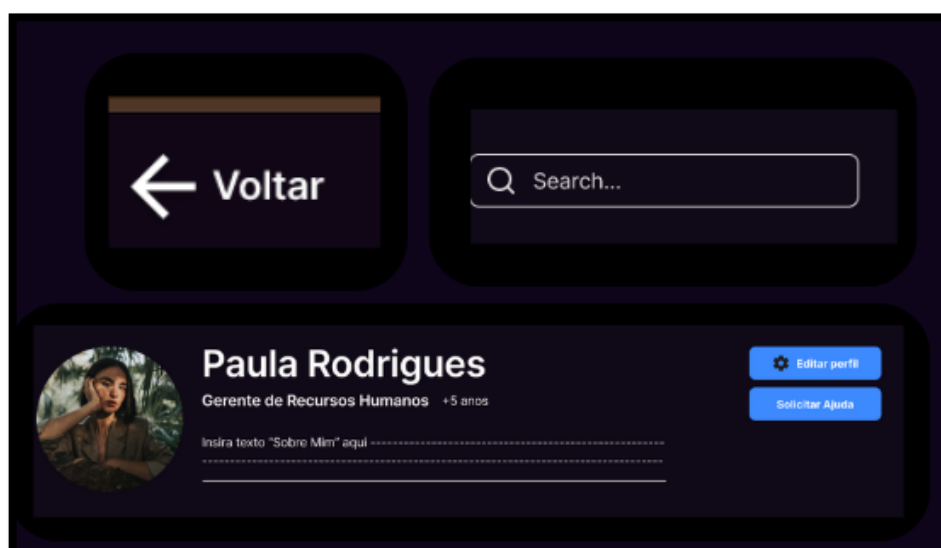


Fonte: (AUTORIA PRÓPRIA, 2025)

Na Figura 23, as opções escritas referem-se exatamente ao que o usuário espere que elas façam, além de ícones que são fáceis de interpretar.

### 2.3.3.3 – Controle e Liberdade do Usuário

Figura 24 - Terceira heurística

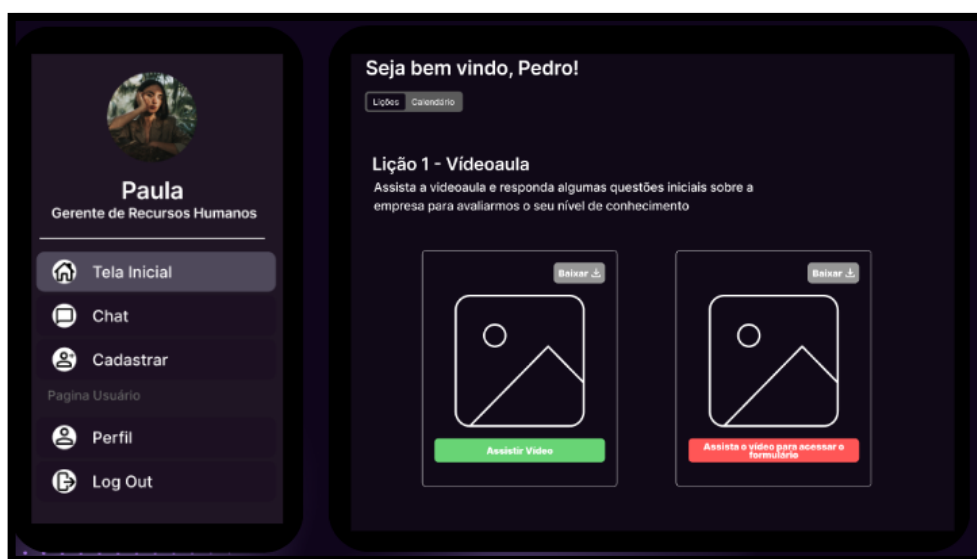


Fonte: (AUTORIA PRÓPRIA, 2025)

Visível na Figura 24, O usuário pode facilmente voltar onde estava caso esqueça de algo ou clique acidentalmente fora do que desejava, por um botão de voltar ou por um método de pesquisa localizado nas páginas. Também há a opção de ele editar o próprio perfil para customização e atualização de dados pessoais.

#### 2.3.3.4 – Consistência e Padrões

Figura 25 - Quarta heurística



Fonte: (AUTORIA PRÓPRIA, 2025)

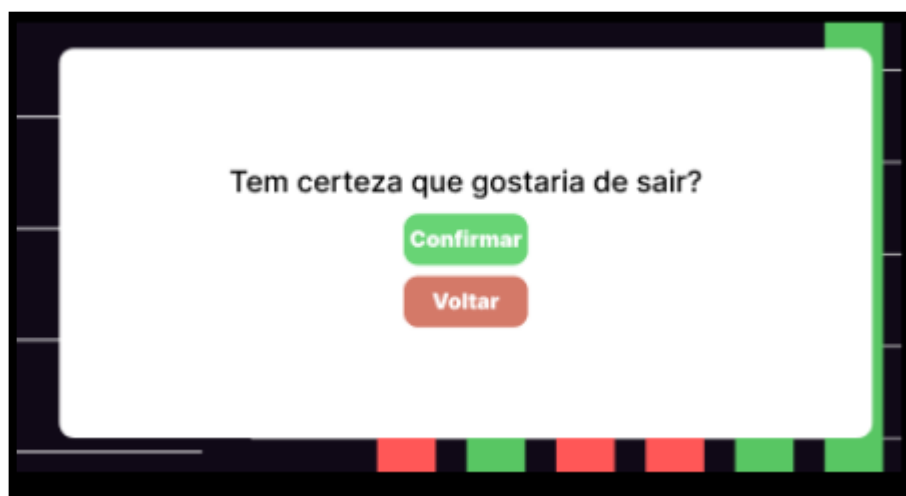
Na Figura 25, o *layout* e o padrão dos botões da aplicação se mantêm constantes conforme cada página, facilitando o entendimento do usuário de saber sempre o que cada botão faz.

Por exemplo, a imagem da esquerda na Figura 25 mostra a barra lateral da aplicação, indicando cada menu com seu símbolo: A tela inicial com uma casinha, o *chat* com um balão de fala, o cadastro com um símbolo de uma pessoa e um “+”.

Na outra imagem da Figura 25 mostra a tela inicial do funcionário, onde os símbolos de *download* podem ficar confusos para alguns usuários, esse problema é resolvido pela palavra “Baixar” estar acompanhando o símbolo.

### 2.3.3.5 – Prevenção de erros

Figura 26 - Quinta heurística

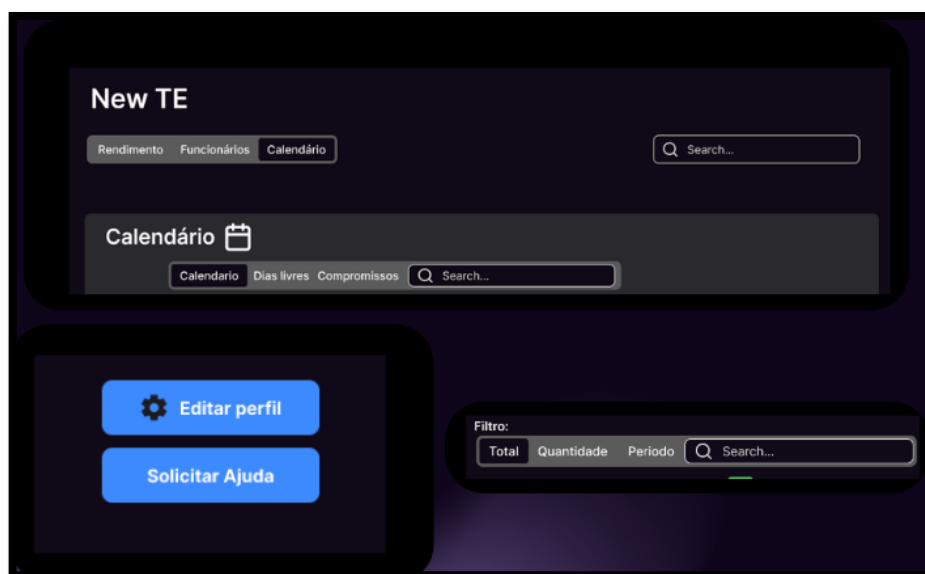


**Fonte:** (AUTORIA PRÓPRIA, 2025)

Quando o usuário clica em log-off, aparece um aviso de confirmação para que ele tenha certeza que quer sair da sua conta. Como mostra a Figura 26, o aviso também ocorre em momentos de exclusão e edições feitas pelo usuário.

### 2.3.3.6 – Reconhecer ao invés de relembrar

Figura 27 - Sexta heurística

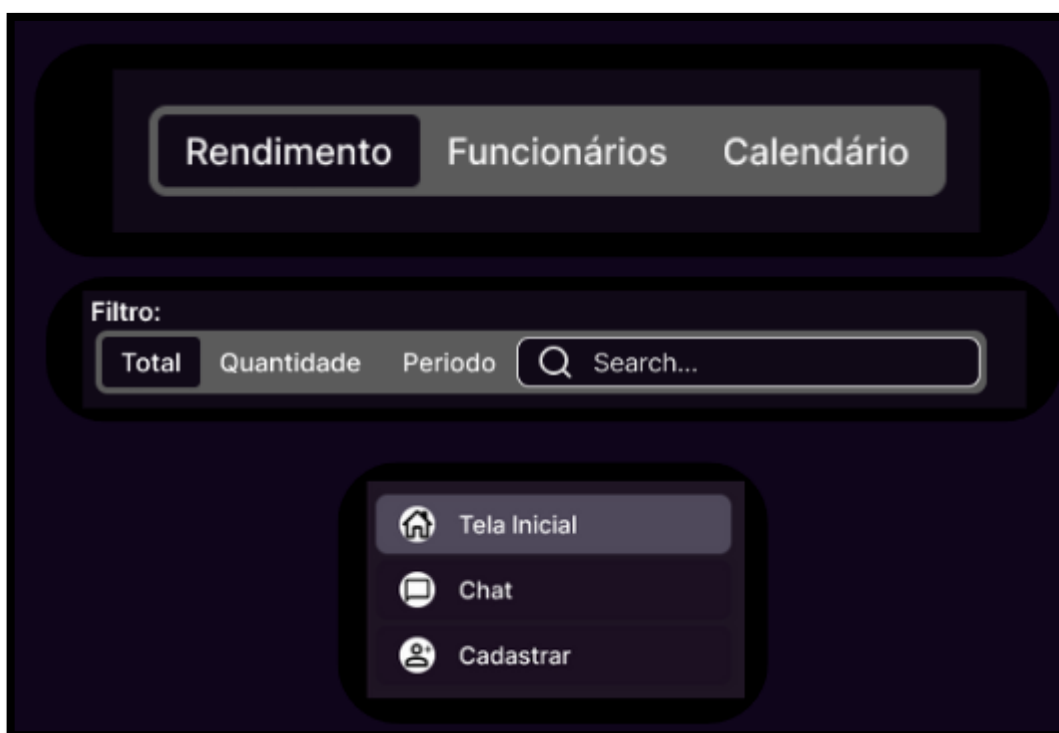


**Fonte:** (AUTORIA PRÓPRIA, 2025)

A aplicação deixa visível e é clara quando se trata da visualização de cada uma das opções necessárias nas suas respectivas categorias. Ao observar a Figura 27, nota-se que não é necessário o usuário ficar decorando onde cada opção se localiza: é apenas clicar e utilizar na hora.

### 2.3.3.7 – Flexibilidade e eficiência de uso

Figura 28 - Sétima heurística



**Fonte:** (AUTORIA PRÓPRIA, 2025)

Como mostra a Figura 28, para usuários que já possuem familiaridade com o ambiente tecnológico e digital, uma das funcionalidades mais valorizadas em qualquer sistema é a implementação de *hotkeys*, ou seja, teclas de atalho. Essas teclas permitem que o usuário execute ações específicas de maneira rápida e eficiente, evitando a necessidade de múltiplos cliques e navegações por menus.

Esse tipo de recurso traz uma enorme vantagem para quem prefere um fluxo de trabalho ágil e otimizado, tornando a interação com o sistema mais fluida e produtiva. Em especial, usuários experientes conseguem navegar por categorias, abas e funções principais simplesmente pressionando teclas numéricas de 1 a 9, ou

utilizando as setas direcionais do teclado para movimentar-se entre as opções disponíveis.

### 2.3.3.8 – Estética e *design* minimalista

Figura 29 - Oitava heurística



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 29 mostra como a aplicação é clara quando se trata de informações: apenas as mais necessárias aparecem no acesso do usuário. Caso haja necessidade de qualquer aprofundamento, terá lugares dedicados à explicação do sistema.

### 2.3.3.9 – Ajudar os Usuários a reconhecer, corrigir e diagnosticar erros

Figura 30 - Nona heurística



Fonte: (AUTORIA PRÓPRIA, 2025)

Presente na Figura 30, a aplicação mostra e disserta erros os quais os usuários possam ter na utilização do sistema, por exemplo, não conseguir assistir a próxima atividade por não ter visto a anterior, ou uma explicação de como a recuperação de senha funciona.

#### 2.3.3.10 – Ajuda e documentação

Embora ainda não esteja implementado no protótipo atual, está prevista a criação de uma página e de uma documentação oficial para orientar tanto os administradores quanto os usuários da aplicação. Essa documentação abrangerá o funcionamento completo do sistema, instruções de uso, boas práticas e resolução de problemas comuns.

A documentação será disponibilizada para *download* pela *internet* e também será referenciada diretamente na interface da aplicação, por meio de uma página dedicada. Adicionalmente, o sistema contará com ícones de ajuda (representados por pontos de interrogação) posicionados ao lado de funcionalidades específicas. Ao passar o cursor sobre esses ícones, o usuário receberá uma breve explicação sobre a finalidade da função e instruções básicas de uso.

## 2.4- CLIENTES

Nesta seção, serão explorados os diferentes tipos de clientes que a New T.E., como empresa, pode encontrar, junto disso, será descrito um perfil dos alvos dos serviços do sistema. Além das 6 subdivisões dos clientes, quem são eles e como tratá-los, acompanhado de uma breve explicação a respeito das consequências esperadas.

Clientes são a base da renda de qualquer empresa, caso não haja um indivíduo (ou outra empresa) que tenha o hábito de consumo de um bem ou serviço, a empresa que atende aos clientes não gera lucro.

Perfil do Cliente

### 2.4.1 – Cliente Interno

**Quem são:** Funcionários da empresa New T.E., que estão diretamente ligados ao desenvolvimento, atendimento e suporte ao produto.

**Importância:** São fundamentais para o bom funcionamento da empresa. Sua motivação e engajamento impactam diretamente na qualidade do atendimento ao cliente final e na reputação da empresa.

**Estratégia de tratamento:** Para manter os funcionários engajados e fidelizados, é necessário oferecer salários ajustáveis e compatíveis com o mercado, planos de carreira bem definidos, ambientes de trabalho confortáveis e flexíveis, benefícios como planos de saúde e alimentação, além de uma cultura organizacional baseada no respeito, comunicação transparente e reconhecimento profissional.

**Resultado esperado:** Aumento da produtividade e da qualidade do trabalho, maior retenção de talentos e clima organizacional mais saudável e colaborativo.

#### 2.4.2 – Cliente Externo

**Quem são:** Serviços e tecnologias das quais o sistema depende, como APIs públicas, hospedagem de banco de dados (Hostinger) e ferramentas utilizadas no desenvolvimento, como PyQt5 e QT Designer.

**Importância:** Esses recursos externos são essenciais para o funcionamento pleno e estável da aplicação. Qualquer falha nesses serviços pode comprometer diretamente a experiência do usuário final, afetando tanto a confiabilidade quanto a funcionalidade do sistema.

**Estratégia de garantia de funcionamento:** Para garantir a continuidade dos serviços, é necessário implementar um monitoramento constante das APIs utilizadas, realizar *backups* frequentes do banco de dados, manter um banco de dados de reserva ativo e utilizar sistemas que detectem automaticamente falhas externas. Além disso, a gestão dos pagamentos relacionados à hospedagem deve ser automatizada ou estritamente regulada por meio de agendas e alertas, a fim de evitar qualquer interrupção por inadimplência.

**Resultado esperado:** Funcionamento contínuo do sistema, redução de falhas críticas e maior confiança dos usuários na estabilidade da aplicação.

#### 2.4.3 – Clientes Ativos

**Quem são:** Empresas que utilizam o sistema de *onboarding* gamificado de forma regular, mantendo uma relação comercial ativa com a New T.E.

**Importância:** Constituem a principal fonte de receita da empresa, além de servirem como base para coleta de *feedbacks*, evolução do produto e promoção da marca por meio de recomendações espontâneas.

**Estratégia de manutenção:** Para mantê-los engajados, o sistema deve oferecer soluções práticas, com usabilidade intuitiva e resultados visíveis, como aumento da produtividade, engajamento de colaboradores e redução do *turnover*. Relatórios claros, suporte eficiente, atualizações constantes e gamificações adaptadas ao perfil do cliente são elementos-chave. A manutenção de preços competitivos também é fundamental para a fidelização.

**Resultado esperado:** Alta taxa de retenção, aumento na renovação de contratos e fortalecimento da relação de confiança entre cliente e empresa.

#### 2.4.4 – Cliente Inativo

**Quem são:** Empresas que já utilizaram o sistema em algum momento, mas que deixaram de utilizá-lo por razões diversas, como falta de adaptação, percepção de pouco retorno ou mudanças internas.

**Importância:** São uma oportunidade de recuperação comercial com menor custo do que a aquisição de novos clientes. Além disso, o retorno dessas empresas pode oferecer *insights* valiosos para melhorias no produto.

**Estratégia reativação:** É necessário identificar os motivos da inatividade por meio de pesquisas ou contatos diretos. A partir disso, podem ser oferecidos tutoriais de reapresentação, novos recursos demonstráveis, planos personalizados e canais diretos de comunicação com o suporte. Atualizações que respondam a críticas anteriores, como melhorias na usabilidade ou na diversidade de atividades, podem ser um diferencial decisivo.

**Resultado esperado:** Reativação de contratos, recuperação de confiança na aplicação e aumento da percepção de valor do sistema por parte do mercado.

#### 2.4.5 – Cliente Satisfeito

**Quem são:** Clientes que utilizam o sistema regularmente e demonstram satisfação com sua funcionalidade, retorno e custo-benefício.

**Importância:** São mais propensos a permanecer com a empresa por longo prazo e a indicar o sistema para outros, contribuindo com o crescimento da marca e com *feedbacks* construtivos para melhorias contínuas.

**Estratégia de manutenção:** Deve-se manter uma comunicação clara, oferecer suporte eficiente e investir em atualizações baseadas nas necessidades reais do cliente. É importante demonstrar que o sistema continua evoluindo e se adaptando às mudanças do mercado. Benefícios adicionais ou conteúdos exclusivos podem reforçar o vínculo com a marca.

**Resultado esperado:** Fidelização de longo prazo, aumento no número de recomendações e fortalecimento da imagem positiva da empresa no setor.

#### **2.4.6 – Cliente Insatisfeito**

**Quem são:** Clientes que, por algum motivo, não estão satisfeitos com a experiência no uso do sistema, seja por questões técnicas, baixa adaptação dos funcionários ou sensação de pouco retorno.

**Importância:** Representam um risco de perda de contratos e de reputação negativa, mas também são essenciais para apontar fragilidades e oportunidades de melhoria no serviço.

**Estratégia de recuperação:** É essencial estabelecer um canal direto de escuta, buscar compreender suas dificuldades e agir com agilidade na resolução. A oferta de tutoriais simples, melhorias no painel de acompanhamento dos colaboradores, variedade nos métodos de avaliação e maior transparência na comunicação são ações que ajudam na reversão da insatisfação. Demonstrar empatia e compromisso com a evolução do sistema reforça a confiabilidade.

**Resultado esperado:** Recuperação da confiança do cliente, possível reconversão para cliente ativo e melhoria contínua da aplicação com base em críticas construtivas.

### **2.5 – ÁREA DE ATUAÇÃO**

A área de atuação representa uma parte geográfica onde a empresa pretende expandir seus negócios, sendo determinada por alguns fatores, como população; viabilidade logística; presença empresarial e compatibilidade de serviços.

Neste TCC, as áreas de atuação da New T.E. serão divididas entre duas principais: Microrregião, com foco inicial na Região Imediata de Mogi Guaçu, e macrorregional, com planos de expansão para toda a Região Sudeste do Brasil.

### 2.5.1 – Microrregião

Em 2017, o Instituto Brasileiro de Geografia e Estatística (IBGE) extinguiu as mesorregiões e microrregiões, implementando uma nova divisão territorial do Brasil, composta pelas Regiões Geográficas Intermediárias e Imediatas (WIKIPÉDIA, 2025).

Mogi Guaçu, que anteriormente integrava a Microrregião de Mogi Mirim, passou a compor a Região Geográfica Imediata de Mogi Guaçu, uma das 53 do estado de São Paulo e uma das onze que integram a Região Geográfica Intermediária de Campinas. Essa nova configuração abrange quatro municípios, com uma população estimada de 326.672 habitantes e área total de 1.903,021 km<sup>2</sup> (\_\_\_\_, 2025).

Essa nova configuração territorial, aliada à infraestrutura logística e à proximidade com centros de inovação como Campinas e Unicamp, favorece o desenvolvimento de soluções digitais voltadas às pequenas e médias empresas da região. Destacam-se, economicamente, os setores industrial, logístico, agrícola e de saúde, que demandam práticas modernas de integração de colaboradores.

A proposta de implementação de uma plataforma de *onboarding* digital gamificado encontra terreno fértil nesse cenário, especialmente considerando o perfil jovem da mão de obra local e o avanço da digitalização empresarial. A solução será adaptada à realidade de cada município, respeitando o porte e o setor predominante de suas empresas.

A estratégia de *marketing* envolve presença regional, parcerias estratégicas, eventos de conscientização e consultorias gratuitas, visando facilitar a transição digital das empresas da região. Estima-se um total de 53.213 empresas na microrregião, sendo entre 5.000 e 7.500 potenciais clientes imediatos, (EMPRESAQUI, 2025).

A meta de adoção inicial é atingir de 200 a 300 empresas no primeiro ano, consolidando a região como referência na modernização de processos de Recursos Humanos.

## 2.5.2 - Macrorregião

A Região Sudeste, composta pelos estados de São Paulo, Minas Gerais, Rio de Janeiro e Espírito Santo, concentra mais de 50% do PIB nacional. Essa região possui também os maiores índices de digitalização e acesso à internet no ambiente corporativo, o que a torna especialmente adequada para a implementação de soluções inovadoras em Recursos Humanos, como o *onboarding* digital gamificado, (BEZERRA, 2025).

A diversidade econômica da região, que abrange setores industriais, comerciais, tecnológicos, financeiros e de serviços, demanda soluções adaptáveis às necessidades de diferentes perfis empresariais. Grandes corporações em São Paulo e Rio de Janeiro requerem plataformas integradas e robustas, enquanto empresas de médio porte em Minas Gerais e Espírito Santo se beneficiam de soluções modulares e escaláveis. *Startups* e empresas de tecnologia, especialmente em polos como Campinas e Belo Horizonte, demandam ferramentas ágeis, focadas no engajamento e performance inicial dos colaboradores.

Para a expansão na macrorregião, a estratégia inclui participação em eventos de RH e tecnologia, como CONARH e Futurecom, além de campanhas digitais segmentadas e parcerias estratégicas com consultorias e instituições locais, como o SEBRAE. A oferta de testes gratuitos e planos flexíveis visa estimular a adoção, especialmente entre pequenas e médias empresas (PMEs).

Segundo estudo do Sebrae, 66% das micro e pequenas empresas brasileiras ainda estão na fase inicial da transformação digital, enquanto apenas 3% são consideradas líderes nesse processo. Embora ferramentas digitais básicas, como banda larga e mídias sociais, sejam amplamente utilizadas, práticas avançadas como *cloud computing*, automação e *e-learning* ainda são pouco exploradas. Esse cenário apresenta grande oportunidade para soluções digitais que auxiliem no avanço da maturidade digital das empresas, alinhando-se às necessidades do mercado da Região Sudeste, (SEBRAE, 2025).

## 2.6 - CONCORRENTES

A análise da concorrência é essencial para compreender o ambiente em que a New T.E. irá atuar, identificando as empresas que oferecem soluções semelhantes no mercado de integração digital com foco em recursos humanos e gamificação. Este estudo permite avaliar os pontos fortes e fracos dos concorrentes, além de destacar os diferenciais da New T.E. para a construção de estratégias eficazes de posicionamento e comunicação.

Neste capítulo, serão apresentados os principais concorrentes diretos, que atuam com produtos semelhantes, e os concorrentes indiretos, que oferecem soluções alternativas para o mesmo público, bem como oportunidades para superar os desafios existentes.

Esta análise fornecerá subsídios para o posicionamento competitivo da New T.E. e para a definição das ações que permitirão fortalecer sua atuação no mercado, garantindo destaque e adequação às necessidades das pequenas e médias empresas em processo de transformação digital.

### 2.6.1 - Concorrentes Diretos

Os concorrentes diretos da New T.E. são empresas que oferecem soluções digitais voltadas especificamente para o processo de integração de novos colaboradores com foco em Recursos Humanos. A maioria já atua com gamificação, automação ou interações digitais otimizadas.

#### 2.6.1.1 - i9Ação

Figura 31 - Logo da I9Ação



Fonte: (I9AÇÃO, 2025)

**Metodologia:** Utiliza gamificação, *microlearning* e personalização de conteúdo para aumentar o engajamento durante o *onboarding*.

**Vantagens da New T.E.:** Plataforma mais leve, com código aberto e alta flexibilidade para personalização, ideal para *startups* e PMEs que buscam soluções acessíveis e escaláveis.

**Desvantagens:** A i9Ação possui ampla base de clientes, soluções já consolidadas e histórico de resultados positivos. Empresas satisfeitas tendem a não trocar de fornecedor.

**Justificativa:** A força da marca e do portfólio da i9Ação torna a conquista de mercado mais difícil para novos entrantes.

**Como a New T.E. pode cobrir essa desvantagem:** Focar em empresas menores, que ainda não possuem soluções estruturadas, e apresentar a proposta como uma solução acessível, flexível e com melhor custo-benefício.

#### 2.6.1.2 - UserGuiding

Figura 32 - Logo da UserGuiding



Fonte: (USERGUIDING, 2025)

**Metodologia:** *Onboarding* interativo dentro de plataformas digitais, com guias visuais, tutoriais e suporte contextual.

**Vantagens da New T.E.:** Foco específico em integração de colaboradores, com estrutura voltada a Recursos Humanos e experiência lúdica interna.

**Desvantagens:** Forte atuação no mercado de *software* como serviço SaaS (*Software* como um Serviço), com integração facilitada em sistemas *web* e aplicações hospedadas na nuvem.

**Justificativa:** Empresas digitalmente maduras podem preferir a integração técnica robusta da UserGuiding.

**Como a New T.E. pode cobrir essa desvantagem:** Investir na personalização humana e cultural do *onboarding*, com foco na integração a ERPs (*Enterprise Resource Planning* – sistemas de planejamento de recursos empresariais) e CRMs (*Customer Relationship Management* – sistemas de gestão de relacionamento com o cliente) nacionais, amplamente utilizados por pequenas empresas.

### 2.6.1.3 - Pipefy

Figura 33 - Logo da Pipefy



Fonte: (PIPEFY, 2025)

Acima, está a logo da Pipefy.

**Metodologia:** Automatização de processos com integração a agendas, *dashboards* e fluxos de trabalho definidos.

**Vantagens da New T.E.:** Experiência de *onboarding* mais envolvente, gamificada e com foco em integração de pessoas, não apenas em processos

**Desvantagens:** A automação de tarefas administrativas do Pipefy pode ser vista como mais produtiva para empresas com estruturas internas complexas

**Justificativa:** Mesmo com menor envolvimento humano, a automação total ainda atrai empresas que priorizam performance de processos

**Como a New T.E. pode cobrir essa desvantagem:** Criar automações essenciais dentro do sistema, mas manter o foco no lado humano da integração, destacando a importância da cultura organizacional e engajamento nos primeiros dias de trabalho.

## 2.6.2 - Concorrentes Indiretos

Os concorrentes indiretos são empresas que atuam em áreas relacionadas ou que atingem o mesmo público-alvo da New T.E., mas oferecem soluções com focos diferentes. Embora não concorram diretamente em todas as funcionalidades, influenciam o mercado e apresentam desafios que devem ser considerados na estratégia da New T.E.

### 2.6.2.1 - Mambo.IO

Figura 34 - Logo da Mambo.IO



Fonte: (MAMBO.IO, 2025)

Acima se encontra a logo da Mambo.IO.

**Metodologia:** Sistema focado em gamificação para campanhas de engajamento empresarial, guiado pela satisfação dos funcionários que providencia seus serviços.

**Vantagens da New T.E.:** Plataforma mais simples, com curva de adoção menor, ideal para empresas que buscam soluções diretas e rápidas. Interessante para pequenas e médias empresas

**Desvantagens:** Menor quantidade de automações comparado ao Mambo.IO. A New T.E. possui um foco maior na usabilidade do sistema pelos usuários

**Justificativa:** Mambo.IO pode atrair clientes que buscam automações robustas para engajamento.

**Como a New T.E. pode cobrir essa desvantagem:** Usar a simplicidade como vantagem, e implementar automações de forma gradual para atender demandas específicas.

### 2.6.2.2 - Wildgoose

Figura 35 - Logo da Wildgoose



Fonte: (WILDGOOSE, 2025)

Acima está a logo da Wildgoose.

**Metodologia:** Realiza atividades de integração e *onboarding* presenciais, como jogos de “caça ao tesouro” empresarial.

**Vantagens da New T.E.:** Solução digital contínua, que não depende de eventos presenciais, podendo ser usada a qualquer momento.

**Desvantagens:** Experiências mais lúdicas e presenciais da Wildgoose podem oferecer maior interação face a face.

**Justificativa:** Empresas que valorizam eventos presenciais podem preferir Wildgoose.

**Como a New T.E. pode cobrir essa desvantagem:** Destacar a eficiência e o baixo custo da plataforma digital, com possibilidade de uso remoto e flexível.

## 2.7 - LEGISLAÇÃO E CONTRATUAL

As leis e regras contratuais são o conjunto de normas que regem a elaboração, execução e extinção dos contratos firmados entre as partes.

Em projetos tecnológicos como o desenvolvimento de um *software* de *onboarding* digital para empresas no modelo SaaS (*Software* como Serviço), a atenção especial deve ser dada a legislações que garantam segurança jurídica, proteção de dados, transparência nas relações comerciais e direitos do consumidor.

No contexto deste projeto, que consiste em um sistema para integração de funcionários com funcionalidades de gamificação por tarefas, comunicação via *chat*, suporte de supervisores, gestão de cadastro de novos funcionários e agendamento de atividades via calendário, o cuidado dessas normas é essencial para a eficácia e legalidade do sistema.

### **2.7.1 - Legislação**

O desenvolvimento do sistema New T.E. de *onboarding* digital, baseado no modelo SaaS depende de normas jurídicas que sustentem sua operação, desde o uso de dados pessoais até a validade de contratos e assinaturas digitais. No contexto municipal, estadual e federal, diversas legislações impactam direta ou indiretamente o projeto, seja promovendo sua viabilidade ou apontando limitações que exigem adaptações.

#### **2.7.1.1 – Legislação Municipal**

Como discutido no tópico de área de atuação, a cidade de Mogi Guaçu tem investido na transformação digital da administração pública. Esse ambiente favorece a adoção de soluções como o New T.E. por empresas locais, especialmente quando há abertura institucional à digitalização de processos.

**Lei nº 5.782/2023 – Programa Guaçu Digital:** “Art. 1º Fica instituído, no âmbito da Administração Pública do Município de Mogi Guaçu, o Programa Guaçu Digital, com vistas à produção, gestão, tramitação, armazenamento, preservação, segurança e acesso a documentos e informações arquivísticas em ambiente digital de gestão documental. ”, (MOGI GUAÇU (SP), 2023)

Institui a digitalização ampla de serviços, processos e assinaturas eletrônicas no município, visando maior eficiência administrativa e sustentabilidade, (\_\_\_\_, 2023).

**Impacto:** favorece a legitimação de *softwares* digitais no ambiente institucional, podendo ser estendido à rede privada de empresas da cidade.

### 2.7.1.2 – Legislação Estadual

**Decreto nº 68.306/2024 – Assinaturas eletrônicas no Estado de SP:** “Artigo 1º - Os órgãos e entidades da Administração Pública estadual direta, autárquica e fundacional observarão as normas deste decreto para: I – definir os níveis mínimos exigidos para a assinatura eletrônica em documentos e em interações com o ente público, em conformidade com o artigo 5º da Lei federal nº 14.063, de 23 de setembro de 2020; II – realizar recadastramentos, atualizações cadastrais, recenseamentos e comprovação de vida;” (SÃO PAULO, 2024)

Regulamenta o uso de assinaturas eletrônicas no âmbito estadual, classificando-as em simples, avançadas e qualificadas, conforme a Lei Federal nº 14.063/2020, (\_\_\_\_, 2024).

**Impacto:** reforça a validade jurídica das assinaturas realizadas na plataforma, viabilizando contratos e aprovações digitais.

**Decreto nº 68.385/2024 – Complemento ao decreto anterior:** “Artigo único - No exercício de 2024, o recadastramento de que trata o Decreto nº 52.691, de 1º de fevereiro de 2008, será realizado por todos os servidores, empregados públicos e militares em atividade, até o dia 30 de abril de 2024. ”, (\_\_\_\_, 2024).

“Parágrafo único - O prazo disposto no "*caput*" deste artigo poderá ser prorrogado por resolução do Secretário de Gestão e Governo Digital.". (NR)”  
Atualiza regras sobre recadastramento digital e detalha o uso da assinatura qualificada, ampliando o escopo digital de processos administrativos, (\_\_\_\_, 2024).

**Impacto:** permite integração plena da plataforma com rotinas administrativas e documentos oficiais.

### 2.7.1.3 – Legislação Federal

**Lei nº 13.709/2018 – LGPD (Lei Geral de Proteção de Dados Pessoais):** “ Art. 46. Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito. ”, (BRASIL, 2018).

Define regras para coleta, tratamento e armazenamento de dados pessoais, estabelecendo direitos dos titulares e obrigações do controlador, (\_\_\_\_, 2018).

**Impacto:** exige do sistema recursos como criptografia, consentimento explícito e governança de dados, fortalecendo a confiabilidade e segurança do *software*.

**Lei nº 13.853/2019 – Criação da ANPD:** “Art. 55-A. Fica criada a Autoridade Nacional de Proteção de Dados (ANPD), autarquia de natureza especial, dotada de autonomia técnica e decisória, com patrimônio próprio e com sede e foro no Distrito Federal.”,(BRASIL, 2019).

Institui a Autoridade Nacional de Proteção de Dados, responsável por fiscalizar e regulamentar a LGPD, (\_\_\_\_, 2019).

**Impacto:** oferece base institucional para cumprimento e atualização constante das regras sobre dados pessoais no sistema.

**Lei nº 14.063/2020 – Assinaturas Eletrônicas:** “Para os fins da Lei nº 14.063/2020, as assinaturas eletrônicas classificam-se em: simples, que permite identificar o signatário e associar dados eletrônicos; avançada, que utiliza certificados não emitidos pela ICP-Brasil ou outros meios aceitos entre as partes, vinculada de forma unívoca ao signatário, gerada sob seu controle exclusivo e capaz de detectar alterações posteriores; e qualificada, que utiliza certificado digital emitido no âmbito da ICP-Brasil, conforme disposto no § 1º do art. 10 da Medida Provisória nº 2.200-2/2001.”, (BRASIL, 2020)

**Impacto:** permite que o sistema New T.E. formalize juridicamente contratos, autorizações e relatórios via assinatura eletrônica.

**Lei nº 9.610/1998 – Lei de Direitos Autorais:** Art. 5º, inciso VIII – Definição de “obra intelectual”:

"obra: ..."

“VIII – obra: ...”

“i) audiovisual – a que resulta da fixação de imagens com ou sem som, que tenha a finalidade de criar, por meio de sua reprodução, a impressão de movimento, independentemente dos processos de sua captação, do suporte usado inicial ou posteriormente para fixá-lo, bem como dos meios utilizados para sua veiculação;”

Protege obras intelectuais, como vídeos, interfaces e textos usados na plataforma, (BRASIL, 1998).

**Impacto:** garante exclusividade ao conteúdo do sistema, protegendo juridicamente suas produções.

**Lei nº 9.279/1996 – Lei de Propriedade Industrial:** “A proteção dos direitos relativos à propriedade industrial, considerado o seu interesse social e o desenvolvimento tecnológico e econômico do País, efetua-se mediante: ... “, (BRASIL, 1996).

“... III — concessão de registro de marca;”, (\_\_\_\_, 1996).

Prevê o registro de marcas, patentes e invenções industriais, (\_\_\_\_, 1996).

**Impacto:** permite o registro do nome “New T.E.” e de suas tecnologias exclusivas, fortalecendo a identidade e o valor de mercado da plataforma.

**CLT – Consolidação das Leis do Trabalho:** Estabelece exigências para o processo admissional de funcionários, como exames médicos, treinamentos e documentação obrigatória.

**Impacto:** o sistema pode estruturar e acompanhar essas etapas, mas algumas ainda exigirão presença, o que reforça a necessidade de uma abordagem híbrida (digital + física).

### 2.7.2 – Contratual

A operação de um sistema como o New T.E. exige, além da adequação legal, a formalização contratual entre a empresa desenvolvedora e seus clientes. Essa formalização ocorre por meio de um contrato de licença de uso e prestação de serviços, documento essencial para definir as responsabilidades, limitações e garantias envolvidas na utilização da plataforma.

Nesse tipo de relação jurídica, é necessário assegurar a proteção dos ativos intelectuais do sistema, a conformidade com as legislações de proteção de dados e direitos autorais, bem como a delimitação clara dos direitos e deveres das partes envolvidas. O contrato se torna, portanto, o principal instrumento jurídico de proteção, regulação e estabilidade da solução tecnológica apresentada.

#### 2.7.2.1 - Cláusula 1 - Objeto

O contrato celebrado entre a empresa New T.E. e o cliente tem por objeto a concessão de uma licença de uso não exclusiva da plataforma, autorizando sua utilização por empresas contratantes em território nacional e internacional.

Essa licença permite o acesso às funcionalidades do sistema para fins de *onboarding* digital de funcionários, incluindo recursos como gamificação de tarefas, comunicação com supervisores, agendamento de atividades e organização de etapas admissionais. A concessão de uso não implica em cessão de titularidade sobre o *software*, que permanece sob domínio integral da New T.E., tampouco permite qualquer modificação não autorizada da plataforma.

#### 2.7.2.2 - Cláusula 2 – Legislação Aplicável

A relação contratual está submetida às normas brasileiras que regem a tecnologia, proteção de dados, assinaturas eletrônicas, propriedade intelectual e relações trabalhistas, com destaque para a LGPD (Lei nº 13.709/2018), a Lei nº 14.063/2020, a CLT, a Lei de Direitos Autorais (Lei nº 9.610/1998) e a Lei de Propriedade Industrial (Lei nº 9.279/1996).

#### 2.7.2.3 - Cláusula 3 – Direitos da New T.E.

A New T.E. detém a propriedade integral do *software*, incluindo seu código-fonte, identidade visual, marca, materiais audiovisuais, trilhas, interfaces e demais elementos de *design* e funcionalidade. É direito da empresa realizar atualizações contínuas na plataforma, com o objetivo de aprimorar a experiência do usuário, corrigir eventuais falhas e manter a conformidade com as legislações em vigor. A empresa também se reserva o direito de suspender o acesso à plataforma em caso de inadimplência contratual, respeitando os prazos e condições previamente acordados.

#### 2.7.2.4 - Cláusula 4 – Deveres da New T.E.

A New T.E. compromete-se a prestar suporte técnico adequado durante toda a vigência contratual, oferecendo canais de atendimento por *e-mail* e *chat*. Também é dever da empresa realizar correções de falhas e *bugs* eventualmente identificados, além de garantir atualizações constantes da plataforma, tanto em termos de segurança quanto de usabilidade. Outro dever essencial é o cumprimento das obrigações previstas na legislação de proteção de dados, como a LGPD, assegurando governança responsável sobre as informações tratadas dentro do sistema.

#### 2.7.2.5 - Cláusula 5 – Direitos do Cliente

Ao contratar a plataforma, o cliente tem direito ao acesso contínuo ao sistema enquanto estiver adimplente, podendo usufruir de todas as funcionalidades disponibilizadas. Também lhe é assegurado o direito ao suporte técnico e à exportação dos dados inseridos na plataforma, mediante solicitação formal. Adicionalmente, o cliente poderá realizar contratos, aprovações e registros diretamente pelo sistema, tendo em vista que a plataforma está amparada por legislação vigente que confere validade jurídica às assinaturas eletrônicas utilizadas.

#### 2.7.2.6 - Cláusula 6 – Deveres do Cliente

Compete ao cliente efetuar pontualmente os pagamentos das mensalidades ou valores acordados, sendo este um requisito indispensável para a manutenção do acesso à plataforma. É de responsabilidade exclusiva do contratante a utilização do sistema dentro dos limites estabelecidos em contrato, sendo vedada a prática de engenharia reversa, reprodução, cópia ou qualquer forma de modificação da estrutura do *software*.

#### 2.7.2.7 - Cláusula 7 – Usuários Finais

Os usuários finais do sistema, ou seja, os funcionários inseridos na plataforma pelas empresas contratantes, têm garantido o direito ao acesso seguro e contínuo à ferramenta, bem como à confidencialidade e proteção de seus dados pessoais.

Em contrapartida, devem utilizar o sistema de forma ética e profissional, mantendo sigilo sobre suas credenciais de acesso e respeitando as políticas de uso estabelecidas pela contratante e pela New T.E. A violação desses deveres poderá acarretar bloqueio de acesso ou outras medidas previstas contratualmente.

A Consolidação das Leis do Trabalho (CLT) estabelece os princípios jurídicos que regem a relação entre empregador e empregado no Brasil. A aplicação da CLT no contexto do sistema New T.E. está diretamente relacionada à formalização do vínculo empregatício e à organização do processo de integração de novos colaboradores.

O Artigo 442 da CLT dispõe que: “Contrato individual de trabalho é o acordo tácito ou expresso, correspondente à relação de emprego.”

Esse artigo fundamenta o uso do sistema como meio válido para firmar contratos de trabalho de forma digital, desde que respeitados os requisitos legais. Por meio de funcionalidades específicas, como coleta de dados, registros de aceite e provas de entrega de conteúdo, o sistema contribui para dar legitimidade à contratação digital e à documentação formal exigida pela legislação trabalhista.

#### 2.7.2.8 - Cláusula 8 – Propriedade Intelectual

Todo o conteúdo desenvolvido para a plataforma está protegido pela legislação vigente sobre propriedade intelectual, como a Lei de Direitos Autorais e a Lei de Propriedade Industrial. Isso inclui, mas não se limita a vídeos instrutivos, conteúdos educativos, interfaces, logotipo, trilhas sonoras e componentes técnicos do sistema. A marca New T.E., bem como eventuais inovações tecnológicas associadas ao *software*, poderá ser registrada junto ao Instituto Nacional da Propriedade Industrial (INPI), garantindo sua exclusividade de uso e exploração.

#### 2.7.2.9 - Cláusula 9 – Confidencialidade

Ambas as partes se comprometem a manter em absoluto sigilo todas as informações técnicas, comerciais, estratégicas ou quaisquer outros dados confidenciais compartilhados durante a execução do contrato. Essa obrigação de confidencialidade se estende mesmo após a rescisão contratual, sendo aplicável a funcionários, parceiros ou representantes das partes envolvidas.

#### 2.7.2.10 - Cláusula 10 – Rescisão

O contrato poderá ser rescindido unilateralmente em caso de inadimplência superior a trinta dias, descumprimento de quaisquer obrigações contratuais ou mediante aviso prévio nos prazos estabelecidos. A rescisão acarretará a suspensão imediata do acesso ao sistema, sendo preservados os direitos da New T.E. em relação à propriedade intelectual e à proteção dos dados processados durante a vigência do contrato.

#### 2.7.2.11 - Cláusula 11 – Foro

Para dirimir eventuais conflitos oriundos do contrato, fica eleito o foro da Comarca de Mogi Guaçu – SP, com renúncia expressa de qualquer outro, por mais privilegiado que seja. Nos casos em que o cliente se localizar fora do Brasil, poderá ser adotada a mediação contratual ou foro alternativo, de acordo com a localidade e as práticas legais aceitas internacionalmente.

### 2.7.3 – Contrato de Licença

#### CONTRATO DE LICENÇA DE USO DE SOFTWARE – PLATAFORMA NEW T.E.

##### CONTRATANTE:

[Nome da empresa contratante], sociedade empresária inscrita no CNPJ sob o nº \_\_\_\_\_, com sede na \_\_\_\_\_, e endereço eletrônico \_\_\_\_\_, neste ato representada por \_\_\_\_\_, nacionalidade \_\_\_\_\_, estado civil \_\_\_\_\_, inscrito(a) no CPF sob o nº \_\_\_\_\_.

##### CONTRATADA:

NEW T.E. TECNOLOGIA LTDA, sociedade empresária inscrita no CNPJ sob o nº \_\_\_\_\_, com sede na cidade de Mogi Guaçu/SP, à [endereço completo], e endereço eletrônico [endereço de *e-mail*], neste ato representada por seus administradores legais, nacionalidade \_\_\_\_\_, estado civil \_\_\_\_\_, inscritos no CPF sob os nºs \_\_\_\_\_.

CONTRATANTE e CONTRATADA, doravante denominadas, em conjunto, Partes, e, individualmente, Parte, firmam o presente Contrato de Licença de Uso de *Software*, com base na boa-fé, na confiança mútua e na vontade livre e expressa das Partes, que acordam as seguintes disposições:

## **DO OBJETO**

O presente contrato tem por objeto a concessão, pela CONTRATADA à CONTRATANTE, de licença de uso não exclusiva, intransferível e limitada da plataforma digital New T.E., desenvolvida com o propósito de otimizar o processo de integração e *onboarding* de colaboradores.

A licença inclui o acesso a funcionalidades como: gamificação de tarefas, comunicação direta com supervisores, agendamento de atividades e organização das etapas admissionais.

A presente licença não transfere qualquer titularidade sobre o *software*, incluindo código-fonte, identidade visual, banco de dados, trilhas sonoras, logotipo ou quaisquer outros ativos associados à plataforma.

## **DA EXECUÇÃO E PAGAMENTO**

A licença será disponibilizada por período definido no Termo de Adesão ou Proposta Comercial assinada pelas Partes, sem que se configure qualquer vínculo empregatício entre as Partes ou entre seus representantes.

A CONTRATANTE pagará à CONTRATADA o valor de R\$ \_\_\_\_\_ (\_\_\_\_\_), sendo as subseqüentes no mesmo dia dos meses seguintes.

Na hipótese de atraso por culpa da CONTRATANTE, incidirão multa de 2% (dois por cento) sobre o valor em atraso, além de juros de mora de 1% (um por cento) ao mês.

Os valores contratados poderão ser reajustados anualmente com base na variação do índice IPCA, mediante assinatura de termo aditivo.

## **DAS OBRIGAÇÕES DA CONTRATADA**

O(A) CONTRATADO(A) compromete-se a:

- a. Manter a plataforma em constante atualização e funcionamento, realizando melhorias técnicas, correções de falhas e ajustes de segurança;
- b. Disponibilizar suporte técnico aos usuários da CONTRATANTE por meio de canais digitais durante todo o período de vigência contratual;
- c. Garantir o tratamento de dados pessoais em conformidade com a Lei Geral de Proteção de Dados (Lei nº 13.709/2018), respeitando as diretrizes do GDPR quando aplicável;
- d. Preservar a confidencialidade das informações e dados acessados durante a prestação dos serviços.

## **DAS OBRIGAÇÕES DA CONTRATANTE**

O(A) CONTRATANTE compromete-se a:

- a. Utilizar a plataforma exclusivamente para fins internos, sem qualquer tentativa de engenharia reversa, cópia, modificação ou redistribuição;
- b. Realizar os pagamentos de forma pontual;
- c. Assegurar a veracidade dos dados inseridos e responsabilizar-se integralmente pelas informações lançadas no sistema;
- d. Fornecer suporte adequado aos seus próprios usuários internos, além de respeitar os termos de uso da plataforma.

## **DOS USUÁRIOS FINAIS**

Os usuários finais, definidos como os colaboradores inseridos na plataforma pela CONTRATANTE, terão direito ao acesso seguro e contínuo ao sistema, bem como à proteção de seus dados. Estes usuários devem utilizar a plataforma de forma ética, sendo vedada a divulgação de suas credenciais ou qualquer forma de uso indevido.

## **DA PROPRIEDADE INTELECTUAL E CONFIDENCIALIDADE**

Todos os direitos sobre a plataforma New T.E., incluindo *design*, interface, componentes técnicos, materiais instrutivos, marca e eventuais inovações, pertencem exclusivamente à CONTRATADA.

Nenhuma das disposições deste contrato implica cessão de direitos de propriedade intelectual.

As Partes obrigam-se a manter sigilo absoluto sobre informações técnicas, comerciais, estratégicas e quaisquer dados classificados como confidenciais, por prazo indeterminado, mesmo após a extinção do presente contrato.

## **DA VIGÊNCIA, RESCISÃO E PENALIDADES**

O contrato terá vigência de \_\_\_\_ (\_\_\_\_) meses, contados a partir da assinatura, podendo ser renovado por igual período mediante acordo entre as Partes.

A rescisão poderá ocorrer a qualquer tempo, por inadimplemento contratual, mediante aviso prévio de 10 (dez) dias úteis ou por comum acordo.

Em caso de descumprimento das obrigações contratuais, a parte infratora estará sujeita a multa de 10% (dez por cento) sobre o valor total do contrato, além de perdas e danos, se cabíveis.

Caso fortuito ou força maior que impossibilite a execução do contrato por mais de 15 (quinze) dias úteis poderá ensejar sua rescisão sem ônus adicionais, mediante notificação expressa.

## **DAS DISPOSIÇÕES GERAIS**

Este contrato não configura qualquer vínculo societário, associativo ou de representação entre as Partes.

As comunicações entre as Partes deverão ocorrer por escrito, mediante *e-mail*, correspondência com aviso de recebimento ou sistemas eletrônicos previamente definidos.

As Partes reconhecem e respeitam os direitos de propriedade intelectual um do outro e assumem, inclusive após o término da vigência, o compromisso de confidencialidade e não concorrência, por prazo de 1 (um) mês.

**DO FORO**

Fica eleito o foro da Comarca de Mogi Guaçu/SP, com renúncia a qualquer outro, por mais privilegiado que seja, para dirimir eventuais controvérsias decorrentes deste contrato.

E por estarem justas e contratadas, firmam o presente instrumento, em duas vias de igual teor, para que produza seus jurídicos e legais efeitos.

Mogi Guaçu/SP, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_

**CONTRATANTE**

Nome:

Cargo:

CPF/CNPJ:

**CONTRATADA – NEW T.E. TECNOLOGIA LTDA**

Nome:

Cargo:

CPF/CNPJ:

**TESTEMUNHAS**

1. \_\_\_\_\_

Nome:

CPF:

2. \_\_\_\_\_

Nome:

CPF:

### 3 - DOCUMENTAÇÃO

A documentação do New T.E. é fundamental para demonstrar de forma clara como o sistema foi desenvolvido e como suas funcionalidades foram implementadas. Neste capítulo, serão apresentadas as telas do *software* acompanhadas de seus respectivos códigos, com explicações detalhadas sobre a função de cada parte.

O objetivo é oferecer uma visão prática e organizada do funcionamento da aplicação, facilitando a compreensão de sua estrutura, lógica e recursos. Dessa forma, este capítulo registra o resultado do desenvolvimento técnico, servindo como referência tanto para usuários quanto para futuros aprimoramentos do sistema.

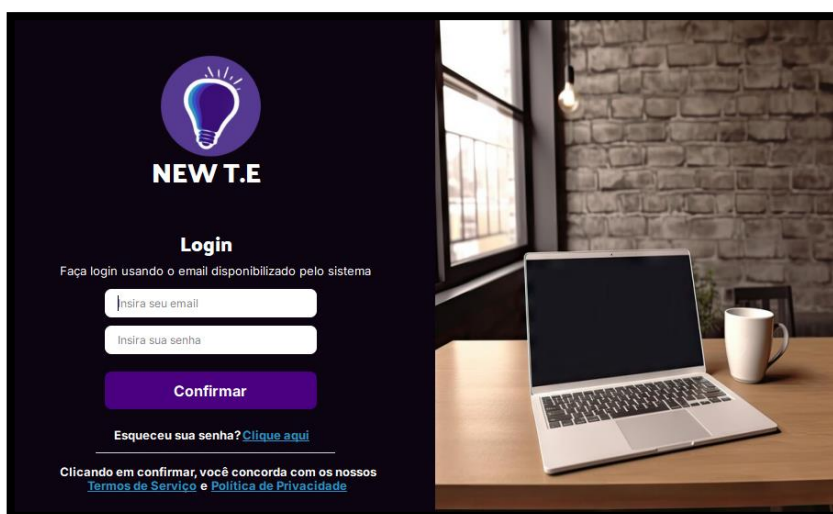
#### 3.1 – TELAS E CÓDIGOS DA APLICAÇÃO EM PYTHON

A seguir, estão as telas da aplicação da New T.E. ordenadas de modo à como um usuário se locomoveria pelo programa, incluindo telas de *Login*; Recuperação de Senha, as telas de *Chat*, perfis de administradores e suas *dashboards*, sendo esses os ambientes onde os *admins* podem ver eventos marcados em um calendário ou a progressão das etapas dos novos colaboradores por meio de gráficos.

Ao relacionar com os colaboradores em si, eles terão disponibilidade de acesso à alguns recursos do administrador, além de seu próprio *dashboard* que estará incluso com seus “guias” (propostos no capítulo 2).

##### 3.1.1 – Tela de *Login*

Figura 36 - Tela de *Login*



Fonte: (AUTORIA PRÓPRIA, 2025)

Aqui, na figura 36, encontra-se a Tela de *Login* do sistema da New T.E, que conta com a logo, e duas caixas de texto, para a inserção do *e-mail* e da senha que são disponibilizadas à companhia-cliente na contratação do *SaaS*. Além disso, conta com três botões de *hiperlink*, sendo eles, respectivamente, de cima para baixo e esquerda para a direita: O Botão de “Esqueceu a senha” o dos “Termos de Serviço” e da “Política de Privacidade”. No caso dos dois últimos, eles levariam à outras telas com os termos e a política descritos para a leitura do usuário, ou do cliente, se for preciso.

### 3.1.2 – Código da Tela de *Login*

Figura 37 - Código da Tela de Login 1

```
class Login(QMainWindow): 2 usages  LeonardoLobos +2
    def __init__(self, stacked_widget):  LeonardoLobos +1
        super().__init__()
        loadUi(uifile: "../design/login.ui", self)
        self.widget = stacked_widget

        self.logar.clicked.connect(self.loginfunction)
        self.senha.setEchoMode(QtWidgets.QLineEdit.Password)
        self.trocar.clicked.connect(self.mudartela)

    def loginfunction(self): 1 usage  LeonardoLobos +2
        email = self.usuario.text()
        senha = self.senha.text()

        self.logarAplicativo()

        if not email or not senha:
            QMessageBox.warning(None, "Aviso", "Por favor, preencha todos os campos.")
            return

        db = bancoDados().conectar()
        if not db:
            return

        cursor = db.cursor()

        query = "SELECT email, senha FROM usuario WHERE email = %s AND senha = %s"
        cursor.execute(query, (email, senha))
        result = cursor.fetchone()
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 37 ilustra a implementação da classe responsável pela tela de *login* da aplicação, utilizando a biblioteca PyQt5 para construção da interface gráfica. A classe *Login* herda de *QMainWindow* e, em seu método construtor (`__init__`), realiza o carregamento do arquivo de *layout* `.ui` da interface gráfica, além de inicializar os *widgets* da tela e conectar os botões às suas respectivas funções.

Um dos principais métodos é “loginfunction”, acionado quando o botão de *login* é clicado. Esse método realiza a coleta dos dados inseridos nos campos de texto da interface gráfica: “usuário”, que representa o campo de *e-mail*, e senha, que representa a senha digitada pelo usuário. Ambos os dados são extraídos utilizando os métodos `text()` dos widgets de entrada. A seguir, é realizada uma verificação para garantir que os campos não estejam vazios. Caso um ou ambos estejam sem preenchimento, é exibida uma mensagem de aviso ao usuário por meio da função `QMessageBox.warning`, solicitando que todos os campos sejam devidamente preenchidos.

Após a validação local, o método tenta estabelecer uma conexão com o banco de dados por meio do método `conectar()` da classe `bancoDados`. Se a conexão for bem-sucedida, um *cursor* é criado para executar comandos SQL. Nesse momento, a consulta é preparada para buscar um registro da tabela “usuário”, verificando se existe um usuário cujo *e-mail* e senha correspondam aos dados informados na interface.

Figura 38 - Código da Tela de Login 2

```
if result is None:
    QMessageBox.warning(None, "Aviso", "Usuário ou senha incorretos.")
else:
    QMessageBox.information(None, "Bem-Vindo!", "Logado com sucesso!")
    self.logarAplicativo()

cursor.close()
db.close()

def mudartela(self): 1 usage  ⤴ LeonardoLobos +1
    createacc = EsqueciSenha(self.widget)
    self.widget.addWidget(createacc)
    self.widget.setCurrentIndex(self.widget.currentIndex() + 1)

def logarAplicativo(self): 2 usages  ⤴ LeonardoLobos +1
    self.main = dashboard.TelaInicial(self.widget)
    self.main.show()
    self.widget.hide()
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 38 complementa o processo de autenticação iniciado anteriormente. Após a execução da instrução SQL, o método verifica o resultado da consulta utilizando `cursor.fetchone()`. Caso o retorno seja `None`, isso indica que nenhum registro na tabela “usuário” corresponde aos dados fornecidos.

Nessa situação, uma mensagem de alerta é exibida informando que o *e-mail* ou a senha estão incorretos, evitando que o acesso ao sistema seja concedido com credenciais inválidas.

Por outro lado, se a consulta retornar um registro válido, significa que o usuário foi autenticado com sucesso. Uma mensagem de confirmação é apresentada por meio de `QMessageBox.information`, e o método `logarAplicativo` é chamado para iniciar o redirecionamento do usuário para a tela principal da aplicação. Esse método instancia a tela inicial (`TelaInicial`), a exibe utilizando `show()`, e oculta a tela de login com `hide()`.

A figura também exibe o método “`mudartela`”, responsável por alternar para a tela de recuperação de senha. Ao ser chamado, esse método cria uma nova instância da tela `EsqueciSenha`, adiciona o *widget* ao sistema de navegação e altera o índice da tela atual, permitindo uma transição fluida entre telas.

### 3.1.3 – Tela de Recuperação de Senha

Figura 39 - Tela de Recuperação de Senha



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 39 exibe a tela de recuperação de senha do sistema New T.E. Nessa interface, o usuário é orientado a inserir seu endereço de *e-mail* no campo disponibilizado, para que possa iniciar o processo de recuperação da conta. Após o preenchimento, o botão “Enviar” permite confirmar a solicitação.

A tela também apresenta um *link* com a frase “Clique aqui”, que pode ser utilizado caso o usuário tenha lembrado da senha e deseje retornar à tela de *login*. Na parte inferior da interface, estão localizados dois *hiperlinks*: “Termos de Serviço” e “Política de Privacidade”, que redirecionam o usuário para páginas com essas informações.

### 3.1.4 – Código da Tela de Recuperação de Senha

Figura 40 - Código da Tela de Recuperação de Senha 1

```
class EmailSender(QThread): 1 usage  ↳ LeonardoLobos
    finished = pyqtSignal(str)

    def __init__(self, receiver_email):  ↳ LeonardoLobos
        super().__init__()
        self.receiver_email = receiver_email

    def run(self):  ↳ LeonardoLobos
        try:
            passCode = random.randint(a: 100000, b: 999999)

            sender_email = "newtetcc2025@gmail.com"
            app_password = "bbog pkqm nexm riyv"

            message = MIME multipart("related")
            message["From"] = sender_email
            message["To"] = self.receiver_email
            message["Subject"] = "New TE - Recuperação de Senha"

            html_content = f"""
            <html>
            <head>
            <style>
            body {{
                font-family: Arial, sans-serif;
            }}
            .header {{
                background-color: #6a1b9a; /* Cor de fundo roxa */
                color: white;
                padding: 20px;
                text-align: center;
                font-size: 24px;
            }}
            """
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 40 apresenta a estrutura funcional da tela de recuperação de senha do sistema. A classe *EsqueciSenha*, construída com *PyQt5*, é responsável por gerenciar os elementos visuais e a lógica de envio de *e-mail* para redefinição de senha. Nessa classe, o método *requisitarSenha* é executado quando o botão de envio da tela é acionado. Primeiramente, o método obtém o *e-mail* informado pelo usuário no campo de entrada. Caso o campo esteja vazio, é exibida uma mensagem de erro com uma cor destacada, informando que um *e-mail* válido deve ser inserido.

Essa mensagem é gerada dinamicamente pela função `mudarTexto`, que retorna uma *string* formatada em HTML para controle da formatação e da cor do texto.

Quando um *e-mail* é inserido corretamente, uma nova instância da classe `EmailSender` é criada, recebendo o *e-mail* digitado como parâmetro. Essa instância inicia uma *thread* paralela para executar o envio da mensagem de forma assíncrona, garantindo que a interface gráfica não seja bloqueada durante a operação. A confirmação da finalização do processo é exibida por meio da alteração do texto do componente de verificação, indicando que o *e-mail* foi enviado ou que ocorreu uma falha no processo.

Figura 41 - Código da Tela de Recuperação de Senha 2

```

class EmailSender(QThread): 1 usage  LeonardoLobos
    finished = pyqtSignal(str)

    def __init__(self, receiver_email):  LeonardoLobos
        super().__init__()
        self.receiver_email = receiver_email

    def run(self):  LeonardoLobos
        try:
            passCode = random.randint(a: 100000, b: 999999)

            sender_email = "newtetco2025@gmail.com"
            app_password = "bbog pkqm nexm riyv"

            message = MIME multipart("related")
            message["From"] = sender_email
            message["To"] = self.receiver_email
            message["Subject"] = "New TE - Recuperação de Senha"

            html_content = f"""
            <html>
            <head>
            <style>
            body {{
                font-family: Arial, sans-serif;
            }}
            .header {{
                background-color: #6a1b9a; /* Cor de fundo roxa */
                color: white;
                padding: 20px;
                text-align: center;
                font-size: 24px;
            }}
            """

```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 41 apresenta a classe `EmailSender`, que herda de `QThread` e é responsável por realizar o envio do *e-mail* de recuperação de senha. O método “run”, executado em uma *thread* separada, realiza a geração de um código numérico aleatório por meio da função “`random.randint`”, o qual pode ser utilizado para verificação de identidade em etapas posteriores.

Em seguida, são definidos o endereço de *e-mail* remetente e a senha de acesso do aplicativo, necessários para autenticação com o servidor SMTP do Gmail. A mensagem é criada utilizando a biblioteca “email.mime”, sendo composta pelos campos "From", "To" e "Subject", este último contendo o título “New TE – Recuperação de Senha”.

Além disso, é definido um conteúdo em HTML para o corpo da mensagem, permitindo formatação personalizada e inclusão de elementos visuais. Esse conteúdo é construído utilizando uma *string* multi-linha com interpolação de variáveis.

Figura 42 - Código da Tela de Recuperação de Senha 3

```
message.attach(MIMEText(html_content, _subtype: "html"))
image_path = "../imagens/logo_new_te.png"
if os.path.exists(image_path):
    with open(image_path, 'rb') as img_file:
        img = MIMEImage(img_file.read())
        img.add_header(_name: 'Content-ID', _value: '<logo_image>')
        img.add_header(_name: 'Content-Disposition', _value: 'inline', filename="image.jpg")
        message.attach(img)

# Connect and send
with smtplib.SMTP_SSL(host="smtp.gmail.com", port=465) as server:
    server.login(sender_email, app_password)
    server.sendmail(sender_email, self.receiver_email, message.as_string())

self.finished.emit("Email enviado!")

except Exception as e:
    self.finished.emit(f"Erro ao enviar: {str(e)}")
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 42 apresenta a continuação da classe EmailSender, com a montagem final da mensagem e o processo de envio. O conteúdo HTML criado é adicionado à mensagem por meio do método “attach”, especificando o subtipo "html". Em seguida, é verificado se o arquivo de imagem localizado em “../images/logo\_new\_te.png” existe. Caso positivo, a imagem é lida e adicionada como anexo à mensagem com cabeçalhos específicos, permitindo sua exibição *inline* no corpo do *e-mail*.

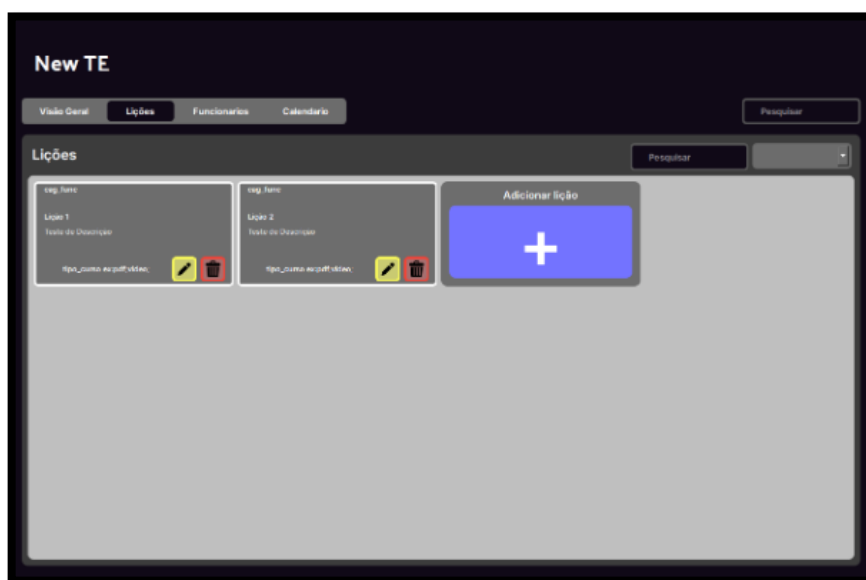
A conexão com o servidor SMTP é feita utilizando smtplib.SMTP\_SSL, no endereço smtp.gmail.com e porta 465. Após autenticação com o *e-mail* remetente e a senha de aplicativo, a mensagem é enviada ao destinatário por meio do método sendmail.

Por fim, o método emite um sinal indicando o sucesso da operação. Caso ocorra alguma exceção durante o processo, uma mensagem de erro é emitida, contendo os detalhes do problema ocorrido.

Nós próximos tópicos do TCC, serão utilizados como base a visão do supervisor para ampliar as funcionalidades do sistema.

### 3.1.5 – Tela de Atividades (Supervisor)

Figura 43 - Tela de Atividades (Supervisor)



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 43 apresenta a interface da aba “Lições” do sistema New T.E., responsável pela gestão de conteúdos didáticos destinados ao processo de integração de novos colaboradores.

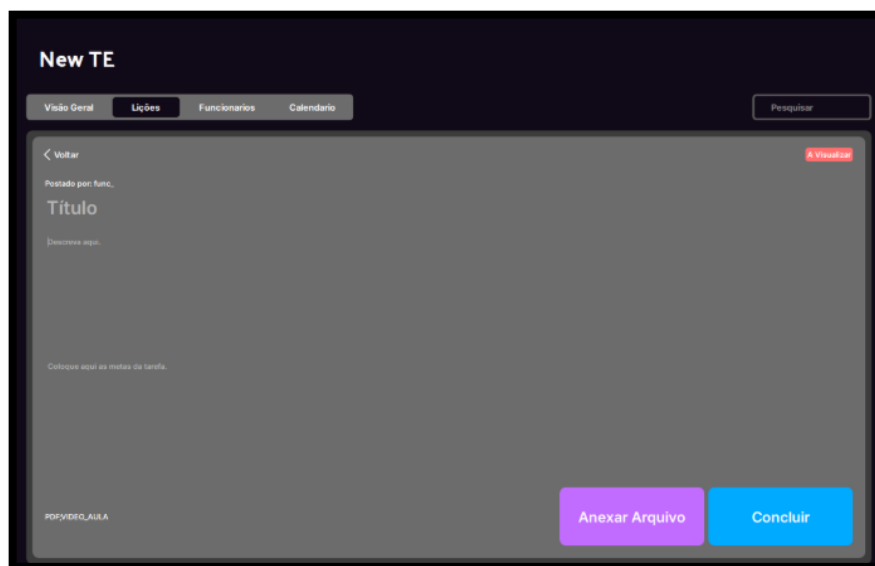
Nessa tela, o usuário visualiza um painel que exibe os módulos cadastrados em formato de cartões, cada um contendo o título da lição, uma breve descrição e ícones de ação para editar ou excluir o registro.

À direita, encontra-se o botão “Adicionar Lição”, destacado em azul, que direciona o usuário para o formulário de criação de novas lições. Na parte superior, há uma barra de pesquisa, permitindo a filtragem de conteúdos já cadastrados.

A interface foi desenvolvida para oferecer usabilidade intuitiva, priorizando o acesso rápido às funcionalidades de manutenção das lições, em consonância com as heurísticas de Nielsen aplicadas ao projeto.

### 3.1.6 – Tela de Lição Individual

Figura 44 - Tela de Lição (Supervisor)



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 44 apresenta a interface da aba “Lições” do sistema New T.E., responsável pela gestão de conteúdos didáticos destinados ao processo de integração de novos colaboradores.

Nessa tela, o usuário visualiza um painel que exhibe os módulos cadastrados em formato de cartões, cada um contendo o título da lição, uma breve descrição e ícones de ação para editar ou excluir o registro.

À direita, encontra-se o botão “Adicionar Lição”, destacado em azul, que direciona o usuário para o formulário de criação de novas lições. Na parte superior, há uma barra de pesquisa, permitindo a filtragem de conteúdos já cadastrados.

A interface foi desenvolvida para oferecer usabilidade intuitiva, priorizando o acesso rápido às funcionalidades de manutenção das lições, em consonância com as heurísticas de Nielsen aplicadas ao projeto.

### 3.1.7 – CRUD das Lições

As imagens de trechos de códigos a seguir referem-se ao CRUD do sistema de lições.

CRUD é a sigla para *Create*, *Read*, *Update* e *Delete*, que representam as quatro operações fundamentais em sistemas de gerenciamento de dados: criar, ler, atualizar e excluir informações. Esse modelo define a estrutura básica de interação entre a aplicação e o banco de dados, sendo amplamente utilizado no desenvolvimento de softwares para garantir a manipulação completa e organizada dos registros armazenados.

Figura 45 - Cadastro de Lições

```
def cadastrarLicao(self):
    db = bancoDados().conectar()
    if not db:
        return

    cursor = db.cursor()

    print(Session.current_user["id_user"])

    valuesDictionaries = {
        "id_user": Session.current_user["id_user"],
        "titulo": self.lineEdit_titulo.text(),
        "conteudo": self.textEdit_desc.toPlainText(),
        "metas": self.textEdit_metas.toPlainText(),
    }

    print("eba")

    for key, value in valuesDictionaries.items():
        if value is None or value == "":
            QMessageBox.warning(self, "Erro", f"Preencha todos os campos")
            return

    query = ""

    if self.licaoAlterar:
        query = f"""
        UPDATE licoes
        SET
            id_user = %,
            titulo = %,
            conteudo = %,
            metas = %
        WHERE id_licao = {self.licaoAlterar};
        """
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 45 apresenta a função `cadastroLicao()`, responsável por cadastrar novas lições ou atualizar registros já existentes no sistema New T.E..

O método conecta-se ao banco de dados por meio da função `bancoDados().conectar()` e coleta os dados inseridos na interface gráfica — título, descrição, metas e o identificador do usuário logado — armazenando-os em um dicionário de valores.

Antes de prosseguir, o sistema verifica se todos os campos obrigatórios foram preenchidos, exibindo uma mensagem de alerta caso algum campo esteja vazio (QMessageBox.warning).

Quando uma lição existente é selecionada para edição, o método executa o comando “SQL UPDATE Licoes”, substituindo as informações anteriores. Caso contrário, um novo registro é criado.

Essa função representa a operação de *Create/Update* do módulo, garantindo a integridade dos dados e a consistência entre a interface e o banco de dados.

Figura 46 - Visualização de Lições

```
def visualizarLicao(self, id_licao):
    """
    """
    self.abrirDashboard()

    db = bancoDados().conectar()
    if not db:
        return

    cursor = db.cursor()
    query = "SELECT titulo, conteudo, metas FROM licoes WHERE id_licao = %s"
    cursor.execute(query, (id_licao,))
    result = cursor.fetchone()

    self.licao_ativa = id_licao

    if not (result is None):
        titulo, conteudo, metas = result

        # self.titulo_cadastro_2.setText("**Alterando {titulo} {self.alterar}")

        self.lbl_titulo.setText(f"<html><head></head><body><p><span style=' font-size:20pt;'>{titulo}</span></p></body></html>")
        self.lbl_desc.setText(f"<html><head></head><body><p><span style=' font-size:12pt;'>{conteudo}</span></p></body></html>")
        self.lbl_metas.setText(f"<html><head></head><body><p><span style=' font-size:10pt; font-weight:600; color:#d1d1d1;'>{metas}</span></p></body></html>")

    cursor.close()
    db.close()
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 46 refere-se à função `visualizarLicao()`, responsável por abrir uma janela dedicada à exibição detalhada da lição selecionada.

Esse método carrega do banco de dados as informações completas da lição: título, descrição, metas e anexos, e as insere nos respectivos campos da interface gráfica.

Além disso, permite que o usuário interaja com o conteúdo da lição de forma individual, sem risco de alteração ou exclusão acidental, oferecendo uma navegação mais segura e organizada.

A função de visualização complementa o ciclo do CRUD, garantindo a consulta de informações de maneira acessível e integrada ao restante do sistema.

Figura 47 - Atualização de Lições

```

def atualizarLicoes(self): #usages: 1 Theuzinh*
    container = self.scroll_Licoes.widget()

    self.scroll_Licoes.setWidgetResizable(True)
    layout = container.layout()

    self.clearLayout(layout)

    db = bancoDados().conectar()
    cursor = db.cursor()

    query = "SELECT id_licao FROM usuario_licao_realizada WHERE id_usuario = %s"
    cursor.execute(query, (Session.current_user['id_user'], ))
    list = cursor.fetchall()
    id_list = [row[0] for row in list]

    cursor.execute("SELECT * FROM licoes")
    rows = cursor.fetchall()

    i = 0
    print(id_list)
    for id_licao, id_user, titulo, desc, metas, criacao, validade in rows:
        print(id_licao)
        row = i // 4
        col = i % 4
        template = licao()
        layout.addWidget(template, row, col)
        template.lbl_tituloCurso.setText(titulo)
        template.lbl_descCurso.setText(desc)
        template.btn_visualizar.clicked.connect(partial(self.visualizarLicao, id_licao))
        i = i + 1

    if id_licao in id_list:
        print("ta")
        template.template_cursos_substituir.setStyleSheet('QWidget{\nbackground-color: rgb(109, 109, 109);\n\n#template_cursos_substituir{\n\n    border-radius:
        template.lbl_situacao_licao.setStyleSheet('background-color:rgb(111, 230, 111);\nborder-radius: 4px;')
        template.lbl_situacao_licao.setText(f'<html><head><body><p align="center">{concluido}</p></body></html>')

```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 47 demonstra a função `atualizarLicoes()`, responsável por atualizar dinamicamente a interface e listar as lições disponíveis para o usuário.

O método realiza consultas SQL que recuperam os registros de lições associados ao usuário logado e, em seguida, cria dinamicamente elementos gráficos para cada resultado.

Cada lição é representada por um bloco contendo o título, a descrição e o status de conclusão, com estilos visuais definidos por comandos Qt StyleSheet.

Além disso, cada bloco possui um botão que, ao ser clicado, chama o método `visualizarLicao()`, permitindo acessar o conteúdo completo da lição selecionada.

Essa função representa a operação de *Read* do CRUD, sendo responsável por sincronizar o banco de dados com a camada visual e garantir que as informações exibidas estejam sempre atualizadas.

Figura 48 - Exclusão de Lições

```

def alterarLicao(self, user_id):
    print(f"Alterar lição {user_id}")
    self.licaoAlterar = user_id
    self.mudarDashboard(4)

def excluirLicao(self, user_id, user_name):
    reply = QMessageBox.question(
        self,
        "Confirmação",
        f"Tem certeza que deseja excluir a lição {user_name} (ID {user_id})?",
        QMessageBox.Yes | QMessageBox.No,
        QMessageBox.No
    )
    if reply == QMessageBox.No:
        print(f"Usuário {user_id} não excluído")
        return

    ifdb = bancoDados().conectar()
    ifCursor = ifdb.cursor()

    query = f"""
        DELETE FROM Licoes WHERE id_licao = {user_id};
    """

    ifCursor.execute(query)
    ifdb.commit()
    ifCursor.close()
    ifdb.close()

    QMessageBox.information(self, "Sucesso", f"Lição {user_name} excluída com sucesso!")
    self.atualizarLicoes()
else:
    QMessageBox.information(self, "Cancelado", f"Exclusão cancelada.")
    print(f"Exclusão da lição {user_id} cancelada")

```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 48 apresenta o método `excluirLicao()`, que executa a exclusão controlada de uma lição no sistema.

O código inicia exibindo uma janela de confirmação (`QMessageBox.question`), solicitando que o usuário confirme a ação de exclusão. Caso o usuário aceite, o sistema conecta-se ao banco de dados e executa o comando SQL `DELETE FROM Licoes WHERE id_licao = {user_id}`, removendo o registro correspondente.

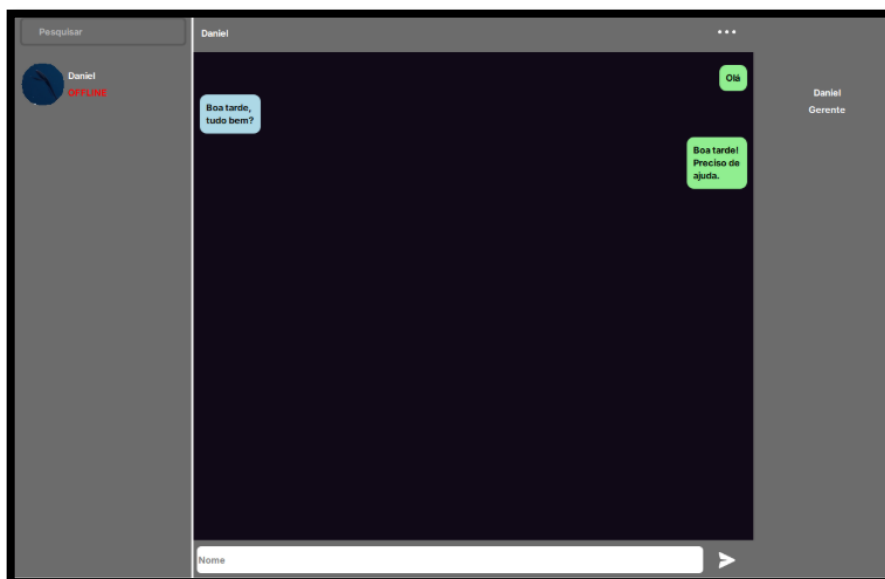
Após a exclusão, o sistema atualiza a listagem de lições (`self.atualizarLicoes()`) e exibe uma mensagem de sucesso (`QMessageBox.information`).

Se a ação for cancelada, uma mensagem informando a interrupção da exclusão é mostrada.

Esse método representa a operação de Delete do CRUD e garante que as ações críticas sejam executadas de forma segura e transparente ao usuário.

### 3.1.8 – Telas e Código do *Chat*

Figura 49 - Tela de *Chat*



Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 49 apresenta a interface do módulo de chat integrado ao sistema New T.E., que possibilita a comunicação direta entre os usuários cadastrados, como gestores e colaboradores.

O layout é dividido em três áreas principais: à esquerda, uma lista de contatos com status de conexão (por exemplo, “OFFLINE” em vermelho); ao centro, a área de mensagens, onde ocorre a troca de textos entre os participantes; e na parte inferior, o campo de entrada de mensagem com o ícone de envio.

As mensagens são exibidas em balões coloridos que distinguem o emissor do receptor, em tons de azul para mensagens recebidas e verde para mensagens enviadas, facilitando a leitura e o acompanhamento da conversa.

Essa funcionalidade promove a interação e suporte entre os membros da equipe, tornando o New T.E. uma plataforma mais dinâmica e colaborativa dentro do processo de integração corporativa.

Figura 50 - Trecho do Código do Chat

```

def run(self):
    while self.running:
        if Session.current_user is None:
            self.stop()
            return

        db = bancoDados().conectar()
        if not db:
            return
        cursor = db.cursor()

        query = f"""
        SELECT *
        FROM mensagens_chat
        WHERE id_mensagem > {Session.last_message_id} AND (
            (remetente_id = {Session.current_user['id_user']} AND destinatario_id = {Session.loaded_chat})
            OR
            (remetente_id = {Session.loaded_chat} AND destinatario_id = {Session.current_user['id_user']})
        )
        ORDER BY data_envio ASC;
        """
        cursor.execute(query)
        results = cursor.fetchall()

        if results:
            self.new_data.emit(results)

        db.close()
        cursor.close()
        time.sleep(2)

def stop(self):
    self.running = False

```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 50 apresenta a implementação da função `run()`, responsável por atualizar em tempo real as mensagens trocadas no módulo de chat do sistema New T.E..

A função opera em um loop contínuo, verificando periodicamente novas mensagens armazenadas no banco de dados. Para isso, conecta-se à tabela `mensagens_chat` e executa uma consulta SQL que recupera apenas os registros mais recentes, comparando o identificador da última mensagem recebida (`id_mensagem`) e os IDs de remetente e destinatário.

Quando novas mensagens são detectadas, o método `emit()` é acionado, enviando os dados para a interface gráfica, que é então atualizada automaticamente.

O processo é interrompido quando a função `stop()` é chamada, encerrando a execução do loop e definindo o valor de “running” como `False`.

Essa estrutura garante a comunicação assíncrona e contínua entre os usuários, proporcionando uma experiência de conversa fluida e sem necessidade de recarregar a página ou reiniciar o aplicativo.

### 3.1.9 – Tela e Código do Cadastro de Funcionários

Figura 51 - Tela de Cadastro de Funcionário



A imagem mostra a tela de cadastro de funcionários. O formulário principal contém os seguintes campos: Nome (campo de texto), Email Pessoal (campo de texto com o exemplo 'Email1234@gmail.com'), Telefone (campo de texto com máscara), Endereço (campo de texto com o exemplo 'Rua Tapaxanas'), CPF (campo de texto com máscara), RG (campo de texto com máscara), Depto (menu suspenso com 'Gerente' selecionado) e Cargo Empresarial (menu suspenso com 'Gerente' selecionado). À direita, há uma seção para a 'Foto do Funcionário' com uma imagem de exemplo e um botão 'Selecionar foto'. Na base da tela, há dois botões: 'Cancelar' (cinza) e 'Confirmar' (azul).

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 51 exibe a interface de cadastro de funcionários do sistema New T.E., projetada para armazenar e gerenciar informações pessoais e profissionais dos colaboradores.

O formulário inclui campos para nome, e-mail pessoal, telefone, endereço, CPF, RG, departamento e cargo empresarial, além da opção de anexar uma foto do funcionário.

No canto direito da tela, há uma área dedicada à exibição da imagem selecionada e botões para confirmar ou cancelar o cadastro.

A interface segue o padrão visual do sistema, com campos bem distribuídos e botões em tons contrastantes, garantindo boa legibilidade e usabilidade.

Esse módulo permite o registro padronizado dos colaboradores, servindo como base para controle interno e integração com outros setores da plataforma.

Figura 52 - Trecho de Código do Cadastro de Funcionário

```
def cadastrarUsuario(self):
    db = bancoDados().conectar()
    if not db:
        return

    cursor = db.cursor()

    # Convert pixmap to binary for DB
    if hasattr(self, "foto_pixmap") and self.foto_pixmap:
        ba = QByteArray()
        buffer = QBuffer(ba)
        buffer.open(QIODevice.WriteOnly)
        self.foto_pixmap.save(buffer, "PNG")
        buffer.close()
        img_data = ba.data()
    else:
        img_data = None # No image selected

    valuesDictionaries = {
        "nome": self.lineEdit_nome.text(),
        "email": self.lineEdit_email.text(),
        "telefone": ''.join(c for c in self.lineEdit_telefone.text() if c.isdigit()),
        "cpf": ''.join(c for c in self.lineEdit_cpf.text() if c.isdigit()),
        "rg": ''.join(c for c in self.lineEdit_rg.text() if c.isdigit()),
        "departamento": self.comboBox_depto.currentText(),
        "cargo": self.comboBox_cargo.currentText(),
        "foto_perfil": img_data,
        "status": "OFFLINE",
        "data_entrada": "2025-09-12",
        "sobre_mim": "Sobre mim...",
        "senha": "1234",
        "endereco": self.lineEdit_endereco.text(),
        "tipo_usuario": "user",
        "experiencias": "Experiências..."
    }
```

Fonte: (AUTORIA PRÓPRIA, 2025)

A Figura 52 apresenta a função `cadastraUsuario()`, responsável por inserir os dados coletados na interface de cadastro no banco de dados.

O método inicia conectando-se ao banco de dados e verificando se uma imagem foi selecionada pelo usuário. Caso positivo, a foto é convertida em formato binário (PNG) para armazenamento.

Em seguida, as informações inseridas nos campos do formulário, como nome, *e-mail*, telefone, documentos, cargo, *status* e endereço, são reunidas em um dicionário denominado `valuesDictionaries`.

Esses dados são então utilizados para compor o comando SQL de inserção na tabela de usuários.

A função também define valores padrão, como o *status* “*OFFLINE*”, senha temporária e data de criação do registro.

Essa rotina garante que todos os novos colaboradores sejam cadastrados de forma consistente e completa, estabelecendo a base do gerenciamento de usuários dentro do sistema New T.E..

## CONSIDERAÇÕES FINAIS

Com a finalização do projeto, foi possível concluir que o sistema desenvolvido alcançou parte significativa dos objetivos propostos. Entre as funcionalidades implementadas, destacam-se o cadastro de funcionários, os sistemas e interfaces em Python, além do cadastro das aulas e materiais voltados ao aprendizado dos novos colaboradores. Também foi implementado um *chat* para conversas entre os usuários, recurso essencial para facilitar a comunicação e a integração dentro da plataforma.

Durante o desenvolvimento do trabalho, a equipe adquiriu conhecimentos relevantes em programação em Python e modelagem de dados, além de aprofundar-se em aspectos teóricos e práticos sobre o funcionamento de empresas com alta rotatividade. Foram igualmente importantes os aprendizados relacionados à ética de trabalho e às entrevistas realizadas, que trouxeram uma visão mais clara sobre a situação atual dos estagiários em empresas e firmas. Esses elementos forneceram base sólida tanto para a construção técnica quanto para a reflexão sobre a aplicabilidade do sistema em cenários reais.

Apesar dos avanços, ainda existem pontos que podem ser aprimorados. Entre eles, a implementação de um *chatbot* integrado a uma inteligência artificial, que teria como objetivo atender demandas mais complexas, como requisições de documentos e o esclarecimento de dúvidas relacionadas a políticas empresariais. Tal funcionalidade serviria como apoio adicional ao trabalho dos supervisores, trazendo mais agilidade e confiabilidade ao processo de integração.

Em suma, o projeto demonstrou-se funcional e viável, oferecendo ferramentas que podem facilitar o processo de *onboarding* e a adaptação de novos colaboradores. Embora não tenha atingido todo o seu potencial até o momento, a continuidade do desenvolvimento pode transformar o sistema em uma solução ainda mais completa e capaz de atender diferentes tipos de empresas.

## REFERÊNCIAS BIBLIOGRÁFICAS

A2WORKS. Evolução dos servidores: veja como foi. 2022. Disponível em: <<https://a2works.com.br/noticias/evolucao-dos-servidores/>>. Acesso em: 11 abr. 2025.

AMAZON WEB SERVICES. O que é Python? – Guia de Python para iniciantes na nuvem. [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://aws.amazon.com/pt/what-is/python/>>. Acesso em: 4 abr. 2025.

AWARI. História do desenvolvimento de software: do início até hoje. 4 ago. 2023. Disponível em: <<https://awari.com.br/historia-do-desenvolvimento-de-software-do-inicio-ate-hoje/>>. Acesso em: 6 abr. 2025.

BARRO, Bruna B. Lista: as 10 linguagens de programação mais usadas em 2025. 17 jan. 2025. Disponível em: <<https://www.hostinger.com.br/tutoriais/linguagens-de-programacao-mais-usadas>>. Acesso em: 11 abr. 2025.

BRASIL. Lei nº 13.853, de 8 jul. 2019. Presidência da República, Casa Civil, Subchefia para Assuntos Jurídicos. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/\\_ato2019-2022/2019/lei/L13853.htm](https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/lei/L13853.htm)>. Acesso em: 24 jun. 2025.

BRASIL. Lei nº 14.063, de 23 set. 2020. Presidência da República, Casa Civil, Subchefia para Assuntos Jurídicos. Disponível em: <[https://planalto.gov.br/ccivil\\_03/\\_ato2019-2022/2020/lei/L14063.htm](https://planalto.gov.br/ccivil_03/_ato2019-2022/2020/lei/L14063.htm)>. Acesso em: 24 jun. 2025.

BRASIL. Lei nº 14.063, de 23 de setembro de 2020. Base Legislação da Presidência da República. Disponível em: <<https://legislacao.presidencia.gov.br/atos/?tipo=LEI&numero=14063&ano=2020&ato=a46MTQU1UMZpWT4e8>>. Acesso em: 1 ago. 2025.

BRASIL. Lei nº 9.279, de 14 maio 1996. Regula direitos e obrigações relativos à propriedade industrial. Presidência da República, Casa Civil, Subchefia para Assuntos Jurídicos. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/leis/l9279.htm](https://www.planalto.gov.br/ccivil_03/leis/l9279.htm)>. Acesso em: 24 jun. 2025.

BRASIL. Lei nº 9.610, de 19 fev. 1998. Altera, atualiza e consolida a legislação sobre direitos autorais. Presidência da República, Casa Civil, Subchefia para Assuntos Jurídicos. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/Leis/L9610.htm](https://www.planalto.gov.br/ccivil_03/Leis/L9610.htm)>. Acesso em: 24 jun. 2025.

CARDOSO, Rodrigo. Lógica de programação: o que é e para que serve? 4 dez. 2024. Disponível em: <<https://www.locaweb.com.br/blog/temas/codigo-aberto/logica-de-programacao-o-que-e/>>. Acesso em: 11 abr. 2025.

CONTROLE NET. O que são servidores e quais são suas aplicações. [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://www.controle.net/faq/o-que-sao-servidores>>. Acesso em: 4 abr. 2025.

DEBIAN. File: NEWS. DEBSOURCES. [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://sources.debian.org/src/pyqt5/5.15.2%2Bdfsg-3/NEWS/>>. Acesso em: 9 abr. 2025.

DIWAN, Amit. History of MySQL. Tutorials Point. 24 fev. 2021. Disponível em: <<https://www.tutorialspoint.com/discuss-the-history-of-mysql>>. Acesso em: 9 abr. 2025.

EDUVEM. 5 Desafios em um *Onboarding* e como uma RH Tech Pode te Ajudar. 8 abr. 2022. Disponível em: <<https://eduvem.com/5-desafios-em-um-onboarding-e-como-uma-rh-tech-pode-te-ajudar/>>. Acesso em: 16 maio 2025.

EMPRESAQUI. Listas de empresas em Mogi Guaçu, SP: confira porte, segmento, dívidas, data de abertura e muito mais! Disponível em: <[https://www.empresaquei.com.br/listas-de-empresas/SP/MOGI\\_GUACU](https://www.empresaquei.com.br/listas-de-empresas/SP/MOGI_GUACU)>. Acesso em: 13 jun. 2025.

I9AÇÃO. I9Ação – Jogos Corporativos – I9Ação Games. Disponível em: <<https://i9acao.com.br/onboarding>>. Acesso em: 21 jun. 2025.

JARMUL, Katherine; LAWSON, Richard. Em Python Web Scraping. Birmingham: Packt Publishing, 2017.

LUZ, Daniel. Quais as principais diferenças entre os antigos e os novos processos de *Onboarding*? 9 ago. 2022. Disponível em:

<<https://evapeople.com.br/onboarding/diferencas-antigo-novo-onboarding>>. Acesso em: 16 maio 2025.

MAMBO.IO. Enterprise Gamification Platform – On-Premise and SaaS Software. Disponível em: <<https://mambo.io/>>. Acesso em: 21 jun. 2025.

MENDES, Pedro. A evolução das linguagens de programação: da 1ª à 6ª geração e sua integração com o Genexus. 4 set. 2023. Disponível em: <<https://pt.linkedin.com/pulse/evolu%C3%A7%C3%A3o-das-linguagens-de-programa%C3%A7%C3%A3o-da-1%C2%AA-%C3%A06%C2%AA-gera%C3%A7%C3%A3o-mendes>>. Acesso em: 6 abr. 2025.

MOGI GUAÇU (SP). Lei Ordinária nº 5.782, de 27 de junho de 2023. Câmara Municipal de Mogi Guaçu. Disponível em: <[https://sistema.camaramogiguacu.sp.gov.br/consultas/norma\\_juridica/norma\\_juridica\\_mostrar\\_proc?cod\\_norma=41730](https://sistema.camaramogiguacu.sp.gov.br/consultas/norma_juridica/norma_juridica_mostrar_proc?cod_norma=41730)>. Acesso em: 24 jun. 2025.

MOGI GUAÇU (SP). Serviços – Programa Guaçu Digital. Prefeitura Municipal de Mogi Guaçu, 27 jun. 2023. Disponível em: <<https://digital.mogiguacu.sp.gov.br/programa-guacu-digital>>. Acesso em: 24 jun. 2025.

MONTEIRO, Leandro Pinho. O que é linguagem de programação? Universidade da Tecnologia. 2017. Disponível em: <<https://universidadedatecnologia.com.br/o-que-e-linguagem-de-programacao/>>. Acesso em: 4 abr. 2025.

PIPEFY. Automação de Processos com Inteligência Artificial | Agentes de IA | Pipefy. Disponível em: <<https://www.pipefy.com/pt-br/>>. Acesso em: 21 jun. 2025.

PYTHON SOFTWARE FOUNDATION. Python Documentation by Version. [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://www.python.org/doc/versions/>>. Acesso em: 6 abr. 2025.

QT COMPANY. About Qt/pt - Qt Wiki. [S. l.]: [s. n.], [s. d.]. Disponível em: <[https://wiki.qt.io/About\\_Qt/pt](https://wiki.qt.io/About_Qt/pt)>. Acesso em: 6 abr. 2025.

QT GROUP. Qt Widgets Designer Manual. [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://doc.qt.io/qt-6/qtdesigner-manual.html>>. Acesso em: 8 abr. 2025.

R.S, Marcos. Desenvolvimento desktop: um guia completo para iniciantes. 2024. Disponível em: <<https://skillstecnologicas.com/desenvolvimento-desktop/>>. Acesso em: 11 abr. 2025.

RIBAS, Pedro Ivo. A origem e evolução dos bancos de dados. Medium. 1 jun. 2022. Disponível em: <<https://medium.com/%40pedroivoflorentino/a-origem-e-evolu%C3%A7%C3%A3o-dos-bancos-de-dados-72be6480c94>>. Acesso em: 9 abr. 2025.

RICARDO. A história dos banco de dados. DevMedia. 2006. Disponível em: <<https://www.devmedia.com.br/a-historia-dos-banco-de-dados/1678>>. Acesso em: 9 abr. 2025.

RIVERBANK COMPUTING LIMITED. PyQt5. PyPI. 19 jul. 2024. Disponível em: <<https://pypi.org/project/PyQt5/>>. Acesso em: 8 abr. 2025.

RIVERBANK COMPUTING LIMITED. Riverbank Computing | Download. [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://www.riverbankcomputing.com/static/Downloads/PyQt5/ChangeLog-5.15.12.dev2410191555>>. Acesso em: 9 abr. 2025.

SAP. O que é modelagem de dados? [S. l.]: [s. n.], [s. d.]. Disponível em: <<https://www.sap.com/brazil/products/data-cloud/datasphere/what-is-data-modeling.html>>. Acesso em: 9 abr. 2025.

SANTOS, Bruno. Algoritmo: da antiguidade aos dias atuais. DIO. 5 nov. 2023. Disponível em: <<https://www.dio.me/articles/algoritmo-da-antiguidade-aos-dias-atuais>>. Acesso em: 11 abr. 2025.

SÃO PAULO. Decreto nº 68.306, de 16 jan. 2024. Assembleia Legislativa do Estado de São Paulo, 18 jan. 2024. Disponível em: <<https://www.al.sp.gov.br/repositorio/legislacao/decreto/2024/decreto-68306-16.01.2024.html>>. Acesso em: 24 jun. 2025.

SÃO PAULO. Decreto nº 68.385, de 12 mar. 2024. Assembleia Legislativa do Estado de São Paulo, 12 mar. 2024. Disponível em: <<https://www.al.sp.gov.br/repositorio/legislacao/decreto/2024/decreto-68385-12.03.2024.html>>. Acesso em: 24 jun. 2025.

SEBRAE. O estágio da transformação digital nas pequenas e médias empresas - Sebrae. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/artigos/o-estagio-da-transformacao-digital-nas-pequenas-e-medias-empresas,4fc28c180dfc5810VgnVCM1000001b00320aRCRD>>. Acesso em: 21 jun. 2025.

SOUSA, Hyalto. A história do Python. 18 ago. 2023. Disponível em: <<https://www.dio.me/articles/a-historia-do-python-JCO7UB>>. Acesso em: 11 abr. 2025.

TAVARES, Lucas. O que é web? Origem, funcionamento e evolução. 12 jan. 2025. Disponível em: <<https://www.melhoreshospedagem.com/o-que-e-web/>>. Acesso em: 6 abr. 2025.

USERGUIDING. Software de *Onboarding* e Adoção Digital – UserGuiding. Disponível em: <<https://userguiding.com/pt-br>>. Acesso em: 21 jun. 2025.

WEBER, Eduardo. Web server: o que é e como funciona? 20 abr. 2023. Disponível em: <<https://www.hostinger.com.br/tutoriais/web-server>>. Acesso em: 11 abr. 2025.

WILDGOOSE. Team Building Solutions | Virtual, Hybrid & In-Person Events | Wildgoose USA. Disponível em: <<https://wearewildgoose.com/usa/>>. Acesso em: 21 jun. 2025.

WK TECHNOLOGY. Aplicações desktop - o que são e como funcionam? 2022. Disponível em: <<https://wktechnology.com.br/aplicacoes-desktop-o-que-sao-e-como-funcionam/>>. Acesso em: 4 abr. 2025.

ZEMBRUZKI, Luciano. Qt Designer - Python Simplificado. 24 dez. 2024. Disponível em: <<https://pythonsimplificado.com.br/2024/12/24/qt-designer/>>. Acesso em: 8 abr. 2025.

## ANEXO

### ANEXO A – COMANDOS DO SCRIPT CONFIGURE.PY PARA PYQT5 5.3

Este anexo apresenta os comandos utilizados no *script* `configure.py` para configuração do PyQt5 versão 5.3, incluindo opções de compilação e suporte a plataformas móveis. Os comandos foram apresentados no capítulo 1.5.2, mais especificamente na página 32.

Comandos principais e suas descrições:

`--target-py-version`

Define a versão do Python para a qual o PyQt5 será compilado. Por exemplo, `--target-py-version=3.8` indica que a compilação deve ser feita para Python 3.8.

`--sysroot`

Especifica o diretório raiz do sistema para compilação cruzada. Essa opção é útil para indicar o ambiente alvo quando se compila para plataformas diferentes da máquina *host*, como iOS ou Android.

`--no-tools`

Impede a compilação das ferramentas auxiliares do PyQt5, como utilitários e exemplos, resultando em uma compilação mais enxuta.