

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DA PRAIA GRANDE CURSO SUPERIOR DE
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

RELATÓRIO TÉCNICO DE CONCLUSÃO DE CURSO

**SISTEMA DE GESTÃO DE DESEMPENHO: GESTÃO DE METAS E
EXPERIÊNCIA DO FUNCIONÁRIO**

**MARIA EDUARDA DA SILVA JOAQUIM
LUCAS CARVALHO SANTOS
RENAN SOARES RIBAS**

ORIENTADOR: PROF. JOSEFFE BARROSO DE OLIVEIRA

**PRAIA GRANDE
DEZEMBRO DE 2023**

MARIA EDUARDA DA SILVA JOAQUIM
LUCAS CARVALHO SANTOS
RENAN SOARES RIBAS

RELATÓRIO TÉCNICO DE CONCLUSÃO DE CURSO

SISTEMA DE GESTÃO DE DESEMPENHO: GESTÃO DE METAS E EXPERIÊNCIA
DO FUNCIONÁRIO

Relatório técnico apresentado como exigência parcial para aprovação no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia da Praia Grande.

Orientador: Prof. Joseffe Barroso de Oliveira

PRAIA GRANDE
DEZEMBRO DE 2023

JOAQUIM, Maria Eduarda da Silva

RIBAS, Renan Soares

SANTOS, Lucas Carvalho

Sistema de Gestão de Desempenho: Gestão de Metas e Experiência do Funcionário / Lucas Carvalho Santos; Maria Eduarda da Silva Joaquim; Renan Soares Ribas, 2023.

Total de folhas: 140 p.

Orientador: Professor Joseffe Barroso de Oliveira. Local: Praia Grande, 2023.

1. Software. 2. OKRs. 3. Clima Organizacional.

Relatório Técnico (Trabalho de Conclusão de Curso) – Centro de Educação Tecnológica Paula Souza, Faculdade de Tecnologia da Baixada Santista (FATEC Praia Grande), Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

ERRATA

JOAQUIM, Maria Eduarda da Silva; RIBAS, Renan Soares; SANTOS, Lucas Carvalho. Sistema de Gestão de Desempenho: Gestão de Metas e Experiência do Funcionário. 140 p. Trabalho de Conclusão de Curso de graduação de tecnólogo em Análise e Desenvolvimento de Sistemas. Praia Grande: Centro de Educação Tecnológica Paula Souza, Faculdade de Tecnologia da Praia Grande, 2023.

MARIA EDUARDA DA SILVA JOAQUIM

LUCAS CARVALHO SANTOS

RENAN SOARES RIBAS

**SISTEMA DE GESTÃO DE DESEMPENHO: GESTÃO DE METAS E EXPERIÊNCIA
DO FUNCIONÁRIO**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Tecnologia de Praia Grande,
como exigência parcial para obtenção do título
de tecnólogo em Análise e Desenvolvimento de
Sistemas.

Praia Grande, 25 de novembro de 2023.

Banca Examinadora

DocuSigned by:

Joseffe Barroso de Oliveira

4AC6A17D4EAD47C...

Prof. Joseffe Barroso de Oliveira (orientador)

Professor Avaliador 1

Professor Avaliador 2

RESUMO

O presente trabalho teve como objetivo desenvolver um sistema para gestão de desempenho organizacional que integre a gestão de metas com a experiência do funcionário, visando a otimização, produtividade e a saúde no alcance de metas. Para isso foram realizadas pesquisas bibliográficas acerca do método OKR (Objectives and Key Results) e da pesquisa organizacional de clima e engajamento. A metodologia utilizada englobou a observação direta da aplicação da gestão empresarial e, a partir dos entendimentos coletados foi desenvolvido um software utilizando o framework Flask para o back-end, a biblioteca React para o front-end e para o armazenamento de informações o banco de dados PostgreSQL. O resultado da aplicação indica que o produto desenvolvido é capaz de fornecer agilidade e facilidade na gestão de metas, além de assegurar a saúde deste processo para toda a organização.

Palavras-chave: Sistema; Gestão de Pessoas; Clima Organizacional; Gestão de Desempenho; OKRs.

LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxo de como definir OKRs.....	30
Figura 2 - Case de OKR.....	30
Figura 3 - Treinamentos do sistema Vydra.....	32
Figura 4 - Tutorial de como preencher a onda de perguntas.....	33
Figura 5 - Controle do desempenho do software.....	34
Figura 6 - Relatório de transparência de segurança do Vydra.....	35
Figura 7 - Casos de uso do colaborador e gestor.....	46
Figura 8 - Casos de uso do gestor.....	47
Figura 9 - Modelo conceitual do banco de dados.....	54
Figura 10 - Modelo lógico do banco de dados.....	55
Figura 11 - Código do arquivo authentication.py.....	61
Figura 12 - Continuação do código do arquivo authentication.py.....	62
Figura 13 - Código do arquivo app.py.....	63
Figura 14 - Continuação do código do arquivo app.py.....	63
Figura 15 - Código do arquivo models.py.....	64
Figura 16 - Primeira continuação do código models.py.....	65
Figura 17 - Segunda continuação do código do arquivo models.py.....	66
Figura 18 - Terceira continuação do código do arquivo models.py.....	66
Figura 19 - Código do arquivo app.jsx.....	67
Figura 20 - Código do arquivo employees.jsx.....	68
Figura 21 - Primeira continuação do código do arquivo employees.jsx.....	69
Figura 22 - Segunda continuação do código do arquivo employees.jsx.....	69
Figura 23 - Terceira continuação do código do arquivo employees.jsx.....	70
Figura 24 - Quarta continuação do código do arquivo employees.jsx.....	71
Figura 25 - Quinta continuação do código do arquivo employees.jsx.....	72
Figura 26 - Sexta continuação do código do arquivo employees.jsx.....	73
Figura 27 - Sétima continuação do código do arquivo employees.jsx.....	74
Figura 28 - Oitava continuação do código do arquivo employees.jsx.....	75
Figura 29 - Nona continuação do código do arquivo employees.jsx.....	76
Figura 30 - Décima continuação do código do arquivo employees.jsx.....	77

Figura 31 - Décima primeira continuação do código do arquivo employees.jsx.....	78
Figura 32 - Décima segunda continuação do código do arquivo employees.jsx.....	78
Figura 33 - Código do arquivo profile.styles.css.....	79
Figura 34 - Primeira continuação do arquivo profile.styles.css.....	80
Figura 35 - Segunda continuação do arquivo profile.styles.css.....	81
Figura 36 - Teste da requisição de recuperar funcionários.....	83
Figura 37 - Teste da requisição de recuperar times.....	83
Figura 38 - Teste da requisição de inserir funcionários.....	84
Figura 39 - Tela de login da aplicação.....	85
Figura 40 - Tela de painel do usuário da aplicação.....	86
Figura 41 - Tela de organograma da aplicação.....	87
Figura 42 - Tela de funcionários da aplicação.....	88
Figura 43 - Tela de times da aplicação.....	89
Figura 44 - Tela de perfil do usuário da aplicação.....	90
Figura 45 - Tela de OKR da aplicação.....	91
Figura 46 - Tela de respostas as ondas de perguntas da aplicação.....	92
Figura 47 - Tela de Analytics da aplicação.....	93

LISTA DE TABELAS

Tabela 1 - Caso de uso "Concluir Tarefas".....	48
Tabela 2 - Caso de uso "Atualizar informações de perfil"	49
Tabela 3 - Caso de uso "Alterar senha".....	50
Tabela 4 - Caso de uso "Responder ondas de perguntas"	51
Tabela 5 - Caso de uso "Criar objetivos"	52
Tabela 6 - Caso de uso "Visualizar funcionários"	101
Tabela 7 - Caso de uso "Concluir tarefas"	102
Tabela 8 - Caso de uso "Editar funcionários"	104
Tabela 9 - Caso de uso "Visualizar times"	106
Tabela 10 - Caso de uso "Editar times"	107
Tabela 11 - Caso de uso "Visualizar objetivos"	108
Tabela 12 - Caso de uso "Editar objetivos"	109
Tabela 13 - Caso de uso "Visualizar resultados-chave"	110
Tabela 14 - Caso de uso "Editar resultados-chave"	112
Tabela 15 - Caso de uso "Visualizar tarefas"	113
Tabela 16 - Caso de uso "Editar tarefas"	115
Tabela 17 - Caso de uso "Visualizar informações de perfil"	116
Tabela 18 - Caso de uso "Visualizar organograma"	117
Tabela 19 - Caso de uso "Criar times"	118
Tabela 20 - Caso de uso "Atualizar times"	119
Tabela 21 - Caso de uso "Excluir times"	120
Tabela 22 - Caso de uso "Criar funcionários"	122
Tabela 23 - Caso de uso "Atualizar funcionários"	123
Tabela 24 - Caso de uso "Excluir funcionários"	125
Tabela 25 - Caso de uso "Atualizar objetivos"	126
Tabela 26 - Caso de uso "Excluir objetivos"	128
Tabela 27 - Caso de uso "Criar resultados-chave"	129
Tabela 28 - Caso de uso "Atualizar resultados-chave"	131
Tabela 29 - Caso de uso "Excluir resultados-chave"	132
Tabela 30 - Caso de uso "Criar resultados-chave"	134

Tabela 31 - Caso de uso "Criar tarefas"	135
Tabela 32 - Caso de uso "Excluir tarefas"	137
Tabela 33 - Caso de uso "Consultar chatBox"	138

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
DDOS	Distributed Denial of Service
EX	Employee Experience
HTTPS	Hyper Text Transfer Protocol Secure
JWT	JSON Web Token
NF	Requisito não-funcional
OKR	Objectives and Key-Results
PDF	Portable Document Format
PWC	PricewaterhouseCoopers
REST	Representational State Transfer
RF	Requisito Funcional
ROI	Return on Investment
TLS	Transport Layer Security
URL	Uniform Resource Locator
USP	Universidade de São Paulo
UX	User Experience

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Justificativa	14
1.2. Hipóteses	15
1.3. Objetivo geral	16
1.3.1. Objetivos específicos.....	16
2. PLANEJAMENTO DO PROJETO.....	16
2.1. Fundamentação teórica.....	16
2.1.1. Experiência do Funcionário.....	17
2.1.2. Pesquisa Organizacional de Clima e Engajamento	18
2.1.2.1. Modelo de Sbragia.....	19
2.1.2.2. Modelo de Litwin e Stringer.....	20
2.1.3. Objectives and Key Results (OKR)	21
3. DESENVOLVIMENTO DO PROJETO.....	23
3.1. Etapas de Desenvolvimento	24
3.1.1. Planejamento.....	24
3.1.1.1. Identificação do cliente.....	24
3.1.1.1.1. Persona 01.....	24
3.1.1.1.2. Persona 02.....	25
3.1.1.1.3. Persona 03.....	25
3.1.1.2. Levantamento de recursos e custos	26
3.1.1.3. Avaliar a viabilidade	26
3.1.1.3.1. Viabilidade Técnica	26
3.1.1.3.2. Viabilidade operacional.....	27
3.1.1.3.3. Viabilidade Financeira.....	27

3.1.2.1.	Versionamento do código-fonte.....	27
3.1.2.2.	Construção da aplicação.....	28
3.1.2.3.	Criação do teste	29
3.1.3.	Implantação.....	30
3.1.3.1.	Treinamento.....	30
3.1.3.2.	Base teórica de OKR	30
3.1.3.2.1.	Tutoriais de funcionalidades	31
3.1.4.	Avaliação e Manutenção.....	34
3.2.	Critérios de inovação	36
3.3.	Divulgação do produto	37
4.	DESENVOLVIMENTO SISTEMA/SOFTWARE/APLICATIVO.....	38
4.1.	Levantamento de requisitos	38
4.1.1.1.	Módulos do sistema.....	39
4.1.2.	Identificação dos requisitos.....	39
4.1.3.	Análise e Classificação dos Requisitos.....	40
4.1.3.1.	Requisitos Funcionais	40
4.1.3.2.	Requisitos Não-Funcionais.....	44
4.2.	Casos de Uso	46
4.3.	Projeto de Banco de Dados.....	53
4.4.	Projeto Técnico.....	58
4.4.1.1.	Detalhes Técnicos.....	58
4.4.1.2.	Desenvolvimento do código	59
4.4.1.3.	Organização de arquivos da aplicação.....	59
4.4.1.3.1.	Organização de arquivos do front-end da aplicação	59
4.4.1.3.2.	Organização de arquivos do front-end da aplicação	60
4.4.1.4.	Códigos da Aplicação.....	60
4.4.2.1.	Sistema de Login e Senha.....	81

4.4.2.2.	Sistema de autenticação da API por Token JWT	81
4.4.2.3.	Ambiente seguro de host.....	82
4.4.2.4.	Protocolos HTTPS	82
4.4.2.5.	Níveis de acesso do usuário	82
4.4.3.	Testes	82
4.4.4.	Descrição das interfaces.....	84
4.5.	Verificações e Correções.....	93
4.5.1.	Testes de Usabilidade.....	93
4.5.2.	Análise e Revisão do Código.....	94
4.5.3.	Correção de Bugs.....	94
5.	CONSIDERAÇÕES FINAIS, RECOMENDAÇÕES E LIMITAÇÕES	94
6.	REFERÊNCIAS BIBLIOGRÁFICAS	96

1. INTRODUÇÃO

No contexto dinâmico e desafiador do ambiente empresarial contemporâneo, a gestão eficaz de metas e clima organizacional emerge como um fator crítico para o sucesso e sustentabilidade das organizações, com isso, este trabalho propõe a análise e o desenvolvimento de um sistema inovador de gestão de desempenho. Nesta sessão serão apresentados os fatores que justificam a existência dele, bem como quais são seus objetivos e quais são as hipóteses de benefícios esperados com sua finalização.

1.1. Justificativa

No mundo corporativo, pouco é tratado o planejamento dos objetivos e resultados-chave inclusos em uma empresa, sendo estes os pontos centrais que a farão evoluir e se destacar no mercado. A falta de centralidade e efetividade na definição e controle desses métodos faz com que a organização perca seu potencial de crescimento e que seus funcionários se sintam distantes dos interesses da empresa.

Dentro da organização, além da falta de visibilidade e das metas gerais e específicas da empresa, os colaboradores enfrentam a pouca clareza de quais são as responsabilidades e os membros de cada time, o que gera insatisfação e dificuldade no momento em que há a necessidade de acionar determinada equipe.

Tanto a falta de visibilidade quanto a satisfação dos colaboradores a respeito da empresa, seu ambiente de trabalho e a sua autoavaliação perante suas atividades faz com que não se possa ter gestão sobre a saúde das etapas estabelecidas para o alcance dos objetivos e o conforto do funcionário em participar do crescimento da empresa.

Se torna importante ressaltar a relevância profissional de se ter um sistema que auxilie os interesses da empresa, evidenciando de forma clara através de conceitos de gestão de desempenho uma maneira da organização disponibilizar, delegar e gerenciar suas metas e resultados-chave. Desta maneira é garantido o

alinhamento dos seus interesses com os dos funcionários e fazendo seu crescimento ser potencializado.

O sistema além de contribuir com o gerenciamento empresarial, também favorece as questões pessoais, acrescentando qualidade e transparência quanto a disponibilização das responsabilidades dentro das áreas. Por conseguinte, tornando o dia a dia do funcionário algo salutar, e assim fortalecendo os laços do mesmo com a corporação.

1.2. Hipóteses

- Disponibilizando um sistema onde tenha a exibição de um organograma empresarial que apresenta as responsabilidades e os funcionários de cada equipe, será possível identificar de maneira mais eficiente e transparente qual equipe ou funcionário acionar em momentos pertinentes.
- Criando um sistema onde haja a criação e o disparo semanal de perguntas estratégicas sobre a qualidade do ambiente corporativo, a empresa poderá avaliar as percepções anônimas dos funcionários sobre o cotidiano do trabalho, saúde ocupacional, autoavaliação e a satisfação com os processos de suas atividades. Desta maneira a empresa consegue avaliar de forma efetiva a qualidade do seu ambiente de trabalho, se as metas propostas estão sendo desenvolvidas de forma saudável pelas equipes, e tem a garantia que os colaboradores se sintam ouvidos, melhorando assim sua experiência dentro da empresa.
- Desenvolvendo um software de fácil usabilidade e interação, onde a organização e os gestores das equipes possam disponibilizar suas futuras metas com prazos, descrições e subtarefas personalizadas de acordo com os interesses da corporação, sendo possível que os funcionários consigam visualizar estes objetivos com dados dinâmicos da sua equipe e da organização por completo, dando visibilidade e facilidade no controle do processo de atingimento das metas da empresa.

1.3. Objetivo geral

Ampliar a qualidade da gestão de desempenho corporativo através de um sistema. Desta forma contribuindo para que o processo de criação e controle de metas seja eficiente, claro e saudável a todos da empresa.

1.3.1. Objetivos específicos

Desenvolver um espaço onde a empresa visualize seu organograma empresarial em que será exibido as responsabilidades e os membros que compõem cada time da organização.

Criar um ambiente onde os gestores poderão definir as metas gerais e específicas da empresa, além das tarefas, responsáveis e prazos a serem cumpridos. Também sendo possível visualizar ao longo do processo a porcentagem de finalização das etapas e das metas impostas;

Desenvolver um sistema que realizará o disparo semanal de perguntas estratégicas sobre a experiência do funcionário de forma anônima. Por meio desses dados será fornecido um relatório onde a empresa terá a visibilidade da qualidade do ambiente corporativo.

2. PLANEJAMENTO DO PROJETO

2.1. Fundamentação teórica

A seção de fundamentação teórica é essencial para contextualizar a relevância do projeto, embasando as escolhas metodológicas e orientando a aplicação prática dos conceitos ao longo do desenvolvimento. Com ela será evidenciado o estabelecimento sólido de uma base de conhecimento, explorando a integração entre a gestão de metas e a avaliação de clima e engajamento.

2.1.1. Experiência do Funcionário

Quando é falado sobre experiência do colaborador, trata-se de um problema muito questionado atualmente, visto que o produto sofre total interferência do trabalhador. Muitas perguntas são tomadas como base para tal assunto, tais como: será que é possível tornar o trabalho dos empregados algo além da obrigação? Se nossas vidas dependem do trabalho, não há jeito para tornar o emprego algo menos desgastante?

Apesar de não aparentar, este problema vem de muito tempo atrás. Mais especificamente, o tema Gestão de Pessoas se divide em 4 épocas (OLIVEIRA A., 2021). Na primeira, é retratada a administração científica, quando se começou a ter enfoque na eficiência, na divisão dos trabalhos e no monitoramento dos empregados, onde o principal objetivo era colocar o colaborador onde ele produzisse de forma rápida e eficiente (OLIVEIRA A., 2021). Aos poucos esta questão foi se desenvolvendo e esta preocupação “evoluiu” para outro tópico: Relações Humanas. Nesta época, foi percebido que a produtividade do trabalhador tinha muita relação com a satisfação dele com a empresa e com seus colegas de trabalho, tornando assim a cooperação entre empregados e empregadores um ponto chave nos resultados. A terceira parte é a Gestão Estratégica de Pessoas, onde seu principal ponto foi a consolidação das estratégias comerciais, fazendo com que as organizações comesçassem a ver seus funcionários como possíveis motivos de vantagens competitivas e aumento no desempenho financeiro delas. Na mais recente e última fase, o principal foco é a gestão sustentável das pessoas, na qual o principal tema abordado é a sustentabilidade na gestão de profissionais, alinhando a economia, o ambiente e os empregados da corporação, tornando-a um lugar que promova o bem estar e a aliança entre esses 3 pontos.

Com a alta do assunto Experiência do Usuário, foram feitos diversos estudos que comprovam que este está diretamente ligado a experiência do colaborador. Como indica Vinícius Oliveira (2014), pesquisas feitas entre 2013 e 2014 por consultorias de grandes empresas do mercado (Salesforce, PWC, Reffind e Gallup), comprovam que é de alta importância a atenção no

engajamento dos colaboradores, já que promove o compromisso dos funcionários com a empresa e suas entregas. Uma fórmula muito usada para fazer esta analogia é $UX=EX$, onde UX significa User Experience (Experiência do Usuário) e EX significa Employee Experience (Experiência do colaborador). Esta fórmula retrata bem o que empresas como IBM, GE e Airbnb praticam desde 2016: um modelo de trabalho onde o principal agente é o trabalhador e com objetivo de deixar uma experiência positivamente memorável a ele (OLIVEIRA V., 2014).

Como forma de identificar os problemas presentes entre os funcionários, Welbourne (2016) aponta o uso das Pulses Surveys. Consiste em pesquisas simplificadas e pontuais onde são identificadas as principais características de seus funcionários, tendo estas ligações ou não com o trabalho. Assim, fica mais fácil identificar quais os empecilhos que os colaboradores estão tendo em funções relacionadas ao trabalho, comunicação e relacionamentos com o ambiente da empresa, tornando a manutenção e ajuda algo mais natural.

2.1.2. Pesquisa Organizacional de Clima e Engajamento

A análise do clima organizacional e do engajamento dos colaboradores é de extrema importância na pesquisa da saúde corporativa e na busca pelo aprimoramento do ambiente de trabalho. Em um contexto empresarial altamente competitivo, compreender o sentimento e o envolvimento dos funcionários nas organizações tornou-se uma prioridade estratégica. Esses modelos fornecem uma base teórica para avaliar e aprimorar o bem-estar dos colaboradores e a eficácia das empresas.

Um modelo de clima e envolvimento organizacional consiste em um conjunto estruturado de conceitos e métricas cujo propósito é avaliar a percepção e o comprometimento dos colaboradores em relação às suas organizações. Tal modelo proporciona uma visão ampla do ambiente de trabalho, englobando elementos como cultura organizacional, liderança, comunicação, desenvolvimento profissional e bem-estar dos funcionários. O objetivo central reside em compreender como os

colaboradores se sentem em relação à empresa e de que forma essa percepção impacta o seu engajamento, produtividade e satisfação.

2.1.2.1. Modelo de Sbragia

O modelo de Sbragia foi publicado em 1983 por Roberto Sbragia, um renomado professor da Universidade de São Paulo (USP). Ele foi baseado no modelo de Litwin e Stringer (1968) e é considerado um modelo mais atual para as práticas de relacionamento entre colaborador e empresa. O autor busca valorizar e potencializar os esforços individuais de cada funcionário, bem como motivar as corporações a estabelecerem um plano de crescimento profissional a eles.

Este estudo utiliza 20 fatores/indicadores para medir o clima. São eles:

- Estado de tensão: avalia as ações das pessoas descrevendo se pensam de forma racional antes de agir pelas emoções;
- Conformidade exigida: avalia a quão alta é a cobrança de formalidade e obediência a normas da empresa e a quão flexíveis são seus colaboradores;
- Ênfase na participação: avalia a liberdade dos funcionários com decisões, ideias e sugestões;
- Proximidade da supervisão: avalia a cobrança da administração com as pessoas e o quanto elas conseguem exercer seus próprios métodos de trabalho;
- Consideração humana: avalia o tratamento das pessoas como humanos;
- Adequação da estrutura: avalia a consistência e a usabilidade do esquema organizacional da empresa;
- Autonomia presente: avalia como os colaboradores se sentem em relação aos seus chefes;
- Recompensas proporcionais: avalia a valorização do funcionário e a ênfase na recompensa antes das punições;
- Prestígio obtido: avalia a projeção profissional fornecida pela empresa ao funcionário;
- Cooperação existente: avalia as relações sociais e o grau de confiança do empregado com sua equipe;
- Padrões enfatizados: avalia a importância dada pelos funcionários as suas metas e a qualidade de seu trabalho;

- Atitude frente a conflitos: avalia como os colaboradores se sentem diante a opiniões opostas às suas;
- Sentimento de identidade: avalia a valorização e identificação do empregado com sua empresa;
- Tolerância existente: avalia a forma com que os erros são tratados e suportados na corporação;
- Clareza percebida: avalia a clareza das tarefas e do assunto destas para o funcionário;
- Justiça predominante: avalia a valorização do funcionário pelas suas habilidades e não por aspectos que fogem disso, como aparência, gostos, etc.;
- Condições de progresso: avalia o quanto a empresa motiva e fornece um crescimento profissional para seus colaboradores;
- Apoio logístico proporcionado: avalia a disposição do ambiente e dos instrumentos de trabalho;
- Reconhecimento proporcionado: avalia o reconhecimento de desempenhos acima do esperado/padrão;
- Forma de controle: avalia a forma com que a organização utiliza seus custos e sua produtividade em relação a ela mesma.

“Embora o conceito de clima tenda a ser usado para descrever organizações, o termo também pode ser usado para caracterizar as percepções das pessoas sobre os grupos ou equipes em que trabalham.” (Sbragia, 1983, p.30).

Embora muito abrangentes, o autor ressalta que estes indicadores podem ser classificados através de 7 fatores que interferem no trabalho do colaborador: condições de trabalho no contexto organizacional; filosofia e práticas voltadas à humanização do colaborador; engajamento da empresa com os valores do funcionário; facilidades organizacionais providas para avanço profissional; características da supervisão; atitude frente a conflitos; e calor humano (Sbragia, 1983, p.38).

2.1.2.2. Modelo de Litwin e Stringer

Motivation and Organization Climate (1968) é uma monografia escrita por George H. Litwin e Robert A. Stringer na divisão de pesquisa, do curso de Administração de Negócios da Universidade de Harvard. Na monografia é relatada a

diferença em que três empresas estadunidenses exibiam sua situação de clima organizacional e a análise desses casos ajudou na impulsão dos estudos sobre este assunto. O modelo apresentado por Litwin e Stringer (1968) traz consigo nove fatores organizacionais a serem analisados, sendo eles: estrutura, responsabilidade, cooperação, conflito, desafio, recompensa, relacionamento, identidade e padrões.

A definição desses fatores se dá por:

- Estrutura: sentimento do colaborador relativo a regras, métodos, procedimentos e demais limitadores impostos pela empresa;
- Responsabilidade: referente a autonomia na tomada de decisões e a obrigação de responder a essas decisões;
- Cooperação: apoio mútuo entre colaboradores de diferentes níveis;
- Conflito: referente a saber que a administração não teme opiniões adversas e a forma que é feita para solucionar problemas;
- Desafio: referente ao sentimento presente em tomadas de decisões difíceis;
- Recompensa: referente ao colaborador sentir-se recompensado pelo trabalho desempenhado, incluindo incentivos, promoção e remuneração;
- Relacionamento: referente à hombridade da organização e apoio entre colaboradores;
- Identidade: referente a se sentir membro da organização como parte importante para o funcionamento dela; e
- Padrões: referente aos métodos e processos adotados pela organização.

O modelo de Litwin e Stringer (1968) foi um dos precursores do assunto, inspirando e sendo utilizado como base de estudo para diversos outros que virão futuramente.

2.1.3. Objectives and Key Results (OKR)

De acordo com Niven e Lamorte (2016) o modelo de OKR é uma maneira de se trabalhar utilizando o pensamento crítico e a disciplina contínua, e que ele toma como objetivo garantir que haja uma colaboração dos funcionários em prol dos objetivos da corporação, assim impulsionando a empresa. Além de ter uma maior abrangência de entendimento, alcançando pessoas que tenham ou não facilidades com números.

A metodologia é composta por três processos principais: 1. A definição de metas inspiradoras e mensuráveis; 2. Ter a garantia que existe a progressão dos times e da organização para o estado desejado; 3. Desenvolver uma cadência que faça todos da equipe lembrarem do que se pretende alcançar.

Toda a empresa deve realizar a criação de um Objetivo e resultado-chave, e por sua vez os departamentos devem realizar a determinação de OKRs que auxiliam na meta principal da corporação, garantindo o sucesso de toda a empresa. Uma equipe pode trabalhar em cima de um único OKR ou a todos os que tiverem relação. todavia cada indivíduo dentro do time pode desenvolver objetivos próprios que refletem tanto no seu crescimento individual quanto no da empresa. (Nivem e Lamorte, 2016)

Um objetivo é uma declaração quantitativa projetada pela empresa, ele procura alavancar o engajamento da organização para o caminho almejado. De acordo com Carvalho (2020) há uma nomenclatura distinta para eles de acordo com a projeção da sua resolução, se um objetivo tem um horizonte de 5 à 10 anos é chamado de visão, caso tenha de 1 ano é chamado de estratégico e se tiver previsão de 1 trimestre é tático. Já quando se refere a resultados-chave é referido a declaração quantitativa, buscando as maneiras de se alcançar a meta proposta, quantificando aquilo que pode-se entender como vago dentro dos objetivos.

Dentro da metodologia existem processos para realizar sua completude, estes iniciam com a etapa de preparação onde será realizado a reintrodução das equipes ao sistema, posteriormente existe a definição dos objetivos quando a empresa e as equipes criam quais serão suas metas e o tempo para realizá-las, depois vem a definição de KRs (resultados-chave) onde existe a identificação dos resultados mensuráveis que indicam a finalização e alcance dos objetivos. Após o processo de preparação e definição é feito a revisão e acompanhamento dos objetivos e resultados-chave para entender como anda seus processos de atingimento, e com isso vem a etapa de finalização do ciclo onde é realizada a retrospectiva final para

saber quais resultados a metodologia trouxe e se podem haver melhorias para o próximo ciclo. (Carvalho, 2020)

Os OKRs trazem equidade e transparência sobre quais são as metas da empresa e como as equipes e indivíduos podem contribuir para atingi-las. Para Niven e Lamorte (2016) existem diversos pontos positivos na aplicação da metodologia, e alguns exemplos são:

- OKRs são de fácil compreensão o que auxilia no aumento de adesão a utilização por parte dos funcionários e da corporação;
- Se utilizado uma cadência curta há a promoção de agilidade e facilidade para mudanças, trazendo uma dinâmica estratégica nas tomadas de decisão na corporação;
- Com o método existe um foco nas questões mais importantes para a empresa, assim não existindo o desgaste das equipes em estabelecer objetivos diferentes do foco principal;
- A transparência da metodologia promove um melhor alinhamento multifuncional;
- OKRS proporcionam conversas focadas que impulsionam o engajamento corporativo; e
- Viabiliza o pensamento visionário no mercado.
- O - Opções: Listar todas as opções possíveis para atingir as metas propostas. Neste momento a quantidade de opções é priorizada.
- W - Will/Futuro: Diz respeito às ações que serão tomadas para atingir as metas propostas. Perguntas são feitas como: “O que será feito?”, “Quando será feito?”, “Por quem será feito?”.

3. DESENVOLVIMENTO DO PROJETO

O sistema de gestão de desempenho focado na gestão de metas e experiência do funcionário, tem por finalidade a viabilização da transparência na gestão das metas utilizando a metodologia de Objetivos e Resultados-Chave (OKR), além de disponibilizar um maior controle para tomadas de decisão por parte da organização através de painéis com indicadores de desempenho, podendo ser aplicado em

qualquer ramo empresarial ou empresa, trazendo também a importância do clima organizacional com base na experiência dos colaboradores da organização.

3.1. Etapas de Desenvolvimento

As etapas de desenvolvimento representam a jornada estrutural que resulta na construção de um software. Com isso, nesta seção, exploraremos as fases que compõem esse processo, delineando desde o planejamento inicial até a implementação prática e avaliação contínua do produto desenvolvido.

3.1.1. Planejamento

3.1.1.1. Identificação do cliente

Empresas de pequeno a grande porte, que entendem a relevância da gestão eficaz do seu desempenho e buscam aprimorar seus resultados em conjunto com a qualidade de seus negócios. Essas organizações compreendem a necessidade de um sistema abrangente que faça a integração da gestão de metas e a avaliação contínua da experiência do funcionário para promover um ambiente de trabalho saudável e produtivo.

Com base na análise detalhada conseguimos definir quais são as personas, ou seja, a representação fictícia do cliente que mais irá se conectar e se beneficiar das soluções propostas no projeto:

3.1.1.1.1. Persona 01

Nome: Carolina, diretora de recursos humanos

Perfil Profissional: Carolina é uma profissional que detém experiência em recursos humanos, visa aprimorar a cultura organizacional, clima e a satisfação dos funcionários da empresa na qual trabalha. Todavia tem dificuldade em auxiliar todos os times no controle destes processos e garantir que os funcionários se sintam integrados com os objetivos da organização.

Desafios: Carolina tem dificuldade em ter um bom controle da gestão de metas dos times da sua empresa, ao mesmo tempo que não consegue administrar como está a satisfação dos colaboradores e os times de forma eficiente.

Objetivos: Está na procura de um sistema que facilite a criação dos processos do método OKR, forneça visibilidade do progresso do atingimento das metas, e que faça a conexão com a coleta de dados da experiência dos funcionários.

3.1.1.1.2. Persona 02

Nome: Gustavo, Gerente de Desenvolvimento de Software

Perfil Profissional: Gustavo é um gestor comprometido que lidera o time de desenvolvimento de software em uma empresa de tecnologia. Ele possui uma visão clara das metas estratégicas da empresa e está determinado a orientar sua equipe na direção certa para alcançá-las, porém se sente frustrado quando não consegue acompanhar como está o processo de atingimento de forma clara.

Desafios: Enfrenta dificuldade na centralização e na definição dos objetivos, resultados-chave e tarefas do seu time e da organização, além de buscar uma ferramenta que traga clareza no acompanhamento do progresso do atingimento das metas estabelecidas.

Objetivos: Busca uma solução que traga a facilidade na gestão de metas, que auxilie o entendimento e a execução para sua equipe. Um local que promova a experiência positiva dos colaboradores, reconhecendo a importância do clima para um progresso saudável da corporação.

3.1.1.1.3. Persona 03

Nome: Maria, analista de Marketing.

Perfil Profissional: Maria é uma analista de marketing em uma empresa de médio porte. Ela é dedicada e comprometida, todavia às vezes se sente desanimada no alcance das metas e tem dificuldade de acompanhar quais tarefas ela deve executar. Ela também possui um grande desejo de melhorar suas habilidades, contribuir para que sua equipe e entender melhor seu papel na organização.

Desafios: Se sente perdida em meio às tarefas da equipe e seus prazos, sentindo dificuldade em manter-se organizada. Ela também gostaria de ter um

ambiente de confiança para expressar sua experiência no trabalho e ter uma visibilidade mais acessível de quem são os membros dos times da organização para que ela possa acioná-los em momentos pertinentes.

Objetivos: Busca uma solução que a ajude a gerenciar suas tarefas de maneira eficiente, onde consiga contribuir para um ambiente de trabalho mais positivo, compartilhando suas experiências e sugerindo melhorias. E onde ela tenha uma facilidade na identificação dos times e membros da corporação para se relacionar e pedir apoio de maneira mais eficaz.

3.1.1.2. Levantamento de recursos e custos

Inicialmente, foi visada a criação de uma base sólida para o projeto, considerando os recursos necessários e os custos associados, com objetivo de oferecer uma solução eficiente e escalável.

Para o banco de dados PostgreSQL, foi utilizada a plataforma Render, na qual possui o gasto de 7 dólares por mês para mantê-lo no ar. Esta escolha foi feita devido à confiabilidade e escalabilidade oferecidos pelo PostgreSQL, essenciais para futuros desenvolvimentos. No Flask, também foi utilizada a Render, porém de forma gratuita, onde não há limite de usuários e apenas um básico processamento, o que não impede um ótimo uso no Vydra. Já para o React, utilizou-se a plataforma Netlify, que fornece hospedagem de forma gratuita e permite utilizar 500GB de memória para disponibilizar o front-end da nossa aplicação.

Conforme a escalabilidade do projeto, serão projetadas melhorias que acompanharão os objetivos do projeto e a necessidade dos usuários.

3.1.1.3. Avaliar a viabilidade

Pensando no desenvolvimento de um sistema eficiente, a viabilidade foi segmentada a partir de três aspectos:

3.1.1.3.1. Viabilidade Técnica

Durante o desenvolvimento do sistema, foram adotadas as melhores práticas de programação e design, assegurando a escalabilidade do sistema para atender às demandas crescentes. Desta forma a viabilidade técnica é atestada pela qualidade da arquitetura e pela capacidade de integração e crescimento do sistema.

3.1.1.3.2. Viabilidade operacional

Viabilidade Operacional está relacionada com a capacidade do sistema de ser implementado em processos de negócio existentes. Com isso o sistema foi projetado levando em consideração uma interface do usuário intuitiva, desta forma facilitando a adoção pelos colaboradores.

3.1.1.3.3. Viabilidade Financeira

Pensando na sustentabilidade econômica do projeto, os custos associados ao desenvolvimento, implementação e manutenção do sistema foram meticulosamente estimados levando em consideração com crescimento saudável do produto.

E realizando uma análise dos processos fornecidos, avaliando os benefícios esperados, como a melhoria da eficiência organizacional, é esperado que haja um retorno sobre o investimento (ROI) significativo para as organizações que investirem no sistema, logo os resultados indicam uma projeção positiva onde os benefícios superam os custos ao longo do tempo.

3.1.2. Construção

3.1.2.1. Versionamento do código-fonte

O versionamento do código fonte desempenha um papel crucial na gestão eficiente do desenvolvimento de software, e para isso, foi utilizado o GitHub como plataforma central. O GitHub é uma plataforma de desenvolvimento colaborativo baseada em controle de versão, permitindo que equipes trabalhem de maneira

colaborativa, registrem alterações no código e coordenem o desenvolvimento de software de maneira eficaz.

Além da utilização do GitHub para o desenvolvimento também foram aproveitadas suas funcionalidades para o deploy da aplicação. Isso significa que se pode não apenas controlar as diferentes versões do código, mas também implementar essas versões em ambientes específicos de maneira simplificada e mantendo o controle de versão.

Em relação a colaboração no controle de versão, utilizou-se os recursos da plataforma para facilitar a aprovação e a revisão de novas funcionalidades. Isso inclui a verificação de mesclagem, onde as alterações propostas são revisadas e integradas de maneira organizada. A prática de revisão de código e o conjunto para cada solicitação garantem que as contribuições individuais estejam alinhadas com os padrões previamente estabelecidos, promovendo a consistência e a qualidade do código.

3.1.2.2. Construção da aplicação

Durante a fase de construção da aplicação Vydra, procurou-se escolher as tecnologias que fossem mais confortáveis e ainda que fossem muito utilizadas no mercado. Além disso, a arquitetura foi pensada para a aplicação possuir alta usabilidade, mas também ser robusta.

Para o desenvolvimento da interface, optou-se pela biblioteca React. Isso foi motivado pela presença da biblioteca em disciplinas da grade curricular do curso, pelo conhecimento prévio dos desenvolvedores e pela eficiência e dinamicidade do React na criação de interfaces.

Para o desenvolvimento da API REST, foi escolhido o framework Flask. Esta preferência se deu pela simplicidade, agilidade e integridade do framework, tornando o desenvolvimento do back-end mais fluido. Além disso, a experiência de parte dos desenvolvedores também contribuiu para a utilização do Flask.

No que diz respeito ao banco de dados, o PostgreSQL foi selecionado visando ter maior robustez, conformidade, facilidade e consistência no gerenciamento e disponibilização dos dados.

O controle de versão do código-fonte foi realizado utilizando o sistema Git, permitindo que os membros da equipe tivessem total controle sobre o código e ainda facilitando o rastreamento de alterações no desenvolvimento.

3.1.2.3. Criação do teste

Durante o desenvolvimento do sistema, a criação dos testes desempenhou um papel crucial na validação e aperfeiçoamento das funcionalidades propostas. Antes da aplicação do teste, foram estabelecidos quais seriam os seus objetivos de forma clara, isso incluiu a usabilidade do sistema, a identificação de eventuais falhas e bugs, e a verificação da capacidade do sistema entregar o que foi planejado.

Com base no levantamento dos objetivos, entendeu-se que tais fatores deveriam ser feitos na fase de teste:

- Teste dos endpoints do sistema com o objetivo de garantir que os dados encaminhados e o tempo de resposta do serviço estejam adequados e eficientes;
- Teste de usabilidade da parte gráfica do sistema, simulando a utilização dele, com o objetivo de verificar a experiência, facilidade de uso e a compreensão das soluções fornecidas;
- Teste com a base de dados, alterando os valores e analisando o comportamento do sistema com diferentes tipos de estrutura de dados;
- Teste do servidor, analisando possíveis bugs da aplicação no ar com o objetivo de mitigar falhas durante a utilização do serviço;
- Realizar o desenvolvimento do código do sistema de forma orientada a teste, assim garantindo a qualidade de cada entrega individualmente.

3.1.3. Implantação

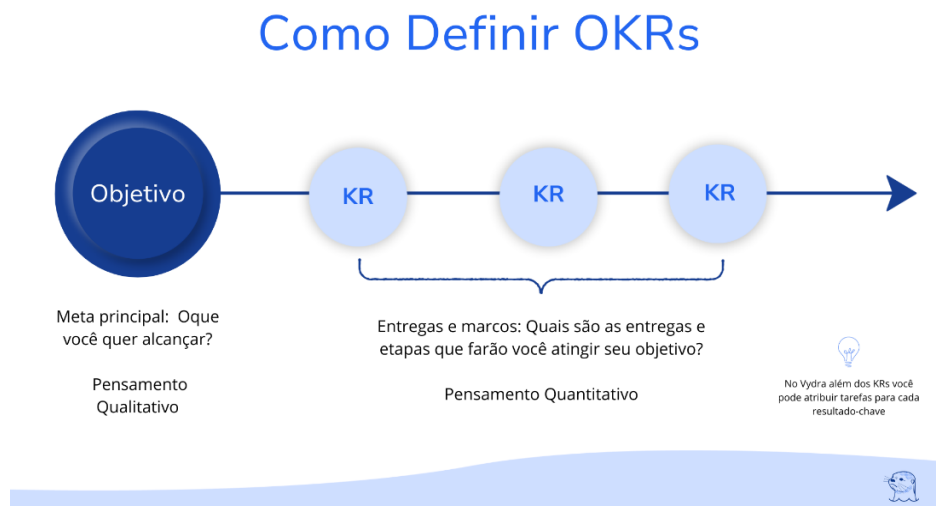
3.1.3.1. Treinamento

Treinamentos e materiais de apoio são muito relevantes para sistemas corporativos, pois eles facilitam a adaptabilidade dos colaboradores e da empresa na utilização da aplicação. O sistema Vydra foi desenvolvido pensando em fornecer uma experiência facilitada e intuitiva aos usuários, todavia foi reconhecido a importância de fornecer conteúdos que possam auxiliar os novos clientes. Nesta sessão serão apresentados quais são os treinamentos fornecidos.

3.1.3.2. Base teórica de OKR

Para realizar a utilização do sistema de maneira eficiente, recomenda-se a aplicação prévia de treinamentos acerca da metodologia OKR para todos os responsáveis por tomadas de decisões estratégicas da organização. Para auxiliar o início desta preparação foi criado os seguintes materiais de apoio:

Figura 1 - Fluxo de como definir OKRs.



Fonte: elaborado pelo autor (2023)

Figura 2 - Case de OKR.

Case - OKR



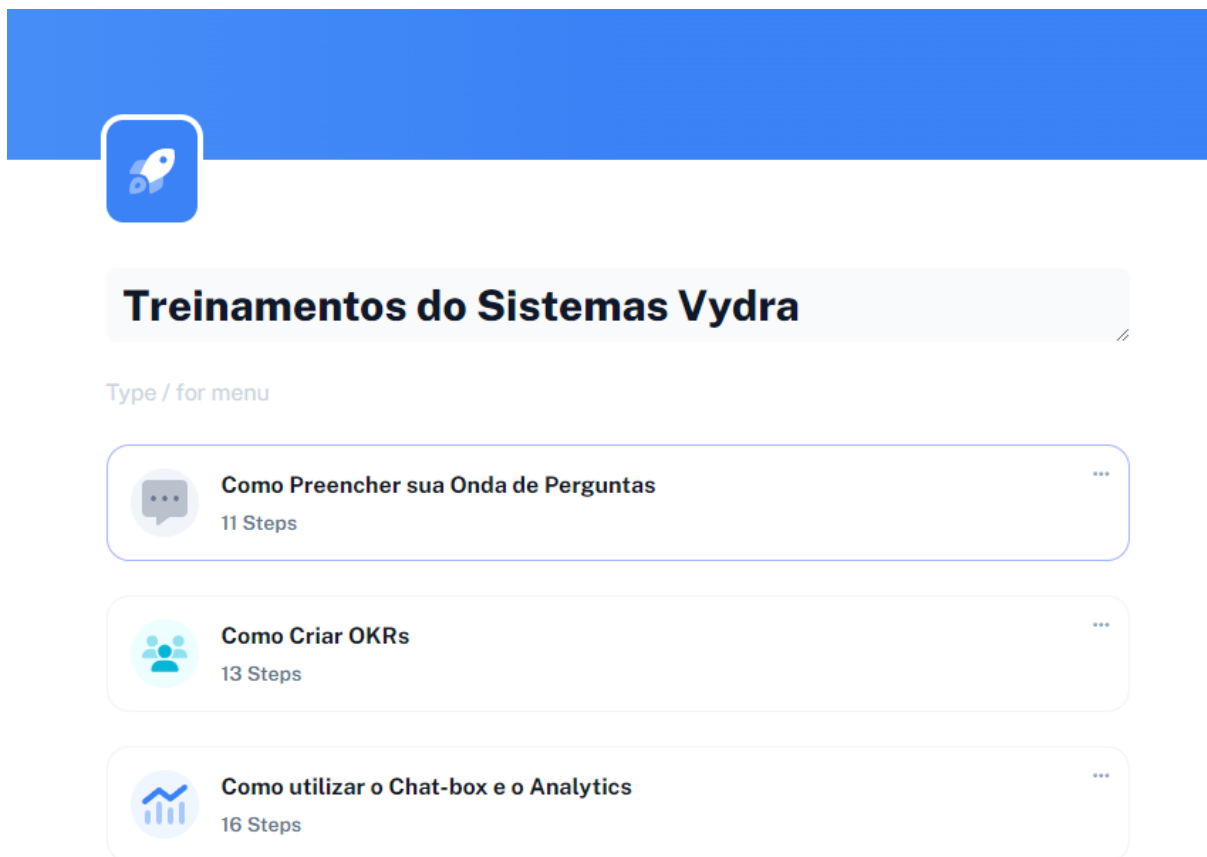
Fonte: elaborado pelo autor (2023)

3.1.3.2.1. Tutoriais de funcionalidades

Foram fornecidos passo a passo para o acesso das principais funcionalidades do sistema, com o objetivo de mitigar dúvidas dos usuários na utilização das funcionalidades.

Para a criação dos materiais foi utilizado a plataforma Scribeshow, com ela é possível centralizar os tutoriais e criá-los com uma maior facilidade. Os clientes podem acessar o material via link ou através de um PDF. Segue abaixo o local de centralização dos treinamentos:

Figura 3 - Treinamentos do sistema Vydra.



Fonte: elaborado pelo autor (2023)

Visando o fornecimento de um processo harmônico e indutivo no material é descrito o passo a passo que o usuário deve fazer para realizar uma tarefa dentro do sistema. Abaixo segue a representação de como é o treinamento disponibilizado:

Figura 4 - Tutorial de como preencher a onda de perguntas.



Fonte: elaborado pelo autor (2023)

Como demonstrado na imagem acima o treinamento além de fornecer em texto do processo a ser seguido, ele exemplifica em imagens e ícones onde o usuário deve clicar ou preencher as informações para concluir determinado objetivo dentro do sistema.

Com isso, estes materiais são o apoio necessário para iniciar a utilização do sistema na organização, todos os recursos foram planejados visando uma melhor

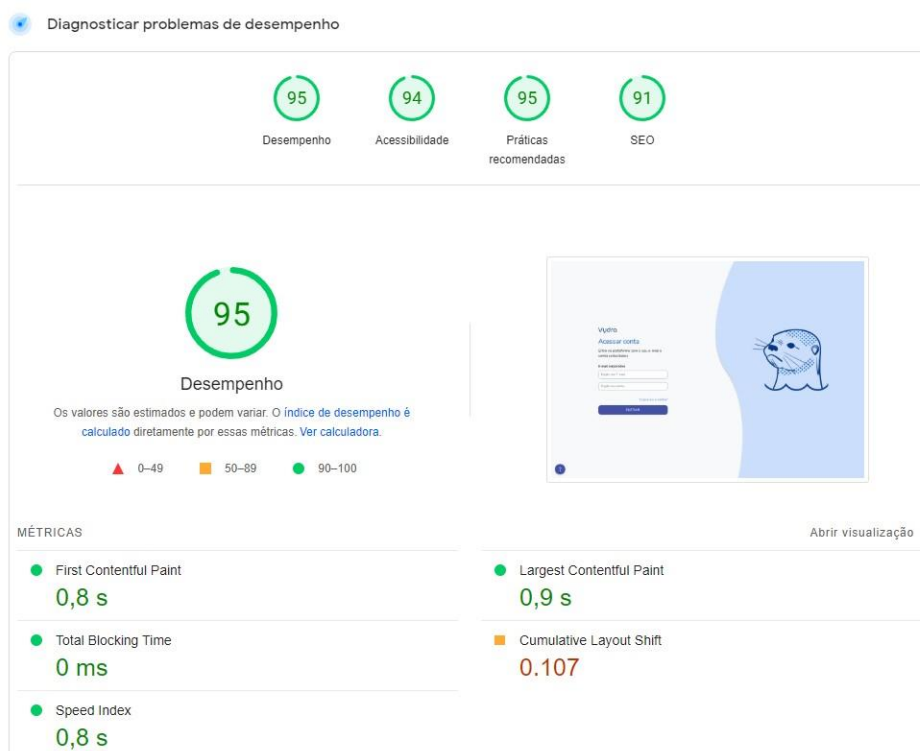
utilização do sistema, porém é importante ressaltar que o objetivo é estar sempre em alinhamento com os clientes para melhor entender suas necessidades e dúvidas, e assim enriquecer os treinamentos fornecidos.

3.1.4. Avaliação e Manutenção

A qualidade do produto é resultado de um compromisso com a excelência e a satisfação dos usuários. Desde a idealização até a implementação, cada aspecto foi cuidadosamente gerenciado para garantir que exista um produto de alta qualidade e durabilidade.

O planejamento de design foi pensado para entregar instintividade, todavia a facilidade não compromete a profundidade das funcionalidades, o que proporciona uma experiência mais agradável e produtiva aos usuários. Além da realização do planejamento foram feitos testes de qualidade rigorosos ao longo do desenvolvimento do projeto, um exemplo deles foi a utilização do PageSpeed Insights que fornece uma avaliação da velocidade e qualidade das páginas. Segue abaixo uma imagem ilustra o retorno do teste realizado:

Figura 5 - Controle do desempenho do software



Fonte: elaborado pelo autor (2023)

A segurança é um fator crucial para a avaliação de qualidade, com isso dentro do projeto foram implementados protocolos de segurança para assegurar os dados e o acesso a funcionalidades do sistema. Abaixo segue um exemplo dos testes aplicados, onde é disponibilizado a URL de acesso à aplicação e o Google retorna um relatório de transparência sobre a segurança na navegação do site:

Figura 6 - Relatório de transparência de segurança do Vydra

The screenshot shows the Google Transparency Report interface. At the top, it says "Google Relatório de Transparência" with a menu icon. Below that, there are two tabs: "Visão geral" and "Status do site", with the latter being selected. The main heading is "Status do site no Navegação segura". A descriptive paragraph explains that Google's Safe Browsing technology analyzes billions of URLs daily to find insecure websites. Below this is a search bar titled "Verificar o status do site" containing the URL "https://vydra.netlify.app/". A green checkmark icon indicates the status: "Status atual: Nenhum conteúdo não seguro foi encontrado". At the bottom, there is a section for "Informações do site" stating the last update was on Nov 23, 2023, and that security can change over time.

Fonte: elaborado pelo autor (2023)

Sobre a manutenção do produto foi pensando em garantir um suporte técnico eficiente, onde será oferecido atendimento para retirar dúvidas dos usuários. E garantir a comunicação aos clientes sobre as atualizações de forma transparente, visando que todos estejam cientes das melhorias que estão sendo realizadas.

Com isso após o desenvolvimento, para garantir a continuidade da qualidade será implementado o monitoramento em tempo real, rastreando o desempenho dos sistemas com o objetivo de identificar possíveis anomalias e proativamente resolvê-las. Além de manter a avaliação contínua de segurança e coletar os feedbacks dos usuários para manter uma melhor usabilidade do sistema.

Com as observações pautadas pode-se concluir que a qualidade do projeto foi atingida de acordo com os critérios previamente planejados, de forma que se aplicado a manutenção de qualidade dos processos o sistema continuará entregando usabilidade facilitada e produtividade para os clientes.

3.2. Critérios de inovação

Em um mercado onde existem diversos sistemas de gestão empresarial, o nosso diferencial está na integração inovadora da gestão de metas com o clima e engajamento. Em geral, frequentemente é oferecido soluções fragmentadas, como o estabelecimento de metas, sem a integração significativa com os indicadores de clima organizacional. Com isso, nosso sistema unifica essas áreas críticas, permitindo que as metas estejam em sintonia com a saúde do clima organizacional.

O organograma empresarial normalmente oferecido se limita a representar a estrutura hierárquica dos times da empresa, sem destacar de forma clara as responsabilidades de cada equipe e quais são seus colaboradores. Em contraste, a solução proposta não traz somente a representação, mas também atribui as responsabilidades de forma transparente, promovendo uma melhor compreensão da dinâmica das equipes, assim contribuindo para uma colaboração mais eficaz.

O ambiente de analytics é um dos pontos focais das soluções propostas, enquanto muitos concorrentes oferecem relatórios padrões, nossa solução proporciona uma análise avançada dos dados dos times. Indo além de simples métricas, mas oferecendo insights valiosos para orientar a corporação em melhores decisões estratégicas.

O chat-box desenvolvido com IA representa uma funcionalidade única da nossa abordagem, enquanto outros produtos apenas mostram os dados do progresso de metas e da pesquisa organizacional, nosso sistema fornece auxílio para lidar com esses resultados, logo ele entende a dor do gestor e fornece em tempo real insights de como melhorar seus processos.

Em resumo, nossa inovação e diferencial estão representados na sinergia entre a gestão de metas e clima e engajamento, na transparência organizacional, na assistência instantânea através da inteligência artificial e na análise avançada oferecida no ambiente de análise. Ao integrar esses elementos, oferecemos uma solução completa e coesa que se destaca entre as ofertas convencionais disponíveis no mercado de gestão empresarial.

3.2.1. Desenvolvimento de um Modelo Próprio de Pesquisa de Clima e Engajamento

Como apresentado no apêndice A deste documento, foi criado um modelo próprio de pesquisa de clima e engajamento com base nas referências teóricas analisadas, tornando-o flexível e aplicável em diversas situações. Oferecemos uma solução padronizada que mantém sua relevância em uma variedade de contextos organizacionais.

As dimensões de análise e suas perguntas foram cuidadosamente alinhadas com as melhores práticas, garantindo sua eficácia em avaliar o clima e o engajamento, independentemente do ambiente específico da empresa. Essa abordagem permite a aplicação do modelo em diferentes organizações, tornando-o uma ferramenta valiosa para aprimorar a gestão de desempenho e a satisfação dos funcionários em diversos cenários.

3.3. Divulgação do produto

No presente capítulo, serão apresentadas as estratégias de marketing para o lançamento do sistema Vydra no mercado, visando destacar suas características inovadoras e reter a atenção do público-alvo.

Slogan: Compreendendo a importância de uma mensagem concisa e impactante, foi pensado o slogan “Do planejamento à performance: Vydra, sua jornada para o sucesso organizacional”. Ele encapsula a proposta de valor do sistema desenvolvido, destacando sua diversidade e impacto nas estratégias empresariais.

Webinars Especializados: A divulgação será amplificada por meio de webinars especializados, onde serão abordados casos de sucesso, demonstrando como o Vydra pode auxiliar de forma efetiva e ser incorporado nas estratégias de gestão.

Artigos e Blog Posts: Para passar uma visão especializada sobre o assunto, serão criados conteúdos informativos que abordam os desafios enfrentados pelas empresas e como o produto oferece soluções inovadoras. Estes textos serão

baseados em análises técnicas detalhadas, evidenciando a arquitetura do sistema e suas principais vantagens.

Redes Sociais: Entendemos que a presença em redes sociais será uma extensão estratégica, com conteúdos técnicos compartilhados regularmente na linguagem do nosso público principal. Hashtags específicas serão utilizadas para atrair a atenção dos profissionais relacionados e engajar potenciais usuários.

Além dos principais tópicos abordados foi pensando em realizar parcerias estratégicas, visando ampliar a captação de clientes. Além da disponibilização inicial de testes grátis com prazos estimados, para que os possíveis clientes possam verificar as principais funcionalidades do produto e investir em garantia de satisfação fazendo com que os nossos usuários auxiliem na divulgação do sistema.

Este planejamento de marketing foi planejado considerando desenvolver uma aceitação sólida no mercado de gestão empresarial e clima e engajamento, dando enfoque para as questões técnicas que tornam o produto único.

4. DESENVOLVIMENTO SISTEMA/SOFTWARE/APLICATIVO

4.1. Levantamento de requisitos

4.1.1. Definição do sistema

O sistema de gestão de desempenho proposto tem como objetivo principal realizar a integração da gestão de metas com a avaliação contínua da experiência do funcionário. Onde na gestão de desempenho é enfatizado a clareza, eficiência e a saúde no processo de criação e monitoramento das metas estabelecidas. Simultaneamente, o sistema visa proporcionar uma análise regular e precisa acerca da experiência dos funcionários, garantindo uma compreensão sobre o clima corporativo.

Este sistema será desenvolvido com base nos princípios de gestão, incorporando o método OKR para estabelecer e monitorar as metas da organização.

Além de adotar uma abordagem estruturada para a coleta de informações sobre a experiência do funcionário, utilizando um modelo de pesquisa de clima próprio, promovendo a transparência e identificação de áreas de melhoria no ambiente de trabalho.

4.1.1.1. Módulos do sistema

- Gerenciamento de metas: Módulo que fornece o cadastro de metas utilizando o método de gestão OKR, bem como o acompanhamento delas e um painel personalizado com dados do time do usuário.
- Módulo de organograma: Módulo que fornece a estrutura de times e funcionários cadastrada no sistema por meio de um organograma empresarial.
- Módulo de clima e engajamento: Fornece o ambiente de pesquisa organizacional, por meio de perguntas semanais com base no modelo desenvolvido.

4.1.2. Identificação dos requisitos

O levantamento de requisitos para o sistema levou em consideração os seguintes pontos:

- a) Gestão de Metas:
 - Definição de objetivos estratégicos dos times.
 - Estabelecimento de resultados-chave (Key Results) atrelado aos objetivos.
 - Atribuição de tarefas, responsáveis e prazos associados a cada meta organizacional.
 - Visualização do progresso das metas em tempo real.
- b) Experiência do Funcionário:
 - Desenvolvimento de questionários semanais sobre a experiência do funcionário.
 - Coleta de dados para avaliação contínua do clima organizacional.
 - Espaço de análise para identificação geral dos dados coletados.

c) Integração e Usabilidade:

- Interface intuitiva para facilitar a criação e acompanhamento dos OKRs.
- Integração eficiente entre a gestão de metas e a avaliação da experiência do funcionário.

d) Escalabilidade e Adaptação:

- Capacidade de expansão para atender empresas de diferentes tamanhos.
- Flexibilidade para se adaptar a diferentes estruturas organizacionais e setores.

4.1.3. Análise e Classificação dos Requisitos

4.1.3.1. Requisitos Funcionais

- **RF001 – CADASTRO DE FUNCIONÁRIOS**

Descrição: O sistema permitirá que usuários gestores cadastrem funcionários, inserindo nome, sobrenome, e-mail, data de nascimento, time, cargo e nível de acesso, classificando-os como gestor e usuário comum. O usuário deve ser criado de forma automática juntamente com o cadastro de funcionário e sua senha gerada de forma automática, podendo ser alterada pelo usuário em seu perfil.

- **RF002 - ATUALIZAÇÃO DE FUNCIONÁRIOS**

Descrição: O sistema permitirá que usuários possam ser atualizados por usuários gestores, modificando nome, sobrenome, e-mail, data de nascimento, time, cargo e nível de acesso.

- **RF003 - DELEÇÃO DE FUNCIONÁRIOS**

Descrição: O sistema permitirá que funcionários possam ser deletados por usuários gestores, deletando juntamente seu cadastro de usuário do sistema.

- **RF004 - LOGIN DE USUÁRIOS**

Descrição: O sistema permitirá o login de usuários e os redireciona para a página inicial. Onde será apresentado um dashboard ao usuário. O login deverá ser realizado com o e-mail e senha do usuário.

- RF005 - CONTROLE DE ACESSO ÀS FUNCIONALIDADES

Descrição: O sistema autorizará o controle de acesso aos diferentes tipos de permissões, de acordo com o nível de acesso presente no perfil do usuário.

- RF006 – CADASTRO DE TIMES

Descrição: O sistema permitirá que usuários gestores criem seus times. Nesse cadastro estará incluso: nome, descrição e time responsável (se for o caso).

- RF007 - ATUALIZAÇÃO DE TIMES

Descrição: O sistema permitirá que usuários gestores possam atualizar os times, podendo modificar o nome, descrição e time responsável (se for o caso).

- RF008 - DELEÇÃO DE TIMES

Descrição: O sistema permitirá que usuários gestores possam deletar os times.

- RF009 – CADASTRO DE OBJETIVOS

Descrição: O sistema permitirá que usuários gestores possam cadastrar objetivos a serem cumpridos pelos funcionários e times. Nesse cadastro estará incluso: nome do objetivo, descrição do objetivo, data de término e responsável.

- RF010 – ATUALIZAÇÃO DE OBJETIVOS

Descrição: O sistema permitirá que usuários administradores possam atualizar objetivos, podendo modificar o nome do objetivo, descrição, data de término, responsável e marcação de finalização do objetivo.

- RF011 - DELEÇÃO DE OBJETIVOS

Descrição: O sistema permitirá que usuários gestores possam deletar objetivos.

- RF012 – CADASTRO DE RESULTADOS-CHAVE

Descrição: O sistema permitirá que usuários gestores criem os resultados-chave a serem cumpridos. Nesse cadastro estará incluso: nome do resultado-chave, descrição do resultado-chave, data de término e responsável.

- RF013 - ATUALIZAÇÃO DE RESULTADOS-CHAVE

Descrição: O sistema permitirá que resultados-chave possam ser modificados por usuários gestores, podendo alterar o nome do resultado-chave, descrição do resultado-chave, data de término, responsável e a finalização do resultado-chave.

- RF014 - DELEÇÃO DE RESULTADOS-CHAVE

Descrição: O sistema permitirá que usuários gestores possam deletar resultados-chave.

- RF015 – CADASTRO DE TAREFAS

Descrição: O sistema permitirá que usuários gestores criem as tarefas a serem cumpridas por seus liderados. Nesse cadastro estará incluso: nome da tarefa, descrição da tarefa e data limite.

- RF016 - ATUALIZAÇÃO DE TAREFAS

Descrição: O sistema permitirá que tarefas possam ser modificadas por usuários gestores, podendo alterar o nome da tarefa, descrição da tarefa, data de término, responsável e a finalização da tarefa.

- RF017 - DELEÇÃO DE TAREFAS

Descrição: O sistema permitirá que tarefas possam ser deletadas por usuários gestores.

- RF018 – FORNECER PERGUNTAS SOBRE O BEM-ESTAR

Descrição: O sistema emitirá perguntas estratégicas a serem respondidas semanalmente pelos funcionários de cada time.

- RF019 – ACESSO AO PERFIL DE USUÁRIO

Descrição: O sistema permitirá que os usuários possam acessar sua página de perfil de usuário, contendo as informações do funcionário e usuário.

- RF020 - ATUALIZAÇÃO DE DADOS DO FUNCIONÁRIO

Descrição: O sistema permitirá que ao acessar o perfil, o funcionário edite seus dados (nome e sobrenome).

- RF020 - ATUALIZAÇÃO DE SENHA DO USUÁRIO

Descrição: O sistema permitirá que ao acessar o perfil, o usuário edite sua senha, fornecendo a nova senha e uma confirmação para validar a atualização.

- RF021 - CONTROLE DE AUTENTICAÇÃO DO USUÁRIO

Descrição: O sistema deverá confirmar a veracidade da autenticação do usuário por meio de um token JWT.

- RF022 – GERAÇÃO DE DASHBOARDS

Descrição: O sistema deverá gerar dashboards analíticos para que os responsáveis pela empresa visualizem e os utilizem para tomada de decisões estratégicas.

- RF023 – ACOMPANHAMENTO DOS OBJETIVOS

Descrição: O sistema deverá disponibilizar o progresso dos objetivos fornecidos pela empresa no dashboard do usuário, de forma transparente, para que todos os funcionários possam ver e ter consciência situacional.

- RF024 – LANÇAMENTO DE PERGUNTAS SOBRE CLIMA

Descrição: O sistema deverá realizar o lançamento de perguntas sobre o clima organizacional a serem respondidas pelos funcionários da empresa, o lançamento das perguntas será realizado semanalmente.

- RF025 – ACESSO A VISÃO GERAL DO TIME

Descrição: O sistema deverá disponibilizar uma visão geral do time e que o funcionário está inserido.

- RF026 – ACESSO AO ORGANOGRAMA DA EMPRESA

Descrição: O sistema deverá disponibilizar um organograma contendo times, cargos e funcionários da empresa.

- RF027 – CHAT COM PERGUNTAS E RESPOSTAS SOBRE ANALYTICS

Descrição: O sistema deverá disponibilizar um chatbox com perguntas e respostas relacionadas à área de analytics do sistema. Esse chatbox deverá ser desenvolvido com auxílio de IA para facilitar e automatizar a solução das perguntas dos usuários.

4.1.3.2. Requisitos Não-Funcionais

- NF001 – ALTA DISPONIBILIDADE DO SISTEMA

Categoria: Disponibilidade e Continuidade de Serviço

Descrição: O sistema deverá ter alta disponibilidade para uso, sempre avisando antes de realizar qualquer manutenção ou atualização da aplicação, e estas pausas previstas na aplicação devem ser realizadas em no máximo 3h e de preferência num momento em que não implique negativamente no uso por parte da empresa.

- NF002 – DESEMPENHO DE MÉDIA PERFORMANCE

Categoria: Desempenho e Eficiência

Descrição: O sistema deverá possuir um desempenho de no mínimo média performance, fazendo com que o tempo de resposta de qualquer funcionalidade não influencie negativamente na jornada de trabalho do funcionário ou administração.

- NF003 – INTERFACE DE FÁCIL UTILIZAÇÃO

Categoria: Experiência do Usuário (UX/UI)

Descrição: O sistema deve possuir uma interface intuitiva e de fácil manejo, aplicando conceitos de UX/UI para melhorar a experiência do usuário.

- NF004 – ALTA COMPATIBILIDADE COM NAVEGADORES

Categoria: Compatibilidade de Plataforma

Descrição: O sistema deve possuir alta compatibilidade com navegadores web diferentes, mantendo a qualidade da execução da aplicação independentemente do navegador utilizado.

- NF005 - RELATÓRIOS CONCISOS

Categoria: Análise e Relatórios

Descrição: O sistema deverá oferecer relatórios completos e concisos por meio de sua análise de dados referentes ao desempenho empresarial, emitindo dashboards e gráficos eficazes.

- NF006 - SEGURANÇA DA INFORMAÇÃO

Categoria: Proteção de Dados e Segurança.

Descrição: O sistema deverá possuir mecanismos de segurança, como criptografia hash em senhas e tokens de verificação de usuário, afim de garantir a segurança da informação na aplicação.

- NF007 – TOLERANCIA A FALHAS

Categoria: Confiabilidade.

Descrição: O sistema deve ser capaz de lidar com falhas de componentes individuais, garantindo que a falha em um módulo não comprometa toda a aplicação, mantendo a continuidade do serviço.

- NF008 – MANUTENIBILIDADE DO CÓDIGO

Categoria: Manutenção de Software.

Descrição: O código do sistema deve ser estruturado e documentado de forma clara, facilitando a manutenção e atualizações futuras por parte dos desenvolvedores.

- NF009 – TEMPO DE CARREGAMENTO DE PÁGINAS

Categoria: Desempenho e Eficiência.

Descrição: As páginas do sistema devem ter um tempo de carregamento rápido, proporcionando uma experiência ágil e responsiva ao usuário.

4.2. Casos de Uso

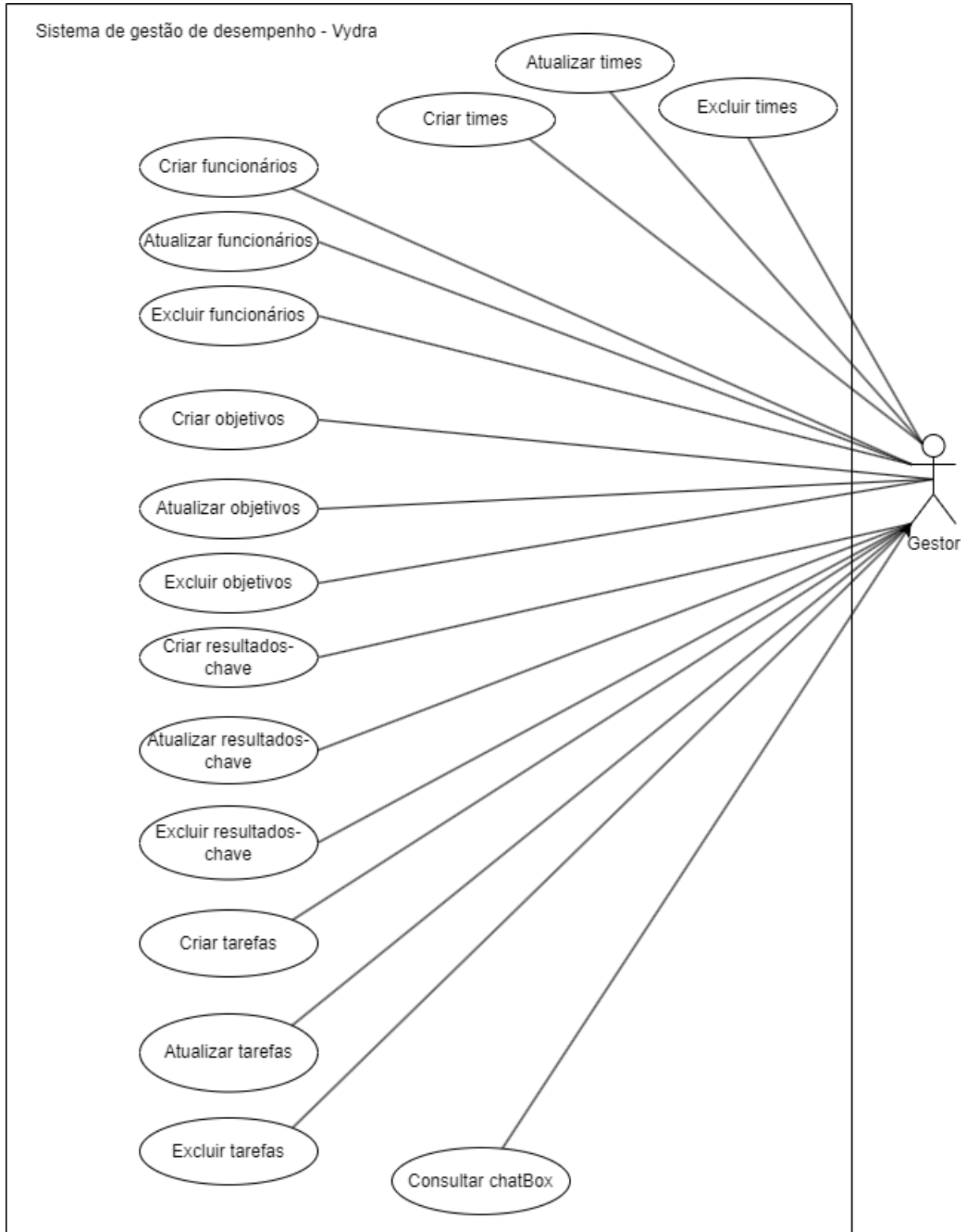
Segue abaixo a representação do diagrama de caso de uso do sistema, contemplando os dois tipos de usuários, colaboradores e gestores.

Figura 7 - Casos de uso do colaborador e gestor



Fonte: elaborado pelo autor (2023)

Figura 8 - Casos de uso do gestor



Fonte: elaborado pelo autor (2023)

4.2.1. Especificação dos Casos de Uso

Abaixo será apresentado como foi construído a especificação dos casos de uso do sistema, evidenciando o processo lógico para utilização das funcionalidades propostas. As especificações de todos os casos de uso do software podem ser analisadas no apêndice B deste documento.

Tabela 1 - Caso de uso "Concluir Tarefas".

Nome: Concluir tarefas
<p>Atores:</p> <ul style="list-style-type: none">• Atores Primários: Usuário colaborador e usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none">• Permitir que colaboradores e gestores concluam tarefas. <p>Pré-condições:</p> <ul style="list-style-type: none">• O usuário está autenticado no sistema. <p>Pós-condições:</p> <ul style="list-style-type: none">• As alterações feitas pelo usuário são refletidas na listagem de tarefas.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário seleciona a opção "Detalhes" no menu de opções do resultado-chave.2. O sistema exibe a lista de tarefas, incluindo o nome, a data prevista de cada e seu respectivo menu de opções, podendo concluir a tarefa.

<ol style="list-style-type: none"> 3. O usuário seleciona a opção “Concluir” no menu de opções. 4. O sistema exibe uma mensagem questionando se o usuário tem certeza que deseja concluir a tarefa. 5. O usuário seleciona a opção “Sim”. 6. O sistema atualiza os dados da tarefa conforme a modificação feita.
<p>Fluxos Alternativos:</p> <p>1a Usuário não autenticado:</p> <ul style="list-style-type: none"> ● O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 2 - Caso de uso "Atualizar informações de perfil".

Nome: Atualizar informações de perfil
<p>Atores:</p> <ul style="list-style-type: none"> ● Atores Primários: Usuário colaborador e usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none"> ● Permitir que colaboradores e gestores atualizem suas próprias informações, tais como nome, sobrenome e senha. <p>Pré-condições:</p> <ul style="list-style-type: none"> ● O usuário está autenticado no sistema. <p>Pós-condições:</p> <ul style="list-style-type: none"> ● As alterações feitas pelo usuário são na tela de perfil.
<p>Fluxo Principal:</p> <ol style="list-style-type: none"> 1. O usuário seleciona a imagem de seu ícone.

2. O sistema exibe as informações correspondentes de seu perfil.
3. O usuário realiza as alterações desejadas.
4. O usuário seleciona o botão “Salvar”.
5. O sistema atualiza os dados do perfil conforme as modificações feitas.

Fluxos Alternativos:

1a Usuário não autenticado:

- O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 3 - Caso de uso "Alterar senha".

Nome: Alterar senha

Atores:

- Atores Primários: Usuário colaborador e usuário gestor

Objetivo:

- Permitir que colaboradores e gestores atualizem suas senhas de login.

Pré-condições:

- O usuário está autenticado no sistema.

Pós-condições:

- A senha de login do funcionário é atualizada.

Fluxo Principal:

1. O usuário seleciona a opção “Alterar senha” na tela de perfil.

2. O sistema redireciona para a tela de alteração de senha.
3. O usuário realiza as alterações desejadas.
4. O usuário seleciona o botão "Salvar".
5. O sistema atualiza a senha de login do funcionário.

Fluxos Alternativos:

1a Usuário não autenticado:

- O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 4 - Caso de uso "Responder ondas de perguntas".

Nome: Responder ondas de perguntas

Atores:

- Atores Primários: Usuário colaborador e usuário gestor

Objetivo:

- Permitir que colaboradores e gestores respondam as perguntas sobre clima organizacional.

Pré-condições:

- O usuário está autenticado no sistema.

Pós-condições:

- As respostas são gravadas no banco de dados de forma anônima.

- O usuário fica impossibilitado de responder a onda de perguntas até que uma nova seja lançada.

Fluxo Principal:

1. O usuário acessa a seção "Ondas" no menu principal.
2. O usuário seleciona o botão "Responder".
3. O usuário responde todas as perguntas informadas com uma nota de 1 a 7 de acordo com sua opinião.
4. O usuário seleciona o botão "Finalizar Onda".
5. O sistema registra as respostas de forma anônima.

Fluxos Alternativos:

1a Usuário não autenticado:

- O sistema redireciona o usuário para a tela de login.

2a Usuário já respondeu a onda de perguntas:

- O sistema exibe a mensagem "Não existe uma onda disponível no momento".

Fonte: elaborado pelo autor (2023)

Tabela 5 - Caso de uso "Criar objetivos".

Nome: Criar objetivos

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores criem os objetivos da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de objetivos.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor seleciona a opção “+ Objetivo”;
3. O gestor preenche o nome, data limite e descrição do novo objetivo;
4. O gestor seleciona o botão “Salvar”; e
5. O novo objetivo já estará listado na lista de objetivos da empresa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na criação do objetivo:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

3a Informações faltantes no formulário:

- O sistema não permite a criação do novo objetivo e retorna a página de OKRs.

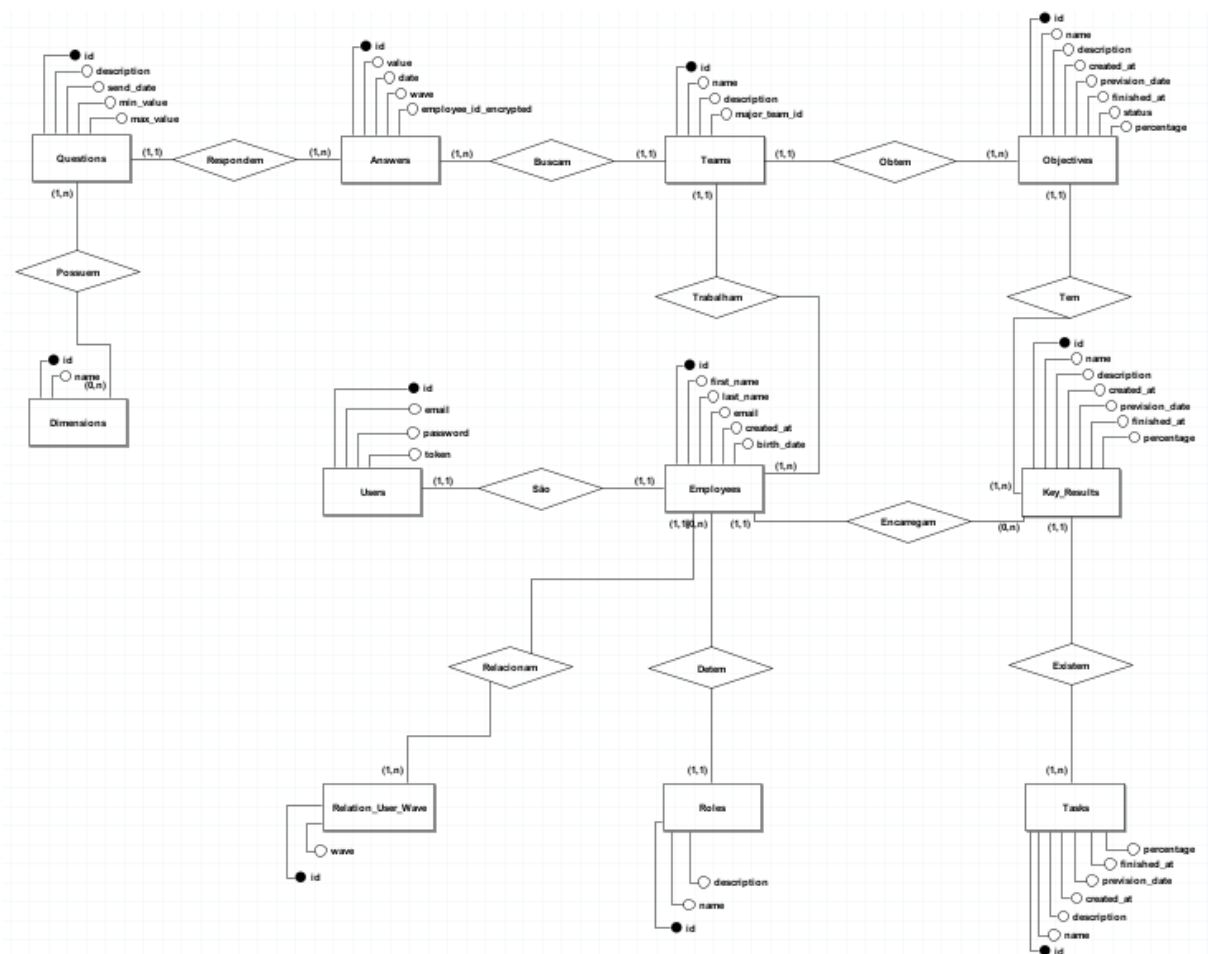
Fonte: elaborado pelo autor (2023)

4.3. Projeto de Banco de Dados

4.3.1. Modelo Conceitual

O modelo conceitual dá ênfase em exibir os atributos, relacionamentos e sua cardinalidade. O banco de dados do sistema dispõe de onze tabelas, cada tabela possuindo seus respectivos atributos e relacionamentos entre demais tabelas do banco de dados. Segue abaixo a imagem do modelo conceitual do banco de dados.

Figura 9 - Modelo conceitual do banco de dados.

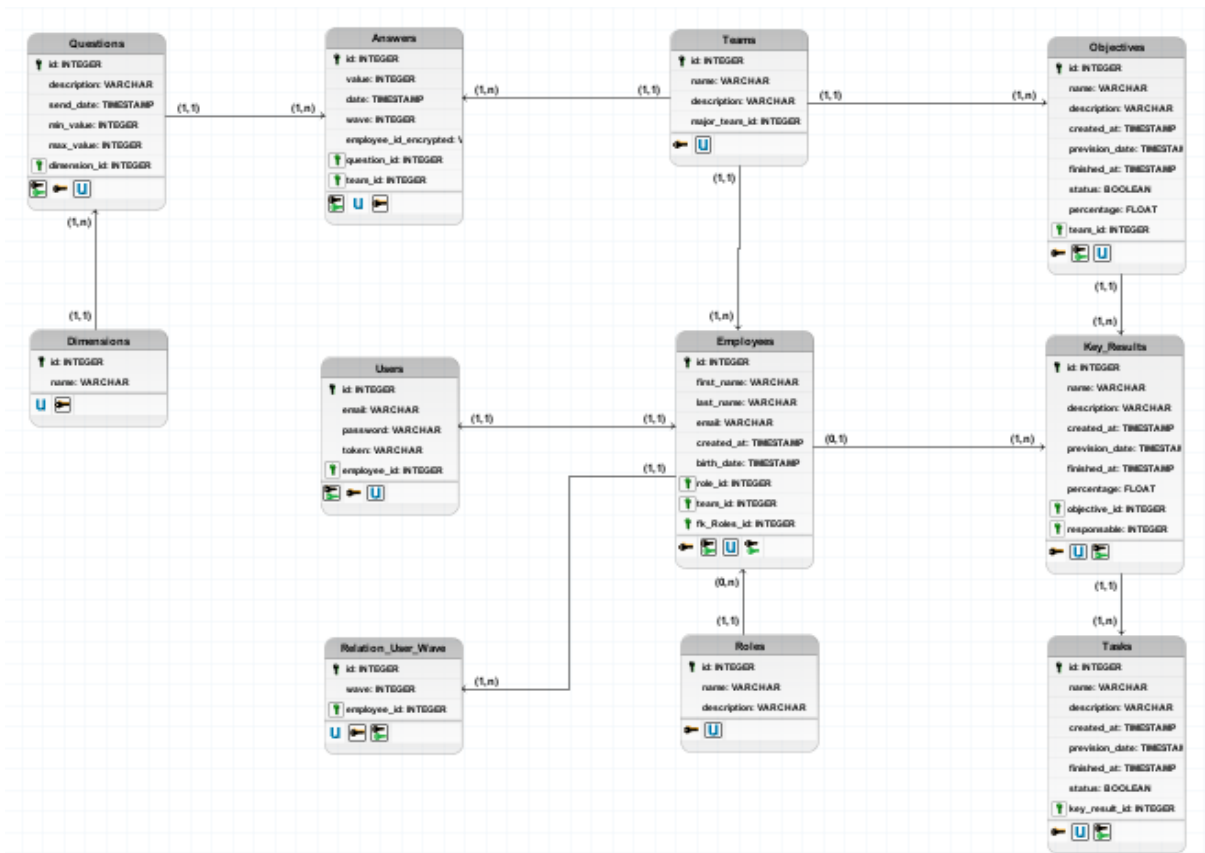


Fonte: elaborado pelo autor (2023)

4.3.2. Modelo Lógico

O modelo lógico do banco de dados traz consigo uma visão mais ampla de como será o banco de dados do sistema, exibindo suas chaves primárias, estrangeiras, atributos únicos e tipagem de atributos, mantendo demais características do modelo conceitual. Segue abaixo o modelo lógico do banco de dados do sistema.

Figura 10 - Modelo lógico do banco de dados.



Fonte: elaborado pelo autor (2023)

4.3.3. Modelo físico

O modelo físico do banco de dados se trata do código de criação em si do banco de dados, como descrito abaixo.

```
CREATE TABLE roles (
  id SERIAL PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  description VARCHAR(150) NOT NULL
);
```

```
CREATE TABLE objectives (
  id SERIAL PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  description VARCHAR(150) NOT NULL,
```



```
    created_at TIMESTAMP NOT NULL,  
    prevision_date TIMESTAMP NOT NULL,  
    finished_at TIMESTAMP,  
    status BOOLEAN NOT NULL,  
    percentage INTEGER DEFAULT 0,  
    team_id INTEGER REFERENCES teams(id)  
);
```

```
CREATE TABLE key_results (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    description VARCHAR(150) NOT NULL,  
    created_at TIMESTAMP NOT NULL,  
    prevision_date TIMESTAMP NOT NULL,  
    finished_at TIMESTAMP,  
    percentage INTEGER,  
    objective_id INTEGER REFERENCES objectives(id),  
    responsible INTEGER REFERENCES employees(id)  
);
```

```
CREATE TABLE tasks (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    description VARCHAR(150) NOT NULL,  
    created_at TIMESTAMP NOT NULL,  
    prevision_date TIMESTAMP NOT NULL,  
    finished_at TIMESTAMP,  
    status BOOLEAN NOT NULL,  
    key_result_id INTEGER REFERENCES key_results(id)  
);
```

```
CREATE TABLE teams (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,
```

```
description VARCHAR(150) NOT NULL,  
major_team_id INTEGER REFERENCES teams(id)  
);
```

```
CREATE TABLE employees (  
id SERIAL PRIMARY KEY,  
first_name VARCHAR(50) NOT NULL,  
last_name VARCHAR(50) NOT NULL,  
email VARCHAR(120) UNIQUE NOT NULL,  
created_at TIMESTAMP,  
birth_date TIMESTAMP NOT NULL,  
role_id INTEGER REFERENCES roles(id),  
team_id INTEGER REFERENCES teams(id)  
);
```

```
CREATE TABLE users (  
id SERIAL PRIMARY KEY,  
email VARCHAR(120) REFERENCES employees(email) UNIQUE,  
password VARCHAR(256) NOT NULL,  
token VARCHAR(256) NOT NULL,  
employee_id INTEGER REFERENCES employees(id)  
);
```

```
CREATE TABLE dimensions (  
id SERIAL PRIMARY KEY,  
name VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE questions (  
id SERIAL PRIMARY KEY,  
description VARCHAR(200) NOT NULL,  
send_date TIMESTAMP,  
min_value VARCHAR(50) NOT NULL,  
max_value VARCHAR(50) NOT NULL,
```

```
dimension_id INTEGER REFERENCES dimensions(id)
);
```

```
CREATE TABLE answers (
  id SERIAL PRIMARY KEY,
  value INTEGER NOT NULL,
  date TIMESTAMP,
  wave INTEGER NOT NULL,
  employee_id_encrypted VARCHAR(256) NOT NULL,
  question_id INTEGER REFERENCES questions(id),
  team_id INTEGER REFERENCES teams(id)
);
```

```
CREATE TABLE relation_user_wave (
  id SERIAL PRIMARY KEY,
  employee_id INTEGER REFERENCES employees(id),
  wave INTEGER NOT NULL
);
```

4.4. Projeto Técnico

Além de realizar o planejamento de uma aplicação de forma eficiente, é necessário garantir que os recursos tecnológicos aplicados para a construção de uma solução tecnológica sejam devidamente construídos e organizados. Desta forma nesta seção serão apresentados os aspectos técnicos , como a codificação e a organização de desenvolvimento, que foram aplicados na construção do software.

4.4.1. Implementação

4.4.1.1. Detalhes Técnicos

Durante o desenvolvimento do sistema de gestão de desempenho empresarial Vydra, foi utilizado o framework Flask para o back-end, sendo hospedado na

plataforma online Render, para o desenvolvimento do front-end da aplicação utilizou-se a biblioteca React, está hospedada na plataforma Netlify. Para o banco de dados da aplicação foi utilizado o PostgreSQL em sua 15ª versão, que por sua vez, foi hospedado na plataforma Render.

4.4.1.2. Desenvolvimento do código

O sistema foi desenvolvido utilizando um sistema de roteirização para consumo do front-end através dos blueprints presentes no Flask, que funcionam como uma maneira de compatibilizar partes da aplicação. Por meio desses blueprints a organização do back-end foi desenvolvida de forma mais coesa e padronizada, separando cada função específica como um componente que pode ser reutilizado de acordo com a necessidade existente. Para o consumo e persistência de dados no banco de dados as rotas blueprint processam os dados e os convertem num arquivo JSON que será consumido pelo front-end da aplicação e vice-versa.

No Flask foi utilizada a criação de classes como models para estabelecimento do esquema de banco de dados, definindo cada classe criada como uma tabela do banco de dados. Bibliotecas Flask como Flask Migrate para efetuar alterações nas classes do banco de dados através de suas migrações, SQLAlchemy para funções do banco de dados como as pesquisas, Pandas e Psycopg também foram utilizadas para demais usos do banco de dados.

No front-end da aplicação, utilizamos o Framework React também utilizando a componentização como peça fundamental da aplicação, com bibliotecas como Material UI que dispõe de um grande acervo de componentes reutilizáveis e de fácil acesso. Para a roteirização utilizamos a biblioteca React Router Dom, com seu eficaz uso para transição entre as telas da aplicação. A estilização das telas foi feita através dos arquivos scss e edição dos componentes do Material UI através das propriedades do componente.

4.4.1.3. Organização de arquivos da aplicação

4.4.1.3.1. Organização de arquivos do front-end da aplicação

A organização de arquivos do front-end foi criada de maneira simples e intuitiva facilitando o entendimento para quem tenha contato com o código, além de facilitar na atualização e manutenção do código. A pasta src abriga todas as partes do front-end, sendo os componentes, páginas, ativos da aplicação, App.jsx e Main.jsx. Dentro da área de componentes estão os componentes criados com reuso contínuo como o navbar do sistema. Dentro de cada componente estão o arquivo jsx possuindo a lógica do componente e o arquivo scss com a estilização do componente. Dentro de pages ficam estabelecidas as páginas da aplicação possuindo três arquivos, o arquivo jsx com a lógica da página, o arquivo test.jsx com os testes da página e o arquivo scss com a estilização. Em assets estão os ativos do sistema como imagens e definições específicas de estilos. O arquivo App.jsx é o responsável pela importação de arquivos e estabelecimento de rotas do arquivo. Por fim, o arquivo Main.jsx estabelece configurações para a aplicação Web.

4.4.1.3.2. Organização de arquivos do front-end da aplicação

A organização de arquivos do back-end da aplicação também foi feita de forma simples e intuitiva, possuindo apenas a pasta migrations, vydra, app.py e arquivos comuns para a aplicação. A pasta Migrations é criada de forma automática pela biblioteca Flask Migrate possuindo cada alteração feita nas models da aplicação. Dentro de vydra estão as rotas da aplicação com seu arquivo de lógica e o arquivo models.py possuindo as especificações do esquema de banco de dados. O arquivo App.py é o responsável pela configuração do banco de dados, importação de arquivos e bibliotecas importantes como Flask CORS, registro das rotas e blueprint. O arquivo Profile é usado para realização do deploy na plataforma Render e o arquivo requirements.txt possui todas as importações de bibliotecas do back-end com suas respectivas versões.

4.4.1.4. Códigos da Aplicação

- Arquivo authentication.py

Figura 11 - Código do arquivo authentication.py

```
import random
import string
import jwt
from datetime import datetime, timedelta
from flask import Blueprint, request, jsonify
from functools import wraps
from vydra.models import Employees, Users
from werkzeug.security import check_password_hash

authentication_routes_bp = Blueprint('authentication_routes', __name__)

def gerador_token(employee_id):
    string_random = string.ascii_letters + string.digits + string.ascii_uppercase
    key = ''.join(random.choice(string_random) for i in range(12))
    employee = Employees.query.filter(employee_id == Employees.id).one()
    employee_information = {
        "id": employee.id,
        "first_name": employee.first_name,
        "last_name": employee.last_name,
        "email": employee.email,
        "role_id": employee.role_id,
        "team_id": employee.team_id
    }
    token = jwt.encode({'username': employee_information, 'exp': datetime.now() + timedelta(hours=12)},
                      key)
    return token

def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.headers.get('token')
        if not token:
            return jsonify({'message': 'token is missing', 'data': []}), 401
        try:
            usuario = Users.query.filter(Users.token == token).one()
        except:
            return jsonify({'message': 'token is invalid or expired', 'data': []}), 401
```

Fonte: elaborado pelo autor (2023)

Figura 12 - Continuação do código do arquivo authentication.py

```
        return f(*args, **kwargs)
    return decorated

@authentication_routes_bp.route('/login', methods=['POST', 'GET'])
def login():
    payload = request.json

    print(payload)
    login=payload['login'],
    senha=payload['password']
    usuario = Users.query.filter(Users.email == login).one()
    print(usuario)
    employee = Employees.query.filter(Employees.email == login).one()
    print(employee)

    try:
        verificacao = check_password_hash(pwhash=usuario.password, password=senha)
        print(verificacao)
    except:
        return jsonify({'message': 'Não foi possível executar o login! Contate o suporte.'}), 400
    if not verificacao:
        return jsonify({'message': 'Login inválido!'}), 400

    return jsonify({'message': 'Login válido!', 'token': usuario.token, 'employee_id': employee.id,
                    'first_name': employee.first_name, 'last_name': employee.last_name, 'email': employee.email,
                    'created_at': employee.created_at, 'birth_date': employee.birth_date, 'role_id': employee.role_id,
                    'team_id': employee.team_id, 'user_id': usuario.id, })

@authentication_routes_bp.route('/validate-token', methods=['POST'])
def validate_token():
    payload = request.json

    token=payload['token']

    try:
        logged_user = Users.query.filter(Users.token == token).one()
    except:
        return jsonify({'message': 'Token inválido.'}), 400

return jsonify({'message': 'Token válido!'})
```

- Arquivo app.py

Figura 13 - Código do arquivo app.py

```

from flask import Flask
from flask_cors import CORS
from flask_migrate import Migrate
from flask_sqlalchemy import SQLAlchemy
from apscheduler.schedulers.background import BackgroundScheduler
import os

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = "postgresql://vydra_96oh_user:LNZSNaXgaB2tnD51TY8eHxNgeJ5PK8zg@dpg-cjju8uuphtvs73eff01g-a/vydra_96oh"
db = SQLAlchemy(app)
migrate = Migrate(app, db)
CORS(app)

if __name__ == "__main__":
    app.run(debug=True)

from vydra.rotas.common_routes.common_routes import rotas_bp
from vydra.rotas.funcionarios.funcionario_routes import employees_routes_bp
from vydra.rotas.tasks.tasks_routes import task_routes_bp
from vydra.rotas.roles.roles_routes import role_routes_bp
from vydra.rotas.teams.teams_routes import team_routes_bp
from vydra.rotas.users.users_routes import user_routes_bp
from vydra.rotas.objectives.objectives_routes import objectives_routes_bp
from vydra.rotas.key_results.key_results_routes import key_results_routes_bp
from vydra.rotas.authentication.authentication import authentication_routes_bp
from vydra.rotas.dashboard.dashboard_routes import dashboard_routes_bp
from vydra.rotas.profile.profile_routes import profile_routes_bp
from vydra.rotas.climate.climate_routes import climate_routes_bp, atualiza_perguntas

#Instancia a classe de tarefas agendadas
scheduler = BackgroundScheduler()
# scheduler.add_job(atualiza_perguntas, 'cron', day_of_week='wed', hour=1, minute=10)
scheduler.add_job(atualiza_perguntas, 'cron', day_of_week='fri', hour=1, minute=10)
scheduler.start()

```

Fonte: elaborado pelo autor (2023)

Figura 14 - Continuação do código do arquivo app.py

```

app.register_blueprint(rotas_bp)
app.register_blueprint(employees_routes_bp)
app.register_blueprint(task_routes_bp)
app.register_blueprint(role_routes_bp)
app.register_blueprint(team_routes_bp)
app.register_blueprint(user_routes_bp)
app.register_blueprint(objectives_routes_bp)
app.register_blueprint(key_results_routes_bp)
app.register_blueprint(authentication_routes_bp)
app.register_blueprint(dashboard_routes_bp)
app.register_blueprint(profile_routes_bp)
app.register_blueprint(climate_routes_bp)

```

Fonte: elaborado pelo autor (2023)

- Arquivo models.py

Figura 15 - Código do arquivo models.py

```
import psycopg2 as pg
import pandas as pd
from app import db, app

class Roles(db.Model):
    __tablename__ = 'roles'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    name = db.Column(db.String(50), nullable=False)
    description = db.Column(db.String(150), nullable=False)

class Objectives(db.Model):
    __tablename__ = 'objectives'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    name = db.Column(db.String(50), nullable=False)
    description = db.Column(db.String(150), nullable=False)
    created_at = db.Column(db.DateTime, nullable=False)
    prevision_date = db.Column(db.DateTime, nullable=False)
    finished_at = db.Column(db.DateTime, nullable=True)
    status = db.Column(db.Boolean, nullable=False)
    percentage = db.Column(db.Integer, nullable=True, default=0)
    team_id = db.Column(db.Integer, db.ForeignKey('teams.id'))

class KeyResults(db.Model):
    __tablename__ = 'key_results'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    name = db.Column(db.String(50), nullable=False)
    description = db.Column(db.String(150), nullable=False)
    created_at = db.Column(db.DateTime, nullable=False)
    prevision_date = db.Column(db.DateTime, nullable=False)
    finished_at = db.Column(db.DateTime, nullable=True)
    percentage = db.Column(db.Integer, nullable=True, default=0)
    objective_id = db.Column(db.Integer, db.ForeignKey('objectives.id'))
    responsable = db.Column(db.Integer, db.ForeignKey('employees.id'))
```

Fonte: elaborado pelo autor (2023)

Figura 16 - Primeira continuação do código models.py

```
class Tasks(db.Model):
    __tablename__ = 'tasks'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    name = db.Column(db.String(50), nullable=False)
    description = db.Column(db.String(150), nullable=False)
    created_at = db.Column(db.DateTime, nullable=False)
    prevision_date = db.Column(db.DateTime, nullable=False)
    finished_at = db.Column(db.DateTime, nullable=True)
    status = db.Column(db.Boolean, nullable=False)
    key_result_id = db.Column(db.Integer, db.ForeignKey('key_results.id'))

class Teams(db.Model):
    __tablename__ = 'teams'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    name = db.Column(db.String(50), nullable=False)
    description = db.Column(db.String(150), nullable=False)
    major_team_id = db.Column(db.Integer, db.ForeignKey('teams.id'), nullable=True)

class Employees(db.Model):
    __tablename__ = 'employees'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    first_name = db.Column(db.String(50), nullable=False)
    last_name = db.Column(db.String(50), nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    created_at = db.Column(db.DateTime, nullable=True)
    birth_date = db.Column(db.DateTime, nullable=False)
    role_id = db.Column(db.Integer, db.ForeignKey('roles.id'))
    team_id = db.Column(db.Integer, db.ForeignKey('teams.id'))

class Users(db.Model):
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    email = db.Column(db.String(120), db.ForeignKey('employees.email'), nullable=True, unique=True)
    password = db.Column(db.String(256), nullable=False)
    token = db.Column(db.String(256), nullable=False)
    employee_id = db.Column(db.Integer, db.ForeignKey('employees.id'))
```

Fonte: elaborado pelo autor (2023)

Figura 17 - Segunda continuação do código do arquivo models.py

```

class Dimensions(db.Model):
    __tablename__ = 'dimensions'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    name = db.Column(db.String(50), nullable=False)

class Questions(db.Model):
    __tablename__ = 'questions'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    description = db.Column(db.String(200), nullable=False)
    send_date = db.Column(db.DateTime, nullable=True)
    min_value = db.Column(db.String(50), nullable=False)
    max_value = db.Column(db.String(50), nullable=False)
    dimension_id = db.Column(db.Integer, db.ForeignKey('dimensions.id'))

class Answers(db.Model):
    __tablename__ = 'answers'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    value = db.Column(db.Integer, nullable=False)
    date = db.Column(db.DateTime, nullable=True)
    wave = db.Column(db.Integer, nullable=False)
    employee_id_encrypted = db.Column(db.String(256), nullable=False)
    question_id = db.Column(db.Integer, db.ForeignKey('questions.id'))
    team_id = db.Column(db.Integer, db.ForeignKey('teams.id'))

class Relation_User_Wave(db.Model):
    __tablename__ = 'relation_user_wave'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True, nullable=False)
    employee_id = db.Column(db.Integer, db.ForeignKey('employees.id'))
    wave = db.Column(db.Integer, nullable=False)

with app.app_context():
    db.create_all()

class Postsql:
    def __init__(self, host, database, user, password):
        self.db = pg.connect(host=host, database=database, user=user, password=password)

    def query(self, query):
        cur = self.db.cursor()

```

Fonte: elaborado pelo autor (2023)

Figura 18 - Terceira continuação do código do arquivo models.py

```

        cur.execute(query)
        retorno = cur.fetchall()
        return retorno

    def database(self, query):
        resp = pd.read_sql_query(query, self.db)
        resp = resp.values
        return resp

    def closedb(self):
        self.db.close()

```

Fonte: elaborado pelo autor (2023)

- Arquivo app.jsx

Figura 19 - Código do arquivo app.jsx

```
import React from "react";
import { Route, BrowserRouter as Router, Routes } from "react-router-dom";
import Employees from "../pages/employees/employees.jsx";
import Login from "../pages/login/login.jsx";
import Okr from "../pages/okr/okr.jsx";
import Organograma from "../pages/organograma/organograma.jsx";
import Analytics from "../pages/analytics/analytics.jsx";
import PaineL from "../pages/paineL/paineL.jsx";
import Climate from "../pages/climate/climate.jsx";
import Profile from "../pages/profile/profile.jsx";
import ChangePassword from "../pages/profile/change_password.jsx";
import Teams from "../pages/teams/teams.jsx";

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Login />} />
        <Route path="/paineL" element={<PaineL />} />
        <Route path="/profile" element={<Profile />} />
        <Route path="/change_password" element={<ChangePassword />} />
        <Route path="/profile/:employee_id/climate" element={<Climate />} />
        <Route path="/teams/:team_id/okr" element={<Okr />} />
        <Route path="/teams" element={<Teams />} />
        <Route path="/employees" element={<Employees />} />
        <Route path="/organograma" element={<Organograma />} />
        <Route path="/teams/:team_id/analytics" element={<Analytics />} />
      </Routes>
    </Router>
  );
}

export default App;
```

Fonte: elaborado pelo autor (2023)

- Arquivo employees.jsx

Figura 20 - Código do arquivo employees.jsx

```
const columns = [
  { id: "name", label: "Funcionário", minWidth: 170 },
  { id: "role", label: "Cargo", minWidth: 100 },
  { id: "team", label: "Time", minWidth: 170 },
];

function Employees() {
  const navigate = useNavigate()
  const token = localStorage.getItem("token")
  const BASE_URL = "https://vydra-back.onrender.com";
  const endpoint = `${BASE_URL}/page_employees`;
  const [searchValue, setSearchValue] = useState("");
  const [employees, setEmployees] = useState([]);
  const [filteredEmployees, setFilteredEmployees] = useState([]);
  const [roles, setRoles] = useState([]);
  const [teams, setTeams] = useState([]);
  const [page, setPage] = useState(0);
  const [rowsPerPage, setRowsPerPage] = useState(10);
  const [openDialog, setOpenDialog] = useState(false);
  const [menuAnchorEl, setMenuAnchorEl] = useState(null);
  const [selectedEmployeeId, setSelectedEmployeeId] = useState(null);
  const [deleteDialogOpen, setDeleteDialogOpen] = useState(false);
  const [editDialogOpen, setEditDialogOpen] = useState(false);
  const [selectedEmployee, setSelectedEmployee] = useState({})
  const [newEmployee, setNewEmployee] = useState({
    first_name: "",
    last_name: "",
    email: "",
    birth_date: "",
    role_id: null,
    team_id: null
  });

  const filterOptions = createFilterOptions({
    stringify: (option) => option.name,
  });
}
```

Fonte: elaborado pelo autor (2023)

Figura 21 - Primeira continuação do código do arquivo employees.jsx

```
const fetchEmployees = async () => {
  try {
    const response = await fetch(endpoint, {
      headers: {
        token: `${token}`
      }
    });
    const data = await response.json();

    const flattenedData = data.flatMap((nestedData) => nestedData);

    const mappedData = flattenedData.map((employee) => ({
      ...employee,
      id: employee.employee_id,
    }));

    setEmployees(mappedData);
    setFilteredEmployees(mappedData);

    const rolesResponse = await fetch(`${BASE_URL}/roles`, {
      headers: {
        token: `${token}`
      }
    });
    const rolesData = await rolesResponse.json();
    setRoles(rolesData);

    const teamsResponse = await fetch(`${BASE_URL}/teams`, {
      headers: {
        token: `${token}`
      }
    });
    const teamsData = await teamsResponse.json();
    setTeams(teamsData);

  } catch (error) {
    console.error(error);
  }
};
```

Fonte: elaborado pelo autor (2023)

Figura 22 - Segunda continuação do código do arquivo employees.jsx

```
const validateToken = async (token) => {
  let endpoint = `${BASE_URL}/validate-token`
  let body = {
    'token': token
  }

  let options = {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(body),
  };

  try {
    let response = await fetch(endpoint, options)
    if (response.status !== 200) return false
    return true
  }
  catch (error) {
    return false
  }
}

async function handleStorageEvent(event) {
  if (event.key === 'token') {
    const validation = await validateToken(event.newValue)
    if (!validation) {
      navigate('/')
      return
    }
  }
}
}
```

Fonte: elaborado pelo autor (2023)

Figura 23 - Terceira continuação do código do arquivo employees.jsx

```
useEffect(() => {
  const token = localStorage.getItem("token")
  if (!token){
    navigate('/')
    return
  }
  window.addEventListener('storage', handleStorageEvent);
  fetchEmployees();
}, []);

const handleChangePage = (event, newPage) => {
  setPage(newPage);
};

const handleChangeRowsPerPage = (event) => {
  setRowsPerPage(+event.target.value);
  setPage(0);
};

const handleSearch = (event, newValue) => {
  setSearchValue(newValue);
  if (newValue === "" || !newValue) {
    setFilteredEmployees(employees);
  } else {
    const filtered = employees.filter((employee) =>
      employee.name.toLowerCase().includes(newValue.toLowerCase())
    );
    setFilteredEmployees(filtered);
  }
  setPage(0);
};

const handleDialogOpen = () => {
  setOpenDialog(true);
};

const handleDialogClose = () => {
  setOpenDialog(false);
};
```

Fonte: elaborado pelo autor (2023)

Figura 24 - Quarta continuação do código do arquivo employees.jsx

```
const handleInputChange = (event) => {
  const { name, value } = event.target;
  setNewEmployee((prevEmployee) => ({
    ...prevEmployee,
    [name]: value
  }));
};

const handleEditInputChange = (event) => {
  const { name, value } = event.target;
  setSelectedEmployee((prevEmployee) => ({
    ...prevEmployee,
    [name]: value
  }));
};

const handleNewEmployee = async () => {
  try {
    const response = await fetch(`${BASE_URL}/employees`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        "token": `${token}`
      },
      body: JSON.stringify(newEmployee)
    });
    if (response.ok) {
      setNewEmployee({
        first_name: "",
        last_name: "",
        email: "",
        birth_date: "",
        role_id: null,
        team_id: null
      });
      handleDialogClose();
      fetchEmployees(); // Refresh the employee list after creating a new employee
    }
  }
};
```

Fonte: elaborado pelo autor (2023)

Figura 25 - Quinta continuação do código do arquivo employees.jsx

```
    } else {
      console.error("Failed to create a new employee.");
    }
  } catch (error) {
    console.error(error);
  }
};

const handleMenuOpen = (event, employee) => {
  setSelectedEmployeeId(employee.id);
  setSelectedEmployee(employee);
  setMenuAnchorEl(event?.currentTarget);
};

const handleMenuClose = () => {
  setSelectedEmployeeId(null);
  setMenuAnchorEl(null);
};

const handleEditEmployee = async () => {
  if (!selectedEmployeeId) {
    console.error("Nenhum funcionário selecionado para edição.");
    return;
  }

  try {
    let body = {
      'first_name': selectedEmployee.first_name,
      'last_name': selectedEmployee.last_name,
      'role_id': selectedEmployee.role_id,
      'team_id': selectedEmployee.team_id
    };

    const response = await fetch(`${BASE_URL}/employees/${selectedEmployeeId}`, {
      method: "PUT",
      headers: {
        "Content-Type": "application/json",
        "token": `${token}`,
      },
      body: JSON.stringify(body)
    });
```

Fonte: elaborado pelo autor (2023)

Figura 26 - Sexta continuação do código do arquivo employees.jsx

```
    });
    body: JSON.stringify(body),
  });

  if (response.ok) {
    fetchEmployees();
  } else {
    console.error("Falha ao excluir o funcionário.");
  }
} catch (error) {
  console.error(error);
}

handleMenuClose();
}

const handleDeleteEmployee = async () => {
  if (!selectedEmployeeId) {
    console.error("Nenhum funcionário selecionado para edição.");
    return;
  }
  try {
    const response = await fetch(`${BASE_URL}/employees/${selectedEmployeeId}`, {
      method: "DELETE",
      headers: {
        token: `${token}`,
      },
    });
  }

  if (response.ok) {
    fetchEmployees();
  } else {
    console.error("Falha ao excluir o funcionário.");
  }
} catch (error) {
  console.error(error);
}

handleMenuClose();
};
```

Fonte: elaborado pelo autor (2023)

Figura 27 - Sétima continuação do código do arquivo employees.jsx

```
const handleEditDialogOpen = () => {
  setEditDialogOpen(true);
};

const handleEditDialogClose = () => {
  setEditDialogOpen(false);
};

const handleDeleteDialogOpen = () => {
  setDeleteDialogOpen(true);
};

const handleDeleteDialogClose = () => {
  setDeleteDialogOpen(false);
};

return (
  <div className="teams">
    <Navbar />
    <div className="h2-teams">Gestão de funcionários</div>
    <div className="search-container">
      <Autocomplete
        value={searchValue}
        onChange={handleSearch}
        options={employees.map((employee) => employee.name)}
        renderInput={({ params } => (
          <TextField {... params} label="Buscar funcionários" />
        ))}
      />
      <Button
        variant="contained"
        color="primary"
        className="new-employee-button"
        onClick={handleDialogOpen}
      />
      Novo funcionário
    </div>
  </div>
  <div className="table-container">
    <Paper sx={{ width: "100%" }}>
```

Fonte: elaborado pelo autor (2023)

Figura 28 - Oitava continuação do código do arquivo employees.jsx

```

<TableContainer sx={{ maxHeight: 440 }}>
  <Table stickyHeader aria-label="sticky table">
    <TableHead>
      <TableRow>
        {columns.map((column) => (
          <TableCell
            key={column.id}
            align="center"
            style={{ minWidth: column.minWidth }}
          >
            {column.label}
          </TableCell>
        ))}
      </TableRow>
    </TableHead>
    <TableBody>
      {filteredEmployees
        .slice(page * rowsPerPage, page * rowsPerPage + rowsPerPage)
        .map((employee) => (
          <TableRow key={employee.name}>
            <TableCell align="center">{employee.name}</TableCell>
            <TableCell align="center">{employee.role_name}</TableCell>
            <TableCell align="center">{employee.team_name}</TableCell>
            <TableCell align="center">
              <HiOutlinePencilAlt onClick={(event) => handleMenuOpen(event, employee)} />
              <Menu
                anchorEl={menuAnchorEl}
                open={Boolean(menuAnchorEl && selectedEmployeeId === employee.id)}
                onClose={handleMenuClose}
              >
                <MenuItem onClick={handleEditDialogOpen}>
                  Atualizar Funcionários
                </MenuItem>
                <Dialog open={editDialogOpen} onClose={handleEditDialogClose}>
                  <DialogTitle>Editar funcionário</DialogTitle>
                  <DialogContent>
                    <TextField
                      autoFocus
                      margin="dense"
                      name="first_name"

```

Fonte: elaborado pelo autor (2023)

Figura 29 - Nona continuação do código do arquivo employees.jsx

```
        label="Nome"
        type="text"
        fullWidth
        value={selectedEmployee.first_name}
        onChange={handleEditInputChange}
      />
      <TextField
        margin="dense"
        name="last_name"
        label="Sobrenome"
        type="text"
        fullWidth
        value={selectedEmployee.last_name}
        onChange={handleEditInputChange}
      />
      <Autocomplete
        onChange={(event, newValue) => setSelectedEmployee({ ...selectedEmployee, role_id: newValue.id})}
        options={roles}
        getOptionLabel={(option) => option.name }
        renderInput={(params) => (
          <TextField {... params} label="Cargo" />
        )}
      />
      <Autocomplete
        onChange={(event, newValue) => setSelectedEmployee({ ...selectedEmployee, team_id: newValue.id})}
        options={teams}
        getOptionLabel={(option) => option.name }
        renderInput={(params) => (
          <TextField {... params} label="Time" />
        )}
      />
    </DialogContent>
    <DialogActions>
      <Button onClick={handleEditDialogClose}>Cancelar</Button>
      <Button onClick={handleEditEmployee}>Salvar</Button>
    </DialogActions>
  </Dialog>
  <MenuItem onClick={handleDeleteDialogOpen}>
    Deletar Funcionários
  </MenuItem>
  <Dialog open={deleteDialogOpen} onClose={handleDeleteDialogClose}>
```

Fonte: elaborado pelo autor (2023)

Figura 30 - Décima continuação do código do arquivo employees.jsx

```
        <Dialog open={deleteDialogOpen} onClose={handleDeleteDialogClose}>
          <DialogTitle>Confirmar exclusão</DialogTitle>
          <DialogContent>
            Tem certeza de que deseja excluir este funcionário?
          </DialogContent>
          <DialogActions>
            <Button onClick={handleDeleteDialogClose}>Cancelar</Button>
            <Button onClick={handleDeleteEmployee}>Deletar</Button>
          </DialogActions>
        </Dialog>
      </Menu>
    </TableCell>
  </TableRow>
)
</TableBody>
</Table>
</TableContainer>
<TablePagination
  rowsPerPageOptions={[10, 25, 100]}
  component="div"
  count={filteredEmployees.length}
  rowsPerPage={rowsPerPage}
  page={page}
  onPageChange={handleChangePage}
  onRowsPerPageChange={handleChangeRowsPerPage}
  />
</Paper>
</div>
<Dialog open={openDialog} onClose={handleDialogClose}>
  <DialogTitle>Novo funcionário</DialogTitle>
  <DialogContent>
    <TextField
      autoFocus
      margin="dense"
      name="first_name"
      label="Nome"
      type="text"
      fullWidth
      value={newEmployee.first_name}
      onChange={handleInputChange}
    />
  </DialogContent>
</Dialog>
```

Fonte: elaborado pelo autor (2023)

Figura 31 - Décima primeira continuação do código do arquivo employees.jsx

```

    onChange={handleInputChange}
  />
  <TextField
    margin="dense"
    name="last_name"
    label="Sobrenome"
    type="text"
    fullWidth
    value={newEmployee.last_name}
    onChange={handleInputChange}
  />
  <TextField
    margin="dense"
    name="email"
    label="Email"
    type="email"
    fullWidth
    value={newEmployee.email}
    onChange={handleInputChange}
  />
  <TextField
    margin="dense"
    name="birth_date"
    label="Data de nascimento"
    type="date"
    fullWidth
    value={newEmployee.birth_date}
    onChange={handleInputChange}
    InputLabelProps={{
      shrink: true
    }}
  />
  <Autocomplete
    onChange={(event, newValue) => setNewEmployee({...newEmployee, role_id: newValue.id})}
    options={roles}
    getOptionLabel={(option) => option.name }
    renderInput={(params) => (
  <TextField {... params} label="Cargo" />
    )}
  />
  <Autocomplete

```

Fonte: elaborado pelo autor (2023)

Figura 32 - Décima segunda continuação do código do arquivo employees.jsx

```

    onChange={(event, newValue) => setNewEmployee({...newEmployee, team_id: newValue.id})}
    options={teams}
    getOptionLabel={(option) => option.name }
    renderInput={(params) => (
  <TextField {... params} label="Time" />
    )}
  />
  </DialogContent>
  <DialogActions>
    <Button onClick={handleDialogClose}>Cancelar</Button>
    <Button onClick={handleNewEmployee}>Salvar</Button>
  </DialogActions>
</Dialog>
</div>
);
}

export default Employees;

```

Fonte: elaborado pelo autor (2023)

- Arquivo profile.styles.css.

Figura 33 - Código do arquivo profile.styles.css

```
body {  
  overflow-y: scroll;  
}  
  
.profile-container {  
  background-color: #fff;  
  border-radius: 15px;  
  box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.1);  
  display: flex;  
  flex-direction: column;  
  align-self: center;  
  justify-content: center;  
  width: 40%;  
  margin-top: 30px;  
  margin-left: 30%;  
  margin-right: 30%;  
  color: #1F1F1F;  
  padding-bottom: 30px;  
}  
  
.profile-details {  
  justify-content: space-between;  
}  
  
.form-container {  
  display: flex;  
  flex-direction: column;  
  align-self: center;  
  justify-content: center;  
  align-items: center;  
  align-self: center;  
  margin-top: 20px;  
  color: #1F1F1F  
}  
  
.form {  
  width: 375px;  
}
```

Fonte: elaborado pelo autor (2023)

Figura 34 - Primeira continuação do arquivo profile.styles.css

```
.label-form {  
  font-size: 14px;  
  color: #1F1F1F;  
  margin-bottom: 10px;  
}  
  
.input-form {  
  font-size: 1rem;  
  width: 100%;  
  height: 30px;  
  padding: 2px 5px;  
  color: #5a5959;  
  border-width: 1px;  
  border-radius: 10px;  
  margin-bottom: 10px;  
  border-color: grey;  
}  
  
.avatar {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  text-align: center;  
  width: 70px;  
  height: 70px;  
  background-color: #464BA8;  
  border-radius: 50%;  
  box-shadow: 0 4px 4px #ccc;  
  margin-bottom: 30px;  
}  
  
.initials {  
  font-size: 35px;  
  color: #fff;  
  font-weight: bold;  
}
```

Fonte: elaborado pelo autor (2023)

Figura 35 - Segunda continuação do arquivo profile.styles.css

```
.success-message {  
  margin-top: 2px;  
  font-size: 14px;  
  color: green;  
}  
  
.div-buttons-change {  
  display: flex;  
  flex-direction: row;  
  justify-content: end;  
  height: auto;  
}  
  
.margin-30 {  
  margin-top: 30px;  
}
```

Fonte: elaborado pelo autor (2023)

4.4.2. Segurança

Em um cenário onde o uso da tecnologia está cada vez mais alto, a segurança em um sistema é vital para garantir a integridade digital dos usuários e garantir que informações críticas mantenham sua confidencialidade e sua disponibilidade sem se comprometer.

Visando isso, buscamos diversas formas para tornar o Vydra um software seguro, onde os usuários possam ter confiança e usá-lo sem preocupações.

4.4.2.1. Sistema de Login e Senha

Através do login e senha do usuário, é garantido que só usuários com devido acesso possam interagir e utilizar o Vydra, tornando-o exclusivo para os usuários cadastrados e não possibilitando o uso sem ter sido cadastrado previamente.

4.4.2.2. Sistema de autenticação da API por Token JWT

O token JWT é utilizado para criptografar e proteger os dados que são inseridos e armazenados no sistema. Ele é gerado após o login do usuário para que este recupere, veja, altere ou exclua dados de forma segura e autenticada.

4.4.2.3. Ambiente seguro de host

Para hospedar o sistema, foram escolhidas plataformas que ofereciam a maior segurança possível para os usuários.

O Netlify (utilizado para hospedagem do React) oferece suporte para TLS, garantindo que a comunicação entre cliente e servidor seja assegurada por criptografia. É fornecido também certificados SSL para o site, tornando possível a utilização de HTTPS sem a necessidade de uma configuração complexa para tal função. Além disso, a infraestrutura do Netlify é totalmente projetada para proteção de seus sites hospedados contra ataques DDoS, garantindo uma melhor e mais segura disponibilidade do software.

4.4.2.4. Protocolos HTTPS

O HTTPS é um protocolo que garante segurança na comunicação entre o navegador web e o servidor que está nele. Através de métodos de criptografia, ele garante a integridade dos dados transferidos e estabelece uma conexão confiável e autêntica para o usuário.

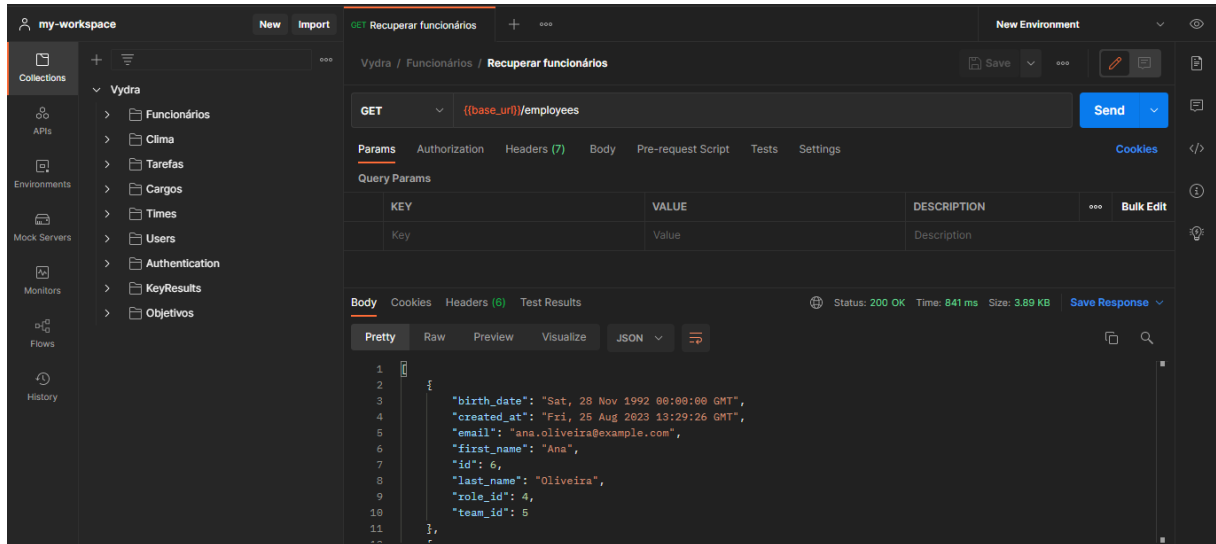
4.4.2.5. Níveis de acesso do usuário

Através dos níveis de acesso (gestor e colaborador) disponíveis no Vydra, o usuário poderá visualizar e interagir apenas com o que realmente tiver permissão. Isso garante que um funcionário comum não veja dados sensíveis da empresa e não execute uma ação que apenas um gestor tenha permissão, como excluir um funcionário ou alterar um resultado-chave da organização.

4.4.3. Testes

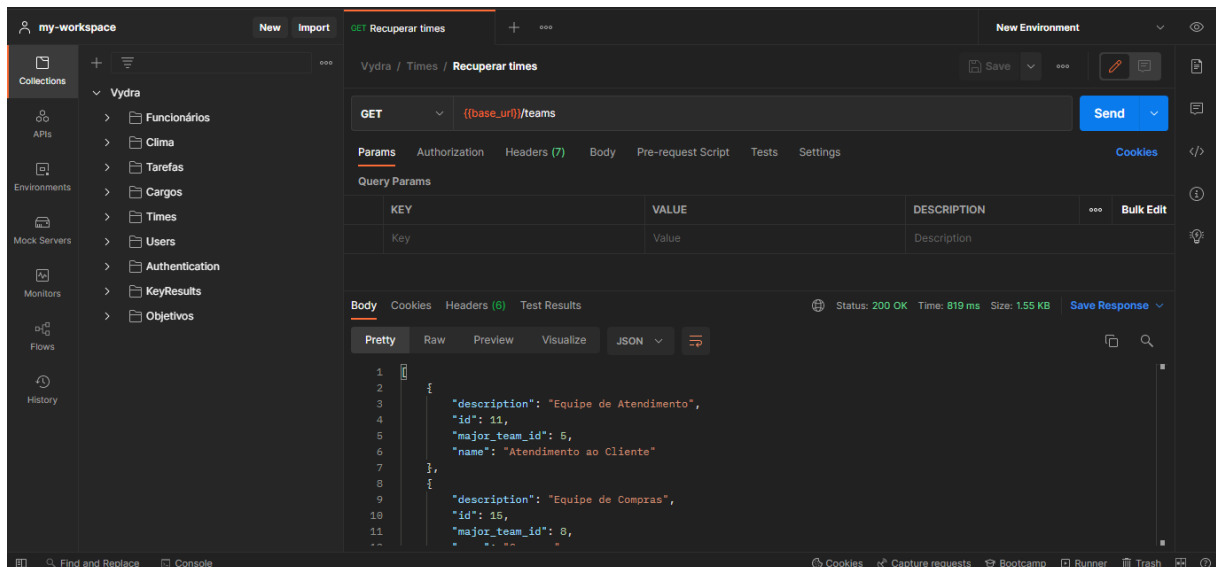
Os testes se baseiam na verificação de todos os endpoints desenvolvidos da API, e se as funcionalidades do front estavam se conectando corretamente com o back-end. Abaixo seguem imagens do processo de testes realizados na API:

Figura 36 - Teste da requisição de recuperar funcionários



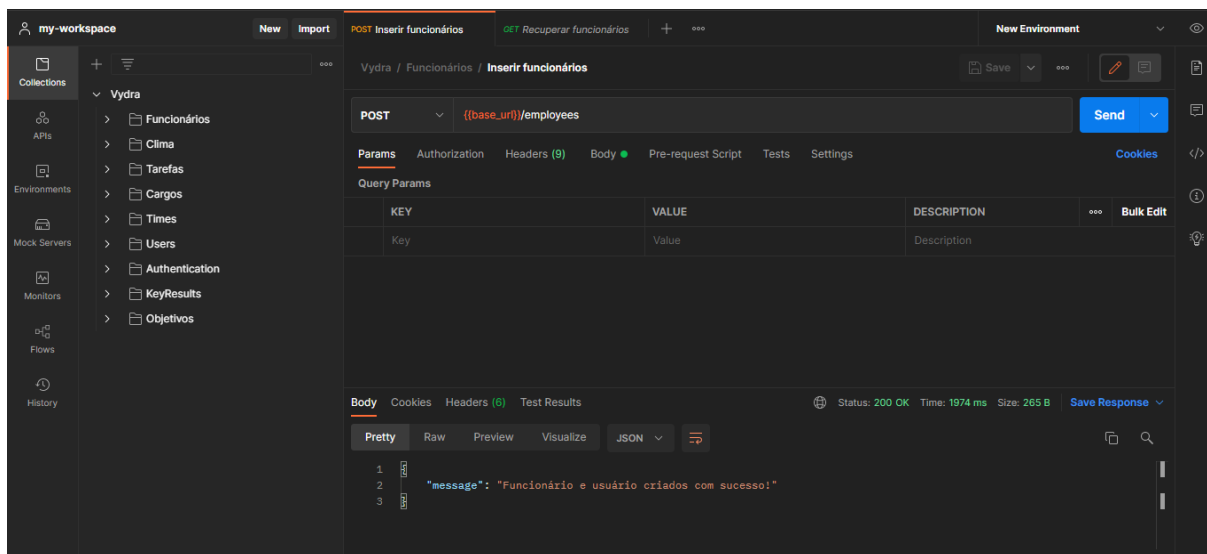
Fonte: elaborado pelo autor (2023)

Figura 37 - Teste da requisição de recuperar times



Fonte: elaborado pelo autor (2023)

Figura 38 - Teste da requisição de inserir funcionários

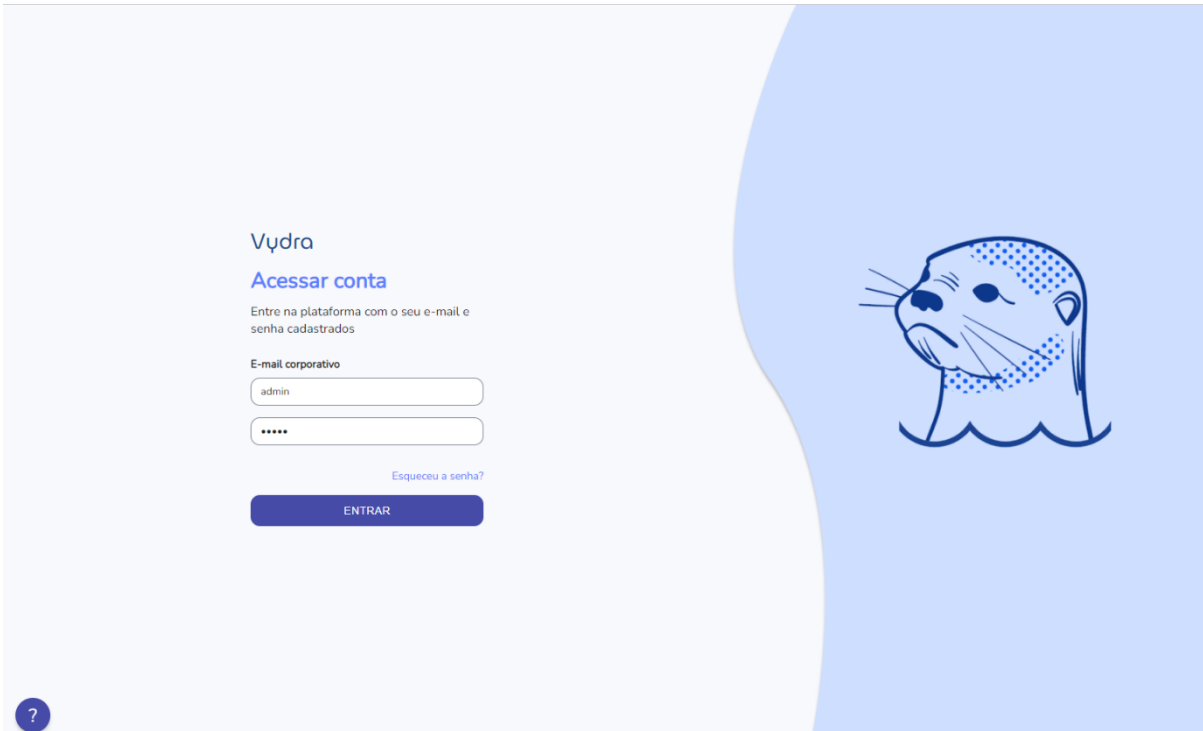


Fonte: elaborado pelo autor (2023)

4.4.4. Descrição das interfaces

- Tela inicial de login do sistema: tela na qual possibilita que o usuário cadastrado no sistema possa efetuar o login utilizando seu e-mail corporativo e sua senha. O usuário pode clicar no botão de ajuda caso esteja enfrentando algum problema ao efetuar o login.

Figura 39 - Tela de login da aplicação



Vydra

Acessar conta

Entre na plataforma com o seu e-mail e senha cadastrados

E-mail corporativo

admin

[Esqueceu a senha?](#)

ENTRAR

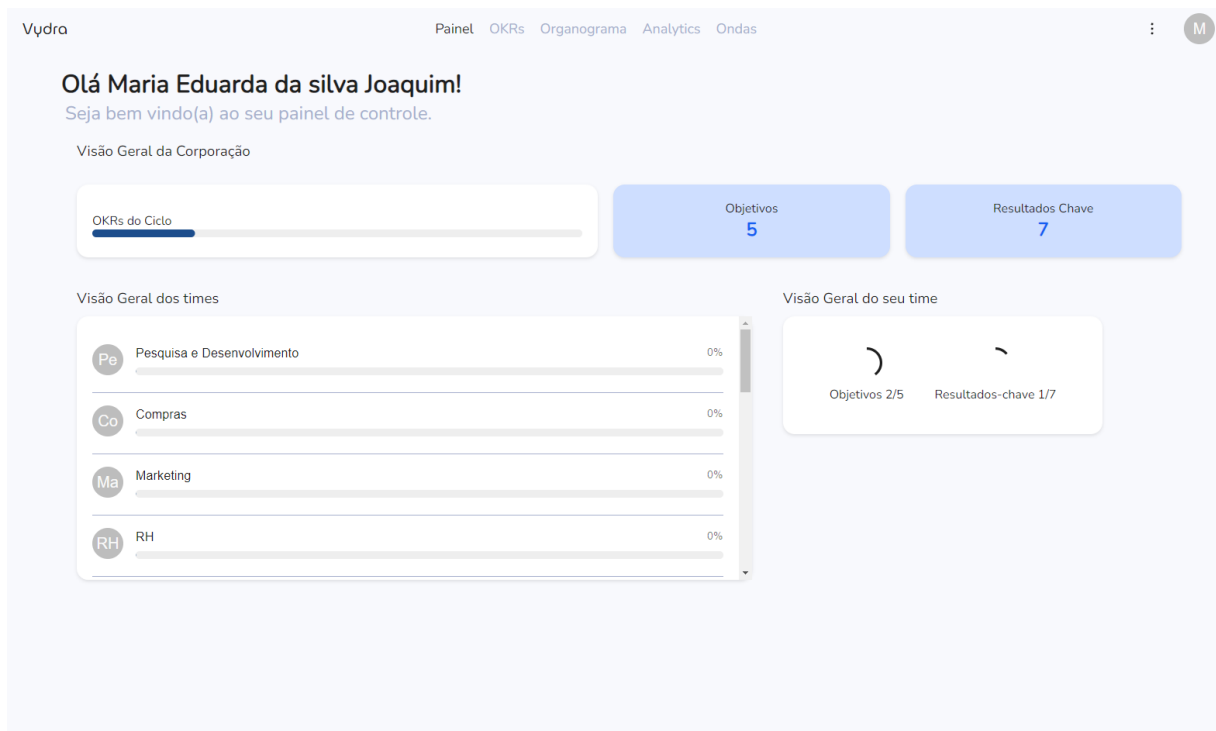
?

A tela de login da aplicação Vydra apresenta um design limpo e moderno. No topo, o nome 'Vydra' é exibido em uma fonte azul. Abaixo dele, o título 'Acessar conta' é destacado em azul escuro. O texto de instrução 'Entre na plataforma com o seu e-mail e senha cadastrados' orienta o usuário. O formulário de login é dividido em duas seções: 'E-mail corporativo' com um campo de texto contendo 'admin' e um campo de senha com pontos cinzas. Um link azul 'Esqueceu a senha?' está posicionado abaixo dos campos. Um botão azul escuro com o texto 'ENTRAR' em branco está centralizado. No canto inferior esquerdo, há um ícone de ajuda (uma interrogação dentro de um círculo azul). À direita, uma ilustração de um foca azul com pontos brancos no rosto e uma faixa de ondas azuis na base, tudo sobre um fundo azul claro que se curva para a esquerda.

Fonte: elaborado pelo autor (2023)

- Painel do usuário: tela na qual o usuário é redirecionado ao efetuar seu login. Nesta tela o usuário consegue observar alguns dados básicos de análise, sendo uma visão geral da empresa, como a quantidade de objetivos e resultados chave, além da barra de porcentagem de conclusão dos objetivos de cada time. Além disso, o usuário consegue visualizar a barra de navegação do sistema, o que permite a navegação entre as demais telas da aplicação.

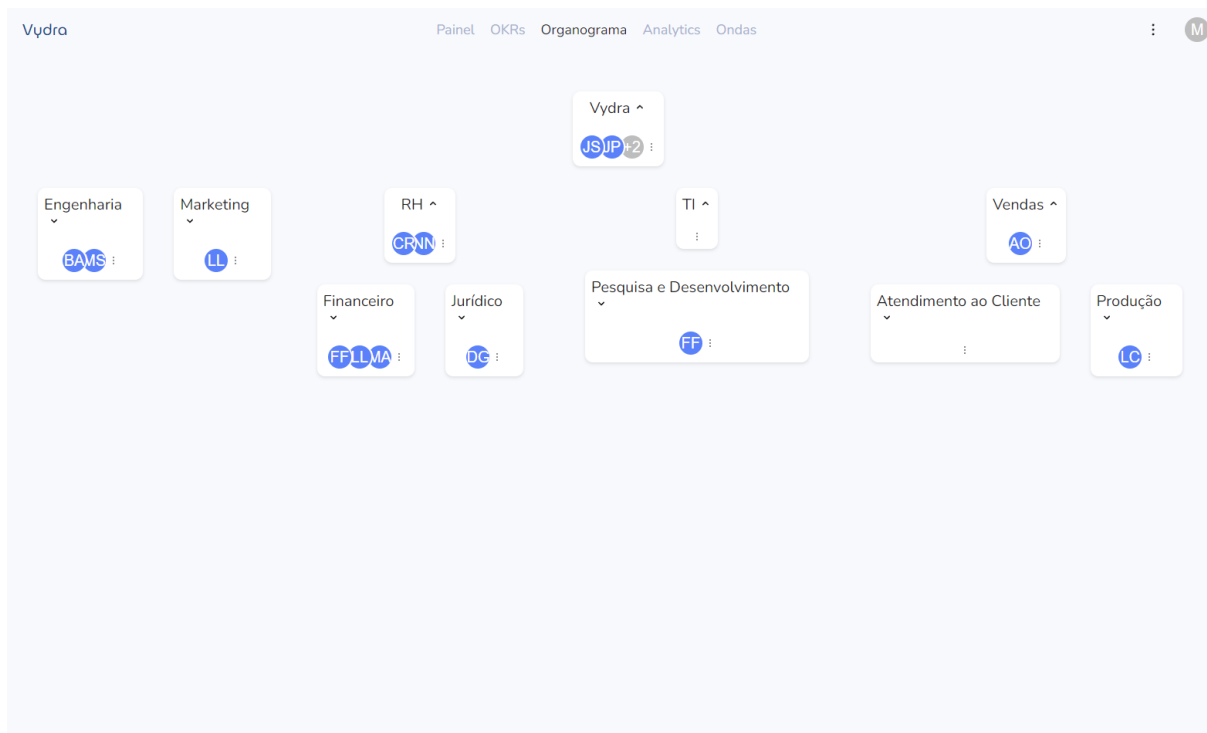
Figura 40 - Tela de painel do usuário da aplicação



Fonte: elaborado pelo autor (2023)

- Tela de organograma: apresenta um organograma empresarial, onde o usuário consegue observar as equipes e os membros que compõe cada equipe da empresa.

Figura 41 - Tela de organograma da aplicação



Fonte: elaborado pelo autor (2023)

- Tela de funcionários: tela na qual o usuário pode observar uma listagem completa de todos os colaboradores da organização, além de alguns dados básicos como o cargo e o time em que ele está situado. No caso de um usuário gestor, há a possibilidade de cadastro de um novo funcionário, atualização e deleção de funcionários.

Figura 42 - Tela de funcionários da aplicação

The screenshot displays the 'Gestão de funcionários' (Employee Management) page in the Vydra application. At the top, there are navigation tabs for 'Painel', 'OKRs', 'Organograma', 'Analytics', and 'Ondas'. The main heading is 'Gestão de funcionários'. Below this, there is a search input field labeled 'Buscar funcionários' and a blue button labeled 'NOVO FUNCIONÁRIO'. The central part of the page features a table with the following data:

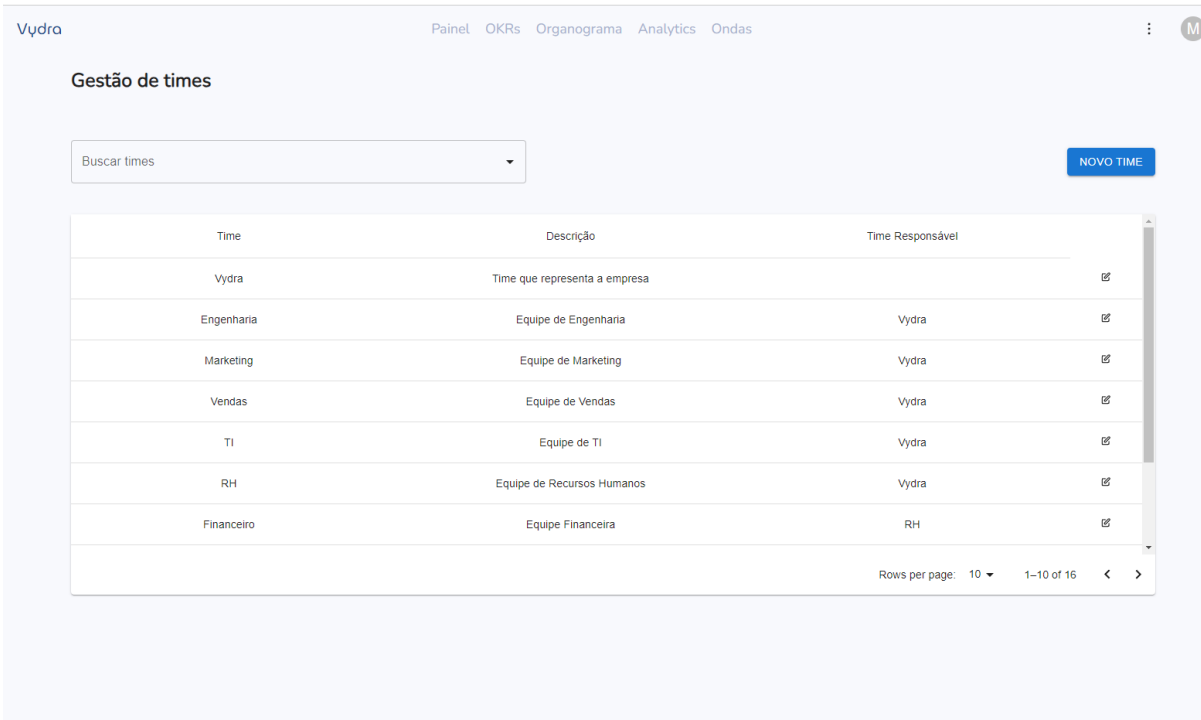
Funcionário	Cargo	Time	
João Silva	Gerente de Engenharia	Vydra	✖
Maria Santos	Especialista de Marketing	Engenharia	✖
Ana Oliveira	Analista de TI	Vendas	✖
Carla Rodrigues	Analista Financeiro	RH	✖
Márcio Alves	Supervisor de Produção	Financeiro	✖
Luciana Costa	Coordenador de Logística	Produção	✖
Daniel Gomes	Cientista de Dados	Jurídico	✖

At the bottom right of the table, there is a pagination control showing 'Rows per page: 10' and '1-10 of 19' with navigation arrows.

Fonte: elaborado pelo autor (2023)

- Tela de times: tela na qual o usuário pode observar uma listagem completa de todos os times da empresa, além de alguns dados básicos como uma breve descrição do time e o time que é responsável, caso exista. No caso de um usuário gestor, há a possibilidade de cadastro de um novo time, atualização e deleção de times.

Figura 43 - Tela de times da aplicação



The screenshot displays the 'Gestão de times' (Team Management) interface. At the top, there is a search bar labeled 'Buscar times' and a blue button labeled 'NOVO TIME'. Below this is a table with the following columns: 'Time', 'Descrição', and 'Time Responsável'. The table lists several teams, including 'Vydra', 'Engenharia', 'Marketing', 'Vendas', 'TI', 'RH', and 'Financeiro'. Each row includes an edit/delete icon. At the bottom right of the table, there is a pagination control showing 'Rows per page: 10' and '1-10 of 16'.

Time	Descrição	Time Responsável
Vydra	Time que representa a empresa	
Engenharia	Equipe de Engenharia	Vydra
Marketing	Equipe de Marketing	Vydra
Vendas	Equipe de Vendas	Vydra
TI	Equipe de TI	Vydra
RH	Equipe de Recursos Humanos	Vydra
Financeiro	Equipe Financeira	RH

Fonte: elaborado pelo autor (2023)

- Tela de perfil do usuário: tela na qual permite que o usuário possa visualizar seus dados cadastrais, além de editar alguns dados como seu nome, sobrenome e atualizar sua senha de usuário.

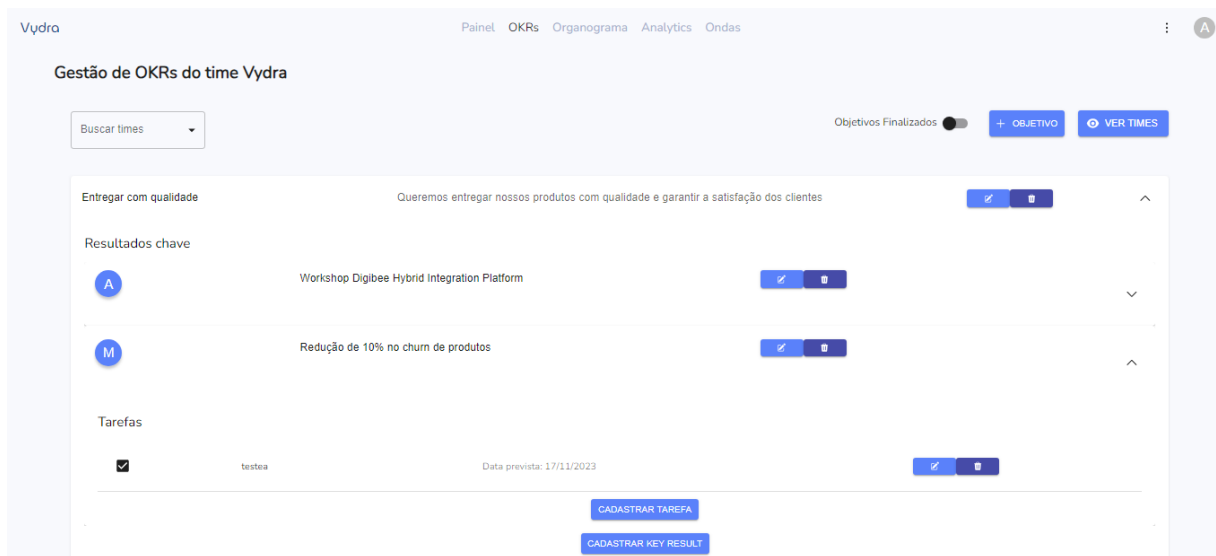
Figura 44 - Tela de perfil do usuário da aplicação

A captura de tela mostra a interface de perfil de usuário da aplicação Vydra. No topo, há uma barra de navegação com o nome 'Vydra' à esquerda e links para 'Painel', 'OKRs', 'Organograma', 'Analytics' e 'Ondas' no centro. À direita da barra, há um ícone de menu e um perfil de usuário com a inicial 'M'. O formulário principal, centralizado, possui um avatar com as iniciais 'Md' em um círculo azul. Abaixo do avatar, há cinco campos de texto rotulados como obrigatórios (Nome*, Sobrenome*, Email*, Time* e Cargo*). Os campos 'Nome*' e 'Sobrenome*' contêm os valores 'Maria Eduarda' e 'da silva Joaquim', respectivamente. O campo 'Email*' contém 'maria@eduada.com'. O campo 'Time*' contém 'Vydra'. O campo 'Cargo*' está vazio. Abaixo dos campos, há dois botões: 'ALTERAR SENHA' (botão desativado) e 'SALVAR' (botão ativo).

Fonte: elaborado pelo autor (2023)

- Tela de OKR: tela que permite que o usuário visualize os objetivos, resultados-chave e tarefas que devem ser executadas pelo time em que o usuário está situado. No caso de um usuário gestor há a possibilidade de criação, atualização e finalização de objetivos, resultados-chave e tarefas do time em questão. A tela de OKR ainda permite a filtragem de OKR, podendo exibir os objetivos em vigência e os objetivos finalizados.

Figura 45 - Tela de OKR da aplicação



Fonte: elaborado pelo autor (2023)

- Tela de respostas as ondas de perguntas: tela na qual as perguntas periódicas sobre clima empresarial são respondidas. As respostas das perguntas fornecidas são armazenadas de forma anônima, com a finalidade de que o colaborador se sinta mais à vontade para responder as perguntas. As respostas possuem numeração que varia de um a sete, como forma de avaliação para a empresa, sendo 1(um) a pontuação mais baixa e 7(sete) a pontuação mais alta.

Figura 46 - Tela de respostas as ondas de perguntas da aplicação

The screenshot displays the Vydra application interface. At the top, the navigation bar includes 'Vydra', 'Painel', 'OKRs', 'Organograma', 'Analytics', and 'Ondas'. A user profile icon with the letter 'M' is visible in the top right corner. The main content area features a survey question: 'Quão satisfeito você está com o seu trabalho atual e seu ambiente de trabalho?'. Below the question is a horizontal scale with seven numbered boxes (1 to 7). The scale is anchored by the text 'Nada satisfeito' on the left and 'Completamente satisfeito' on the right. At the bottom of the scale, there are two blue buttons: 'REVISAR ANTERIOR' and 'PRÓXIMA PERGUNTA'.

Fonte: elaborado pelo autor (2023)

- Tela de Analytics: tela na qual somente usuários gestores possuem acesso. Nessa tela os dados coletados do banco de dados são exibidos em forma de dashboards e gráficos para uso estratégico da organização, desta forma podendo auxiliar a empresa na tomada de decisões, visando a melhora da empresa. Além dos dashboards e gráficos, a tela de analytics possui um chatbox desenvolvido com auxílio de inteligência artificial, onde o gestor pode usar o chatbox para retirada de dúvidas relacionadas a área de analytics do sistema.

Figura 47 - Tela de Analytics da aplicação



Fonte: elaborado pelo autor (2023)

4.5. Verificações e Correções

4.5.1. Testes de Usabilidade

O Vydra passou por vários testes de usabilidade e alguns deles resultaram em melhorias na experiência do usuário. Nestas identificações, foram aprimoradas diversas áreas do software, como a tela de OKRs, que inicialmente era tida como confusa e baixa usabilidade mas, depois de alinhamentos, passou a ter uma interface mais sucinta e intuitiva.

4.5.2. Análise e Revisão do Código

Para garantir a integridade do código, utilizou-se um processo rigoroso de análise e revisão dele antes de cada deploy. Foram avaliados conjuntamente os conjuntos de códigos propostos antes de sua implementação na aplicação, visando identificar e corrigir eventuais problemas e garantir a consistência e qualidade do software.

4.5.3. Correção de Bugs

Durante o desenvolvimento, foram identificados e solucionados bugs que afetavam a experiência do usuário. Na tela de OKRs, por exemplo, foi notado que a interação com os botões dos objetivos acionava inesperadamente o componente "accordion" do React. Essa questão foi prontamente atendida com intuito de melhorar a usabilidade e facilitar a visualização de resultados-chave e tarefas.

5. CONSIDERAÇÕES FINAIS, RECOMENDAÇÕES E LIMITAÇÕES

Ao finalizar o desenvolvimento do sistema de gestão de desempenho, que teve como objetivo ampliar a qualidade do processo de gerência, realizando a integração da gestão de metas com a experiência do funcionário, torna-se importante refletir sobre os resultados obtidos e considerar as perspectivas futuras da aplicação.

No tocante às considerações finais, destaca-se o papel central da sinergia coerente entre a gestão de metas, realizadas através do método OKR, e a experiência do funcionário, desenvolvida com base em um modelo próprio de pesquisa organizacional de clima e engajamento. Esta ligação inovadora resulta em um alcance de resultados organizacionais robustos que ampliam a eficiência no crescimento empresarial de forma que seja saudável à corporação e aos seus colaboradores.

No entanto, é importante mencionar as limitações identificadas durante o planejamento e o desenvolvimento do sistema. Destaca-se entre elas a importância de os clientes terem uma base sobre OKR para melhor aproveitamento dos recursos

do projeto, bem como encorajar e engajar seus colaboradores, pois a resistência cultural pode resultar em uma falha na melhoria de seus resultados com o software.

A garantia de que os funcionários sintam que existe um ambiente de confiança para as suas respostas na pesquisa de clima é algo que sempre deve ser trabalho pelo sistema e pela organização para evitar que haja um baixo engajamento nas respostas, o que resultará em uma falha no processo de identificação de riscos operacionais afetando a melhoria contínua e por conseguinte o atingimento das metas das organizações.

Para a continua progressão do produto desenvolvido é considerado a implementação de módulos complementares no sistema ao longo do processo de implementação e acompanhamento de resultados em conjunto aos clientes, como a criação de um processo de programa de desenvolvimento pessoal, para que as limitações relacionadas ao engajamento sejam minimizadas.

Desta forma, deve-se manter a continuidade do monitoramento e aprimoramento constante do sistema. A coleta contínua de feedbacks dos usuários e as adaptações às mudanças de mercado serão um fator extremamente relevante para o crescimento contínuo da aplicação. Além disso é considerado essencial a promoção de uma cultura organizacional que valorize a transparência, a comunicação aberta e a colaboração, pois estes são elementos fundamentais para o sucesso de um sistema de gestão de desempenho.

Em síntese, este projeto não apenas proporcionou uma visão aprofundada sobre a interseção entre gestão de metas e experiência do funcionário, mas também ofereceu contribuições práticas para aprimorar a performance organizacional. Contudo este processo não se encerra aqui, e os desafios e oportunidades servem como base para futuras explorações e refinamentos do produto. Este trabalho representa não apenas uma conclusão, mas um ponto de partida para a evolução contínua no âmbito da gestão de desempenho organizacional.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Análise comparativa entre modelos de pesquisa em clima organizacional. [S. l.], 28 ago. 2014. Disponível em: <https://www.revistaespacios.com/a14v35n09/14350916.html>. Acesso em: 8 set. 2023.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. **NBR 10719: informação e documentação – Relatório Técnico e/ou científico – Apresentação.** Rio de Janeiro, 2011.

ÁVILA, Rafael. **Os 5 pilares da gestão de pessoas. 2018.** Disponível em: <<https://jornadadogestor.com.br/o-que-e/os-5-pilares-da-gestao-de-pessoas/>>. Acesso em: 18 nov. 2022.

BEZERRA, Giovanna. **Benefícios da utilização do Objectives and key results (OKR) em diferentes áreas de aplicação.** 2021. Disponível em: <<http://repositorioinstitucional.uea.edu.br/handle/riuea/3743/>>. Acesso em: 17 nov. 2022.

CAMFIELD, CLAUDIO EDUARDO RAMOS; MOGNOL, CAMILA; PICETI, PATRICIA; GODOY, TAIS PENTIADO; WEGNER, ROGER DA SILVA. **PESQUISA DE CLIMA ORGANIZACIONAL: UMA PESQUISA EMPÍRICA COM BASE EM DOIS MODELOS.** XXI SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, [S. l.], p. 1-14, 10 nov. 2014. Disponível em: https://www.researchgate.net/profile/Claudio-Camfield-2/publication/308710477_ORGANIZATIONAL_CLIMATE_RESEARCH_AN_EMPIRICAL_RESEARCH_BASED_ON_TWO_MODELS/links/57ec2a5b08ae92eb4d264b51/ORGANIZATIONAL-CLIMATE-RESEARCH-AN-EMPIRICAL-RESEARCH-BASED-ON-TWO-MODELS.pdf. Acesso em: 8 set. 2023.

CAMPOS, Sarah C. **Gestão de Pessoas. TECNOLOGIA EM HOTELARIA.** 2009. Disponível em: <<https://educapes.capes.gov.br/bitstream/capes/429526/2/gestao-de-pessoas.pdf>>. Acesso em: 18 nov. 2022.

CARVALHO, Henrique. **Tudo sobre OKR: o que é, e como implementar**. Disponível em: <<https://vidadeproduto.com.br/okr/>> Acesso em: 15 nov. 2022.

DA CUNHA, Paulo Roberto; COSTA, Filipe Jose; KRESPI, Nayane Thais; GOMES, Giancarlo; DA SILVA, Tarcísio Pedro. **PESQUISA DE CLIMA ORGANIZACIONAL: UM ESTUDO REALIZADO EM UMA INDÚSTRIA DE ALIMENTOS NA CIDADE DE GASPAR, SC.** [S. l.], 2014. Disponível em: <https://dialnet.unirioja.es/descarga/articulo/5160893.pdf>. Acesso em: 5 set. 2023.

Litwin, G. H., & Stringer, R. A. (1968). **Motivation and organizational climate**. Cambridge: Harvard University Press.

MARKOVIC, Isidora. **What is a 'Pulse Survey'?**. Disponível em: <<https://www.edume.com/blog/pulse-survey>>. Acesso em: 01 dez. 2022.

MARQUES, José R. 2020. **PLANO DE DESENVOLVIMENTO INDIVIDUAL – O QUE É O PDI?**. Disponível em: <<https://www.ibccoaching.com.br/portal/coaching-carreira/plano-de-desenvolvimento-individual-o-que-e-o-pdi/>>. Acesso em: 28 nov. 2022.

MARTINS, Kaio T. 2019. Proposta de abordagem para identificar a efetividade do plano de desenvolvimento individual com base nas avaliações de desempenho. Disponível em: <<https://app.uff.br/riuff/handle/1/13062>>. Acesso em: 25 nov. 2022.

NIVEN, Paul R. and LAMORTE, Ben. (2016). **Objectives and Key Results: Driving Focus, Alignment, and Engagement with OKRs**. Wiley.

OLIVEIRA, Adrielle M. 2021. **DO ONBOARDING AO OFFBOARDING: contribuições da Gestão do Conhecimento para a experiência do colaborador na Gestão de Pessoas**. Disponível em: <<https://repositorio.ufpe.br/bitstream/123456789/41732/1/ADRIELE%20MARQUES%20DE%20OLIVEIRA.pdf>>. Acesso em: 01 dez 2022.

OLIVEIRA, Vinícius R. 2018. **EMPLOYEE EXPERIENCE: A COMPLEXIDADE DA EXPERIÊNCIA DOS FUNCIONÁRIOS EM AMBIENTES ORGANIZACIONAIS.**

Disponível em: <http://www.comcult.cisc.org.br/wp-content/uploads/2019/05/GT4_Vini%CC%81cius_Riqueto_de_Oliveira_FACA SPER.pdf>. Acesso em: 01 dez. 2022.

PERINI M; M e ZANOL, S. C. **Gestão de Pessoas: guia para entender o que é e como fazer.** 2022. Disponível em: <<https://www.metadados.com.br/blog/gestao-de-pessoas>>. Acesso em: 18 nov. 2022.

RIGO, Charles; MACIEL, Alessandra Costenaro; OLIVA, Eduardo de Camargo; BAGGIO, Daniel Knebel; DE CAMPOS, Bethina FERRAZ. **Clima organizacional: um estudo em uma Indústria e Comerciante de Ferragens.** Revista ESPACIOS, [S. I.], p. 2-13, [2017].

SANTANA, Thiago Miguez de. **Metodologia OKR: Um Estudo Multicaso em Startups Baianas.** Revista Formadores - Vivências e Estudos, Cachoeira - Bahia, v. 12, n. 8, p. 54- 76, Dez, 2019.

SBRAGIA, R. **Um estudo empírico sobre clima organizacional em instituições de pesquisa.** Revista de Administração, v. 18, n. 2, p. 30-39, 1983.

WELBOURNE, Theresa M. 2016. **The Potential of Pulse Surveys: Transforming Surveys into Leadership Tools. Employment Relations Today.** Disponível em: <<http://eepulse.com/files/2017/12/The-potential-of-pulse-surveys-TWelbourne.pdf>>. Acesso em: 01 dez. 2022.

WHITMORE, John; **Coaching para Performance: Aprimorando Pessoas, Desempenhos e Resultados.** v.1, p.5, p.11 , p.54 - 55, p.65, p.76. 2006.

BISPO, Carlos Alberto Ferreira. Um novo modelo de pesquisa de clima organizacional. [S. I.], 4 out. 2006. Disponível em: <https://www.scielo.br/j/prod/a/4Cy7Wz5QsYJrPBnQBWt5R7x/>. Acesso em: 5 set. 2023.

7. APÊNDICES

7.1. Apêndice A – Modelo de pesquisa organizacional de clima e engajamento

7.1.1. Liderança e Visão Estratégica

Sua liderança demonstra os valores e comportamentos esperados pelos funcionários. (1- Discordo totalmente) (7 - Concordo plenamente)

Em que medida sua liderança está acessível a suas sugestões e preocupações? (1- Nada acessível) (7 - Muito acessível)

O quanto seu líder te inspira para dar o seu melhor no trabalho? (1- Não me sinto inspirado(a) (7 - Me sinto muito inspirado(a))

Sobre seu líder ser alinhado com os objetivos estratégicos da organização. (1- Não é alinhado) (7 - É muito conectado)

7.1.2. Comunicação e Feedback

O que você acha das comunicações internas e informações sobre mudanças na empresa? (1- Acho pouco transparente e irrelevantes) (7 - Acho muito claras e relevantes)

Com que frequência você recebe feedback sobre o desempenho no trabalho? (1 - Nunca recebo feedbacks) (7 - Em todos os momentos pertinentes)

Você se sente confortável para compartilhar suas ideias e preocupações com seus colegas e gestores? (1 - Me sinto nada confortável) (7 - Sempre me sinto confortável)

Há o incentivo e a prática de uma comunicação não violenta na organização? (1 - Não existe incentivo e prática) (7 - Sempre há o incentivo e prática)

7.1.3. Aprendizado e Desenvolvimento

Existem oportunidades de treinamento e desenvolvimento disponíveis para você na empresa? (1 - Não, nenhuma) (7 - Sim, muitas)

Você acredita que a empresa incentiva ativamente o aprendizado contínuo dos funcionários? (1 - Nunca incentiva) (7 - Sempre há o incentivo)

O quão acessível são os recursos que te mantém atualizado em relação às melhores práticas do setor? (1 - Nada acessível) (7 - É de fácil acesso)

A empresa promove a troca de conhecimento entre os funcionários? (1 - Nunca promove) (7 - Promove muitas vezes)

7.1.4. Envolvimento dos Funcionários

Quão comprometido você se sente com a missão e os objetivos da empresa? (1 - Nada comprometido) (7 - Estou muito comprometido)

O quanto você acredita que a sua contribuição individual é valorizada pela empresa? (1 - Não sou valorizado(a)) (7 - Me sinto fundamental)

Qual o nível de conexão entre você e seus colegas de equipe? (1 - Não há conexão alguma) (7 - Estamos muito conectados)

As iniciativas e eventos específicos da empresa o deixaram mais engajado e motivado? (1 - Nunca me sinto motivado) (7 - Sempre me engajo)

7.1.5. Reconhecimento e Recompensas

Se fosse oferecido um emprego com o mesmo salário, você pensaria no caso? (1 - Com certeza) (7 - De forma alguma)

Você acredita que a empresa reconhece adequadamente o seu desempenho e contribuição? (1 - Nunca sou reconhecido) (7 - Sempre sou reconhecido)

O sistema de reconhecimento e recompensas da empresa é justo e transparente? (1 - Não é nada justo) (7 - Sim, é muito transparente e justo)

Você está ciente dos programas de recompensas da empresa e como eles funcionam? (1 - Não tenho conhecimento) (7 - Tenho plena ciência)

7.1.6. Participação e Tomada de Decisão

Você se sente encorajado(a) a compartilhar suas ideias e sugestões para melhorias na empresa? (1 - Nunca fui encorajado(a)) (7 - Sempre há encorajamento)

Você sente que há liberdade e autonomia para tomar decisões na sua rotina de trabalho? (1 - Não tenho liberdade) (7 - Sempre posso tomá-las)

Com qual frequência sua opinião é levada em consideração nas decisões que afetam o seu trabalho ou equipe? (1 - Nunca é levada) (7 - Sempre é considerada)

O quão valorizada é a diversidade de perspectivas nas tomadas de decisões? (1 - Nada valorizada) (7 - Muito valorizada)

7.1.7. Satisfação e Bem-Estar

Quão satisfeito você está com o seu trabalho atual e seu ambiente de trabalho? (1 - Nada satisfeito) (7 - Completamente satisfeito)

Você se sente encorajado a buscar ajuda ou recursos para lidar com o estresse ou problemas de bem-estar no trabalho? (1 - Nunca me sinto encorajado) (7 - Sempre me sinto encorajado)

Existem aspectos do trabalho que impactam negativamente o seu bem-estar e satisfação? (1 - Existem vários) (7 - Não, nada impacta)

Há suporte necessário para manter um bom estado do seu bem-estar no trabalho? (1 - Não há suporte) (7 - Sempre estou amparado)

7.2. Apêndice B - Especificações de Caso de Uso

Tabela 6 - Caso de uso "Visualizar funcionários"

Nome: Visualizar funcionários
<p>Atores:</p> <ul style="list-style-type: none"> ● Atores Primários: Usuário colaborador e usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none"> ● Permitir que colaboradores e gestores visualizem a lista de funcionários.

Pré-condições:

- O usuário está autenticado no sistema.

Pós-condições:

- O sistema exibe a lista de funcionários para o usuário.

Fluxo Principal:

1. O usuário acessa o menu de opções de exibição.
2. O usuário seleciona a opção "Funcionários".
3. O sistema exibe a lista de funcionários, incluindo nomes, cargos e seus respectivos times.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 7 - Caso de uso "Concluir tarefas"

Nome: Concluir tarefas

Atores:

- Atores Primários: Usuário colaborador e usuário gestor

Objetivo:

- Permitir que colaboradores e gestores concluam tarefas.

Pré-condições:

- O usuário está autenticado no sistema.

Pós-condições:

- As alterações feitas pelo usuário são refletidas na listagem de tarefas.

Fluxo Principal:

1. O usuário seleciona a opção “Detalhes” no menu de opções do resultado-chave.
2. O sistema exibe a lista de tarefas, incluindo o nome, a data prevista de cada e seu respectivo menu de opções, podendo concluir a tarefa.
3. O usuário seleciona a caixa de seleção.
4. O sistema exibe uma mensagem questionando se o usuário tem certeza que deseja concluir a tarefa.
5. O usuário seleciona a opção “Sim”.
6. O sistema atualiza os dados da tarefa conforme a modificação feita.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 8 - Caso de uso "Editar funcionários"

Nome: Editar funcionários**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores editem informações dos funcionários, como nome, sobrenome, cargo e time.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo usuário são refletidas nos perfis dos funcionários.

Fluxo Principal:

1. O usuário seleciona um funcionário para editar.
2. O sistema apresenta um formulário com os detalhes atuais do funcionário, permitindo a edição.
3. O usuário realiza as alterações desejadas, como atualização de informações de cargo ou time.
4. O usuário seleciona o botão “Salvar”.
5. O sistema atualiza os dados do funcionário conforme as modificações feitas.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Informações faltantes no formulário:

- O sistema não permite a edição daquele funcionário e retorna ao formulário com as informações preenchidas anteriormente.

Fonte: elaborado pelo autor (2023)

Tabela 9 - Caso de uso "Visualizar times"

Nome: Visualizar times
<p>Atores:</p> <ul style="list-style-type: none">• Atores Primários: Usuário colaborador e usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none">• Permitir que colaboradores e gestores visualizem a lista de times. <p>Pré-condições:</p> <ul style="list-style-type: none">• O usuário está autenticado no sistema. <p>Pós-condições:</p> <ul style="list-style-type: none">• O sistema exibe a lista de times para o usuário.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário acessa o menu de opções de exibição.2. O usuário seleciona a opção "Times".3. O sistema exibe a lista de times, incluindo nomes, descrição e seus responsáveis.
<p>Fluxos Alternativos:</p> <p>1a Usuário não autenticado:</p> <ul style="list-style-type: none">• O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 10 - Caso de uso "Editar times"

Nome: Editar times
<p>Atores:</p> <ul style="list-style-type: none">• Atores Primários: Usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none">• Permitir que gestores editem informações dos times, como descrição e time responsável. <p>Pré-condições:</p> <ul style="list-style-type: none">• O usuário está autenticado no sistema.• O usuário possui permissões de gestor. <p>Pós-condições:</p> <ul style="list-style-type: none">• As alterações feitas pelo usuário são refletidas na listagem de times.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário seleciona um time para editar.2. O sistema apresenta um formulário com os detalhes atuais do time, permitindo a edição.3. O usuário realiza as alterações desejadas.4. O usuário seleciona o botão "Salvar".5. O sistema atualiza os dados do time conforme as modificações feitas.
<p>Fluxos Alternativos:</p> <p>1a Usuário não autenticado:</p> <ul style="list-style-type: none">• O sistema redireciona o usuário para a tela de login.

2a Informações faltantes no formulário:

- O sistema não permite a edição daquele time e retorna ao formulário com as informações preenchidas anteriormente.

Fonte: elaborado pelo autor (2023)

Tabela 11 - Caso de uso "Visualizar objetivos"

Nome: Visualizar objetivos
<p>Atores:</p> <ul style="list-style-type: none">● Atores Primários: Usuário colaborador e usuário gestor
<p>Objetivo:</p> <ul style="list-style-type: none">● Permitir que colaboradores e gestores visualizem a lista de objetivos.
<p>Pré-condições:</p> <ul style="list-style-type: none">● O usuário está autenticado no sistema.
<p>Pós-condições:</p> <ul style="list-style-type: none">● O sistema exibe a lista de objetivos para o usuário.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário acessa a seção "OKRs" no menu principal.2. O sistema exibe a lista de objetivos, incluindo nomes, descrição e seu respectivo menu de opções, podendo exibir os resultados-chave desse objetivo.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 12 - Caso de uso "Editar objetivos"

Nome: Editar objetivos**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores editem informações dos objetivos, como nome, data-limite e descrição.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo usuário são refletidas na listagem de objetivos.

Fluxo Principal:

1. O usuário seleciona um objetivo para editar.
2. O sistema apresenta um formulário com os detalhes atuais do objetivo, permitindo a edição.
3. O usuário realiza as alterações desejadas.
4. O usuário seleciona o botão "Salvar".
5. O sistema atualiza os dados do objetivo conforme as modificações feitas.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Informações faltantes no formulário:

- O sistema não permite a edição daquele objetivo e retorna ao formulário com as informações preenchidas anteriormente.

Fonte: elaborado pelo autor (2023)

Tabela 13 - Caso de uso "Visualizar resultados-chave"

Nome: Visualizar resultados-chave

Atores:

- Atores Primários: Usuário colaborador e usuário gestor

Objetivo:

- Permitir que colaboradores e gestores visualizem a lista de resultados-chave.

Pré-condições:

- O usuário está autenticado no sistema.

Pós-condições:

- O sistema exibe a lista de resultados-chave para o usuário.

Fluxo Principal:

1. O usuário acessa a seção "OKRs" no menu principal.
2. O sistema exibe a lista de objetivos, incluindo nomes, descrição e seu respectivo menu de opções, podendo exibir os resultados-chave desse objetivo.
3. O usuário seleciona a opção "Detalhes" no menu de opções do objetivo.
4. O sistema exibe a lista de resultados-chave, incluindo o ícone do responsável, o nome e a data-limite de cada um.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

Tabela 14 - Caso de uso "Editar resultados-chave"

Nome: Editar resultados-chave
<p>Atores:</p> <ul style="list-style-type: none">• Atores Primários: Usuário gestor
<p>Objetivo:</p> <ul style="list-style-type: none">• Permitir que gestores editem informações dos resultados-chave, como nome, data-limite, descrição e responsável.
<p>Pré-condições:</p> <ul style="list-style-type: none">• O usuário está autenticado no sistema.• O usuário possui permissões de gestor.
<p>Pós-condições:</p> <ul style="list-style-type: none">• As alterações feitas pelo usuário são refletidas na listagem de resultados-chave.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário seleciona um resultado-chave para editar.2. O sistema apresenta um formulário com os detalhes atuais do resultado-chave, permitindo a edição.3. O usuário realiza as alterações desejadas.4. O usuário seleciona o botão "Salvar".5. O sistema atualiza os dados do resultado-chave conforme as modificações feitas.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Informações faltantes no formulário:

- O sistema não permite a edição daquele resultado-chave e retorna ao formulário com as informações preenchidas anteriormente.

Fonte: elaborado pelo autor (2023)

Tabela 15 - Caso de uso "Visualizar tarefas"

Nome: Visualizar tarefas**Atores:**

- Atores Primários: Usuário colaborador e usuário gestor

Objetivo:

- Permitir que colaboradores e gestores visualizem a lista de tarefas.

Pré-condições:

- O usuário está autenticado no sistema.

Pós-condições:

- O sistema exibe a lista de tarefas para o usuário.

Fluxo Principal:

1. O usuário acessa a seção "OKRs" no menu principal.
2. O sistema exibe a lista de objetivos, incluindo nomes, descrição e seu respectivo menu de opções, podendo exibir os resultados-chave desse objetivo.
3. O usuário seleciona a opção "Detalhes" no menu de opções do objetivo.
4. O sistema exibe a lista de resultados-chave, incluindo o ícone do responsável, o nome, a data-limite e seu respectivo menu de opções, podendo exibir as tarefas desse resultado-chave.
5. O usuário seleciona a opção "Detalhes" no menu de opções do resultado-chave.
6. O sistema exibe a lista de tarefas, incluindo o nome, a data prevista de cada e o botão para conclusão da tarefa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 16 - Caso de uso "Editar tarefas"

Nome: Editar tarefas
<p>Atores:</p> <ul style="list-style-type: none">• Atores Primários: Usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none">• Permitir que gestores editem informações das tarefas, como nome, data-limite e descrição. <p>Pré-condições:</p> <ul style="list-style-type: none">• O usuário está autenticado no sistema.• O usuário possui permissões de gestor. <p>Pós-condições:</p> <ul style="list-style-type: none">• As alterações feitas pelo usuário são refletidas na listagem de tarefas.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário seleciona uma tarefa para editar.2. O sistema apresenta um formulário com os detalhes atuais da tarefa, permitindo a edição.3. O usuário realiza as alterações desejadas.4. O usuário seleciona o botão "Salvar".5. O sistema atualiza os dados da tarefa conforme as modificações feitas.
<p>Fluxos Alternativos:</p> <p>1a Usuário não autenticado:</p> <ul style="list-style-type: none">• O sistema redireciona o usuário para a tela de login. <p>2a Informações faltantes no formulário:</p>

- O sistema não permite a edição daquela tarefa e retorna ao formulário com as informações preenchidas anteriormente.

Fonte: elaborado pelo autor (2023)

Tabela 17 - Caso de uso "Visualizar informações de perfil"

Nome: Visualizar informações de perfil
<p>Atores:</p> <ul style="list-style-type: none"> • Atores Primários: Usuário colaborador e usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none"> • Permitir que colaboradores e gestores visualizem as suas informações, tais como nome, sobrenome, email, time e cargo. <p>Pré-condições:</p> <ul style="list-style-type: none"> • O usuário está autenticado no sistema. <p>Pós-condições:</p> <ul style="list-style-type: none"> • As informações do funcionário são exibidas na tela de perfil.
<p>Fluxo Principal:</p> <ol style="list-style-type: none"> 1. O usuário seleciona a imagem de seu ícone. 2. O sistema exibe as informações correspondentes de seu perfil.
<p>Fluxos Alternativos:</p> <p>1a Usuário não autenticado:</p> <ul style="list-style-type: none"> • O sistema redireciona o usuário para a tela de login.

Fonte: elaborado pelo autor (2023)

Tabela 18 - Caso de uso "Visualizar organograma"

Nome: Visualizar organograma
<p>Atores:</p> <ul style="list-style-type: none">• Atores Primários: Usuário colaborador e usuário gestor <p>Objetivo:</p> <ul style="list-style-type: none">• Permitir que colaboradores e gestores visualizem as informações do organograma da empresa, incluindo os times e os funcionários que o incluem. <p>Pré-condições:</p> <ul style="list-style-type: none">• O usuário está autenticado no sistema. <p>Pós-condições:</p> <ul style="list-style-type: none">• O organograma é apresentado ao usuário.
<p>Fluxo Principal:</p> <ol style="list-style-type: none">1. O usuário acessa a seção "Organograma" no menu principal.2. O sistema exibe o organograma da empresa, informando o time que não possui responsável e dando a opção do usuário de ver as ramificações deste e dos demais que forem listados.
<p>Fluxos Alternativos:</p> <p>1a Usuário não autenticado:</p> <ul style="list-style-type: none">• O sistema redireciona o usuário para a tela de login. <p>2a O usuário seleciona o menu de opções do time apresentado:</p> <ul style="list-style-type: none">• O sistema exibe os integrantes daquele time e dá a opção do usuário visualizar os OKRs deste. <p>3a O usuário seleciona o botão "OKRS":</p>

- O sistema é redirecionado para a tela de OKRs do time selecionado anteriormente.

Fonte: elaborado pelo autor (2023)

Tabela 19 - Caso de uso "Criar times"

Nome: Criar Times

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores criem os times da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de times e no organograma da empresa.

Fluxo Principal:

1. O gestor seleciona o menu de configurações ao lado do avatar do usuário;
2. O gestor seleciona a opção "Times";
3. O gestor seleciona o botão "Novo Time";
4. O gestor preenche o nome, descrição e seleciona o time responsável pelo time criado, caso exista;
5. O gestor seleciona o botão "Salvar"; e
6. O novo time já estará listado entre os times da empresa e no organograma empresarial.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na criação do time:

- O gestor seleciona o botão “Cancelar” e retorna a página de times.

3a Informações faltantes no formulário:

- O sistema não permite a criação do novo time e retorna a página de times.

Fonte: elaborado pelo autor (2023)

Tabela 20 - Caso de uso "Atualizar times"

Nome: Atualizar times**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores editem os times da empresa.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de times e no organograma da empresa.

Fluxo Principal:

1. O gestor seleciona o menu de configurações ao lado do avatar do usuário;
2. O gestor seleciona a opção “Times”;
3. O gestor visualiza o time a ser editado e seleciona no menu de opções;
4. O gestor seleciona a opção “Atualizar time”;
5. O gestor altera os dados necessários podendo ser o nome, descrição e o time responsável pelo time a ser alterado;
6. O gestor seleciona o botão “Salvar”; e
7. O time já estará atualizado entre os times da empresa e no organograma empresarial.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na atualização do time:

- O gestor seleciona o botão “Cancelar” e retorna a página de times.

3a Informações faltantes no formulário:

- O sistema não permite a atualização do time e retorna a página de times.

Fonte: elaborado pelo autor (2023)

Tabela 21 - Caso de uso "Excluir times"

Nome: Excluir times

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores excluam os times da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de times e no organograma da empresa.

Fluxo Principal:

1. O gestor seleciona o menu de configurações ao lado do avatar do usuário;
2. O gestor seleciona a opção “Times”;
3. O gestor visualiza o time a ser excluído e seleciona no menu de opções;
4. O gestor seleciona a opção “Deletar time”;
5. Uma tela de confirmação de exclusão aparecerá;
6. O gestor seleciona o botão “Deletar”; e
7. O time estará excluído.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na exclusão do time:

- O gestor seleciona o botão “Cancelar” e retorna a página de times.

Fonte: elaborado pelo autor (2023)

Tabela 22 - Caso de uso "Criar funcionários"

Nome: Criar funcionários**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores criem os funcionários da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de funcionários e no organograma da empresa.

Fluxo Principal:

1. O gestor seleciona o menu de configurações ao lado do avatar do usuário;
2. O gestor seleciona a opção “Funcionários”;
3. O gestor seleciona o botão “Novo funcionário”;
4. O gestor preenche o nome, sobrenome, email, data de nascimento, seleciona o cargo e o time do funcionário;
5. O gestor seleciona o botão “Salvar”; e
6. O novo funcionário já estará listado entre os funcionários da empresa e no organograma empresarial.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na criação do funcionário:

- O gestor seleciona o botão “Cancelar” e retorna a página de funcionários.

3a Informações faltantes no formulário:

- O sistema não permite a criação do novo funcionário e retorna a página de funcionários.

Fonte: elaborado pelo autor (2023)

Tabela 23 - Caso de uso "Atualizar funcionários"

Nome: Atualizar funcionários

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores atualizem os dados dos funcionários da empresa.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de funcionários e no organograma da empresa.

Fluxo Principal:

1. O gestor seleciona o menu de configurações ao lado do avatar do usuário;
2. O gestor seleciona a opção “Funcionários”;
3. O gestor visualiza o funcionário a ser editado e seleciona o menu de opções;
4. O gestor seleciona a opção “Editar funcionário”;
5. O gestor pode alterar o nome, sobrenome, email, data de nascimento, cargo e time do funcionário;
6. O gestor seleciona o botão “Salvar”; e
7. O funcionário estará com seus dados atualizados, listado entre os funcionários da empresa e no organograma empresarial.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na atualização do funcionário:

- O gestor seleciona o botão “Cancelar” e retorna a página de funcionários.

3a Informações faltantes no formulário:

- O sistema não permite a atualização do funcionário e retorna a página de funcionários.

Fonte: elaborado pelo autor (2023)

Tabela 24 - Caso de uso "Excluir funcionários"

Nome: Excluir funcionários**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores excluam os funcionários da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de funcionários e no organograma da empresa.

Fluxo Principal:

1. O gestor seleciona o menu de configurações ao lado do avatar do usuário;
2. O gestor seleciona a opção “Funcionários”;
3. O gestor visualiza o funcionário a ser excluído e seleciona no menu de opções;
4. O gestor seleciona a opção “Deletar funcionário”;
5. Uma tela de confirmação de exclusão aparecerá;
6. O gestor seleciona o botão “Deletar”; e
7. O funcionário estará excluído.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na exclusão do funcionário:

- O gestor seleciona o botão “Cancelar” e retorna a página de funcionários.

Fonte: elaborado pelo autor (2023)

Tabela 25 - Caso de uso "Atualizar objetivos"

Nome: Atualizar objetivos

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores atualizem os objetivos da empresa.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de objetivos.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor visualiza o objetivo a ser alterado e seleciona a opção de edição;
3. O gestor pode alterar o nome, data limite, descrição e finalizar o objetivo;
4. O gestor seleciona o botão “Salvar”; e
5. O objetivo atualizado estará listado na lista de objetivos da empresa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na atualização do objetivo:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

3a Informações faltantes no formulário:

- O sistema não permite a atualização do novo objetivo e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 26 - Caso de uso "Excluir objetivos"

Nome: Excluir objetivos**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores excluam os objetivos da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de objetivos.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor visualiza o objetivo a ser excluído e seleciona a opção de deleção;
3. Uma mensagem de confirmação de exclusão aparecerá;
4. O gestor seleciona o botão “Deletar”; e
5. O objetivo estará excluído.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na exclusão do objetivo:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 27 - Caso de uso "Criar resultados-chave"

Nome: Criar resultados-chave

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores criem os resultados-chave da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de resultados-chave nos objetivos da empresa.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor visualiza o objetivo no qual será criado o resultado chave e seleciona o botão de expansão do objetivo;
3. O gestor seleciona a opção “Cadastrar key result”;
4. O gestor preenche o nome, data limite, seleciona o responsável pelo resultado-chave e descrição do novo resultado-chave;
5. O gestor seleciona o botão “Salvar”;
6. O novo resultado-chave já estará listado na lista deste objetivo da empresa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na criação do resultado-chave:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

3a Informações faltantes no formulário:

- O sistema não permite a criação do novo resultado-chave e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 28 - Caso de uso "Atualizar resultados-chave"

Nome: Atualizar resultados-chave**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores atualizem os resultados-chave da empresa.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de resultados-chave nos objetivos da empresa.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor visualiza o objetivo no qual o resultado chave será atualizado e seleciona o botão de expansão do objetivo;
3. O gestor visualiza o resultado-chave que será atualizado e seleciona a opção de edição;
4. O gestor pode atualizar o nome, data limite, responsável pelo resultado-chave, descrição e finalizar o resultado-chave editado;
5. O gestor seleciona o botão “Salvar”; e
6. O resultado-chave estará atualizado e listado na lista deste objetivo da empresa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na atualização do resultado-chave:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

3a Informações faltantes no formulário:

- O sistema não permite a atualização do resultado-chave e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 29 - Caso de uso "Excluir resultados-chave"

Nome: Excluir resultados-chave

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores excluam os resultados-chave da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de resultados-chave nos objetivos da empresa.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor visualiza o objetivo no qual o resultado chave será excluído e seleciona botão de expansão do objetivo;
3. O gestor visualiza o resultado-chave que será excluído e seleciona a opção de exclusão;
4. Uma mensagem de confirmação de exclusão aparecerá;
5. O gestor seleciona o botão “Deletar”; e
6. O resultado-chave estará excluído.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na exclusão do resultado-chave:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 30 - Caso de uso "Criar resultados-chave"

Nome: Criar resultados-chave**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores criem os resultados-chave da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de resultados-chave nos objetivos da empresa.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor visualiza o objetivo no qual será criado o resultado chave e seleciona o botão de expansão do objetivo;
3. O gestor seleciona a opção “Cadastrar key result”;
4. O gestor preenche o nome, data limite, seleciona o responsável pelo resultado-chave e descrição do novo resultado-chave;
5. O gestor seleciona o botão “Salvar”;
6. O novo resultado-chave já estará listado na lista deste objetivo da empresa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na criação do resultado-chave:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

3a Informações faltantes no formulário:

- O sistema não permite a criação do novo resultado-chave e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 31 - Caso de uso "Criar tarefas"

Nome: Criar tarefas

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores criem as tarefas dos resultados-chave da empresa dentro do sistema.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de tarefas nos resultados-chave.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação;
2. O gestor seleciona a expansão do objetivo em que a tarefa será inserida;
3. O gestor seleciona a expansão do resultado-chave em que a tarefa será inserida;
4. O gestor seleciona a opção “Cadastrar tarefa”;
5. O gestor preenche o nome, data limite e descrição da nova tarefa;
6. O gestor seleciona o botão “Salvar”; e
7. A nova tarefa já estará listado na lista deste resultado-chave da empresa.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na criação da tarefa:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs.

3a Informações faltantes no formulário:

- O sistema não permite a criação de nova tarefa e retorna a página de OKRs.

Fonte: elaborado pelo autor (2023)

Tabela 32 - Caso de uso "Excluir tarefas"

Nome: Excluir tarefas**Atores:**

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores excluam as tarefas da empresa.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de tarefas nos resultados-chave da empresa.

Fluxo Principal:

1. O gestor seleciona a página “OKRs” na barra de navegação.
2. O gestor seleciona a expansão do objetivo em que a tarefa está inserida.
3. O gestor seleciona a expansão do resultado-chave em que a tarefa está inserida.
4. O gestor visualizará a tarefa que será excluída e seleciona a opção de exclusão.
5. Uma mensagem de confirmação de exclusão aparecerá.
6. O gestor seleciona o botão “Deletar”.
7. A tarefa estará excluída.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.

2a Desistência na exclusão da tarefa:

- O gestor seleciona o botão “Cancelar” e retorna a página de OKRs

Fonte: elaborado pelo autor (2023)

Tabela 33 - Caso de uso "Consultar chatBox"

Nome: Consultar chatBox

Atores:

- Atores Primários: Usuário gestor

Objetivo:

- Permitir que gestores consultem o chatBox para sanar dúvidas.

Pré-condições:

- O usuário está autenticado no sistema.
- O usuário possui permissões de gestor.

Pós-condições:

- As alterações feitas pelo gestor são refletidas na listagem de tarefas nos resultados-chave.

Fluxo Principal:

1. O gestor seleciona a página “analytics” na barra de navegação.
2. O gestor seleciona a expansão da caixa “Consultar Ia”.
3. O gestor escreve sua dúvida.
4. O sistema retorna a resposta de acordo com a pergunta.

Fluxos Alternativos:**1a Usuário não autenticado:**

- O sistema redireciona o usuário para a tela de login.