

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**DAVID JONAS DE OLIVEIRA
LUCAS ROSA CASSIANO**

SISTEMA DE GESTÃO DE TREINAMENTOS EMPRESARIAIS

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Fausto Gonçalves Cintra

FRANCA/SP

2023

SISTEMA DE GESTÃO DE TREINAMENTOS EMPRESARIAIS

David Jonas De Oliveira¹
Lucas Rosa Cassiano²
Fausto Gonçalves Cintra³

Resumo

A capacitação profissional dos funcionários de uma empresa é fundamental para que eles possuam os devidos conhecimentos e qualificações necessárias para receber e realizar tarefas com excelência. Alguns fatores como a falta de tempo para ministrar treinamentos e a dificuldade em reunir equipes presencialmente dificulta e inviabiliza tal prática, causando impactos negativos para uma organização. Este trabalho tem como objetivo oferecer uma solução sistêmica para padronizar os métodos de treinamento e que possa auxiliar as empresas no treinamento e capacitação profissional de seus colaboradores, proporcionando um método de treinamento à distância por meio de uma ferramenta de aprendizagem *online*, que possibilite o oferecimento e realização dos treinamentos de maneira prática e flexível. O projeto foi desenvolvido utilizando alguns métodos e ferramentas da Engenharia de *Software* como: entrevistas com *stakeholders*, Modelo Canvas, Matriz SWOT, 5W2H e BPMN. Estes métodos foram utilizados para o estudo de caso do problema em questão em uma empresa do ramo alimentício, e para a criação de toda documentação necessária. Os resultados obtidos foram satisfatórios, pois atendem às expectativas iniciais do projeto, um *software* de gestão de treinamentos viável que pode auxiliar a empresa na capacitação profissional de seus funcionários, solucionando o problema em questão.

Palavras-chave: aprendizagem *online*. capacitação profissional. educação corporativa. engenharia de *software*. qualificação profissional.

Abstract

The professional development of a company's employees is essential for them to have the necessary knowledge and qualifications to receive and carry out tasks with excellence. Factors such as a lack of time to provide training and the difficulty of gathering teams in person hinder and make such a practice unfeasible, causing negative impacts on an organization. This paper is to offer a systemic solution to help companies to standardize training methods and can help companies in the training and professionalize their employees, providing a method of distance training through an online learning tool that makes it possible to offer and carry out training in a practical and flexible way. The project developed using Software Engineering methods and tools, such as stakeholder interviews, Canvas Model, SWOT Matrix, 5W2H, and

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: davidjonasoliveira@gmail.com.

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: lucasrosac2012@gmail.com.

³ Docente em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: fausto.cintra@fatec.sp.gov.br.

BPMN. These methods used for the case study of the problem in a food industry company and for the creation of all necessary documentation. The results obtained were satisfactory as they met the initial project expectations, a viable training management software that can help the company in the professional development of its employees, solving the problem.

Keywords: *corporate education. online learning. professional development. professional qualification. software engineering.*

1 Introdução

O sistema educacional brasileiro apresenta metodologias ineficazes para o pleno desenvolvimento das pessoas e, conseqüentemente, para a capacitação profissional dos cidadãos. O baixo nível de capacitação profissional dos trabalhadores pode ser observado em diversos setores da economia, levando as empresas a implementar estratégias para a capacitação profissional interna de seus novos colaboradores.

A partir deste contexto, foi proposto como tema deste projeto entender as dificuldades enfrentadas pelas empresas na implementação de programas de capacitação profissional, propondo soluções mediante a utilização de ferramentas tecnológicas.

Por meio de entrevista presencial com *stakeholders* identificou-se que, na área de treinamento profissional dentro das empresas, é possível mitigar vários problemas pela implementação de ferramentas tecnológicas, entre elas, a dificuldade de reunir toda a equipe de forma presencial para realizar um treinamento, e a falta de padronização das metodologias de ensino, em especial quando o método utilizado consiste na transmissão de conhecimento entre um profissional mais experiente e um novo colaborador. Dessa forma, foi definida a seguinte questão problema para o projeto: como gerenciar e tornar mais prático e eficaz o processo de treinamento e capacitação dos funcionários de uma empresa, empregando recursos e ferramentas tecnológicas?

O presente documento estrutura as etapas do desenvolvimento do projeto em seis capítulos divididos conforme definido pela instituição. O projeto foi desenvolvido utilizando alguns métodos e ferramentas da Engenharia de *Software* como: entrevistas com *stakeholders*, Modelo Canvas, Matriz SWOT, 5W2H e BPMN.

No Capítulo 1 estão dispostos a introdução e o TAP. Na introdução foram apresentados o tema do projeto, a questão problema a ser estudada, os objetivos e

justificativas e as metodologias utilizadas. No TAP foram apresentados os objetivos do Termo de Abertura do Projeto, os objetivos e justificativas do projeto e detalhado os pontos levantados no TAP.

No Capítulo 2 foi apresentado o estudo de viabilidade do projeto por meio da análise do quadro Canvas, da matriz SWOT e da tabela 5W2H elaboradas para o projeto.

No Capítulo 3 foi realizado o levantamento de requisitos de interesse dos usuários e apresentado as tabelas e diagramas gerados para estruturar o *software* resultante do projeto.

No Capítulo 4 foram relacionadas as ferramentas e a metodologia utilizadas para o desenvolvimento da solução sistêmica.

No Capítulo 5 foi apresentado partes importantes do código fonte do *software* e as respectivas telas da interface de usuário.

Por fim, o Capítulo 6 apresenta uma análise dos resultados obtidos com o projeto e declaradas possibilidades de implementações futuras.

1.1 Termo da Abertura do Projeto (TAP)

O Termo de Abertura do Projeto (TAP) é um documento formal que agrupa as principais informações do projeto e descreve a empreitada com todos os seus pormenores. Ele é capaz de descrever desde as metas até seus problemas, além de alinhar toda a sua estrutura, ajudando as equipes a compreenderem todos os detalhes do projeto de maneira rápida e fácil (Proj4me, 2023).

Para Lourenço (2022, *online*):

O TAP tem, por propósito, ser o grande alvo a ser alcançado em relação aos resultados e benefícios do projeto, além de oficializar perante a instituição que o projeto existe e está aprovado investir e alocar recursos nele, deixando claro o propósito do que o projeto quer atingir em uma visão macro, além de fornecer a autoridade e limites ao gerente do projeto para a gestão do mesmo, e registrar quem são os *stakeholders* e o *roadmap* / entregas marcos iniciais previstas.

1.1.1 Nome do Projeto

Projeto de Sistema de Gestão de Treinamentos

1.1.2 Objetivo do Projeto

O objetivo proposto neste projeto é oferecer ao cliente uma solução sistêmica que auxilie a empresa no treinamento e capacitação de seus funcionários, visando proporcionar praticidade na disponibilização do conteúdo, flexibilidade na realização dos treinamentos e recursos de avaliação de aproveitamento, além da emissão de certificados digitais de conclusão.

1.1.3 Justificativa do Projeto

Os colaboradores são o coração de uma empresa, são eles que fazem tudo funcionar na prática. Para que esse funcionamento flua da melhor maneira possível, com mais facilidade e agilidade, é extremamente importante que eles tenham os devidos conhecimentos e qualificações necessárias para que estejam preparados para receber tarefas e executá-las com excelência. O treinamento e capacitação de funcionários é uma prática que permite o desenvolvimento contínuo dos colaboradores de uma empresa, impactando diretamente a qualidade e eficiência dos processos, a relação com clientes e a lucratividade. Ela permite aperfeiçoar habilidades técnicas dos colaboradores de uma empresa, trazendo novas metodologias a serem implementadas nas rotinas diárias e na forma de resolver problemas. Alguns obstáculos que as empresas enfrentam para realizar programas de treinamento eficazes são: a falta de tempo dos gerentes em ministrar treinamentos; impossibilidade em reunir toda equipe de forma presencial, por falta de flexibilidade de horários; dificuldade em avaliar o desempenho individual dos participantes. Para suprimir essas barreiras e dificuldades encontradas na realização de treinamentos empresariais, é preciso canalizar os esforços em uma proposta que englobe todas essas questões. A utilização de um método de treinamento à distância, através de ferramentas de aprendizagem *online*, é uma ótima alternativa que possibilita a utilização de diferentes tipos de mídias digitais para a disponibilização dos treinamentos. Com essa alternativa, o funcionário poderá adaptar a realização dos treinamentos de acordo com seus horários disponíveis. Outras vantagens são as avaliações aplicadas pela plataforma e o *feedback* de informações que ela fornece aos gestores.

1.1.4 Stakeholders

Diretor, gerente, representantes e outros colaboradores da empresa.

1.1.5 Premissas

Consideramos as seguintes premissas para o projeto:

- a) Utilização de ferramenta de gerenciamento de projeto (Trello) para a organização das tarefas.
- b) Todos os envolvidos trabalharão de forma colaborativa, visando sempre o sucesso do projeto.
- c) Todos os envolvidos cumprirão com os prazos e tarefas que lhe serão atribuídos.
- d) O projeto será executado de acordo com as diretrizes de Engenharia de *Software*.
- e) As comunicações entre a equipe do projeto, os *stakeholders* e os usuários finais serão definidas e gerenciadas.

1.1.6 Restrições

Consideramos as seguintes restrições para o projeto:

- a) O projeto deve atender a todos os requisitos do cliente.
- b) O cronograma do projeto deverá ser obrigatoriamente cumprido.
- c) O *software* deve ser executado em *hardware* específico ou cumprir requisitos de desempenho de *hardware*.

1.1.7 Riscos

Consideramos os seguintes riscos para o projeto:

- a) O projeto pode exceder o orçamento inicial, o que pode resultar em cortes de funcionalidades ou qualidade comprometida.
- b) Erros de codificação e problemas de qualidade podem resultar em atrasos na entrega e em custos adicionais para correções.
- c) O *software* pode não atender às metas de desempenho, resultando em lentidão ou instabilidade.
- d) Tecnologias subjacentes podem mudar durante o projeto, exigindo adaptações.
- e) Dependência de bibliotecas de terceiros, ferramentas ou serviços pode representar riscos se esses terceiros falharem.

1.1.8 Custos e Prazos Estimados do Projeto

- a) Prazo Estimado do Projeto: 3 meses.
- b) Custo Estimado do Projeto: R\$ 4.510,08.

2 Viabilidade do Projeto

“Viabilidade do projeto é considerar se determinado projeto é realizável ou não, analisando informações sobre seu desenvolvimento, implementação, possíveis resultados, impactos, prós e contras” (UCJ, 2023).

2.1 Canvas de Negócio (*Business Model Canvas* - BMC)

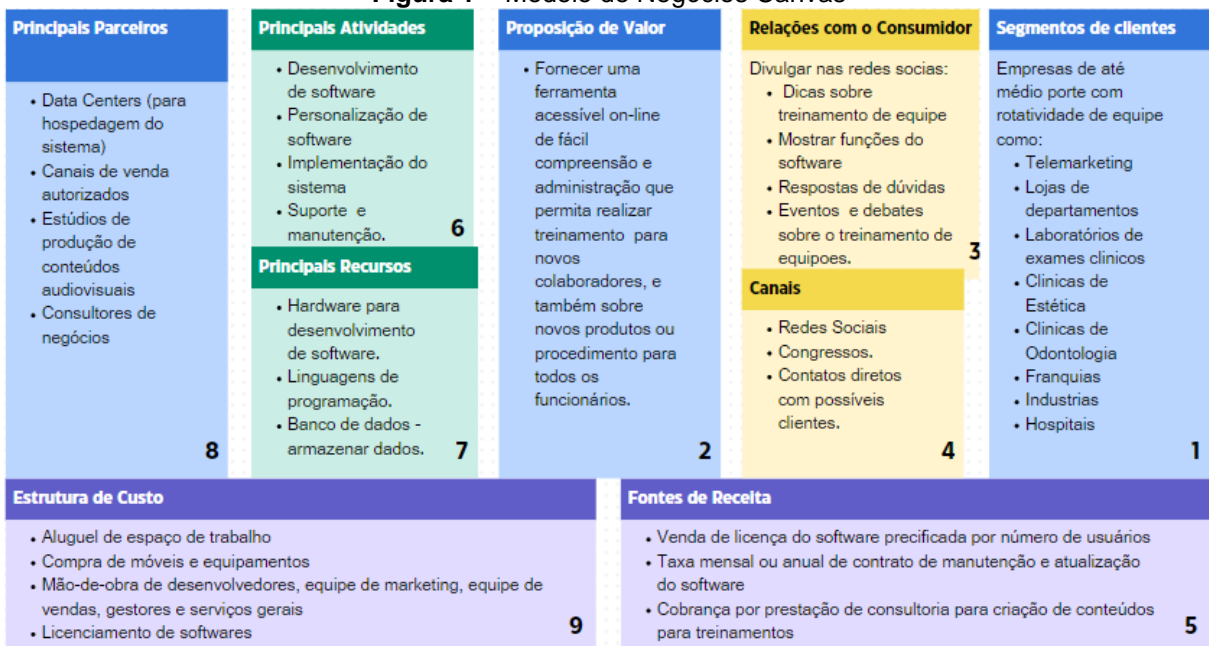
“O Canvas foi desenvolvido por Osterwalder e Pigneur em 2010, em seus livros ‘Business Model Canvas’, com o objetivo de sintetizar as funções da empresa em um único Quadro” (Cordeiro *et al.*, 2017, p. 6).

Visto que, Rabello (2023, *online*) descreveu o Canvas como uma ferramenta de gestão estratégica utilizada no desenvolvimento de modelos de negócios. Conta com nove blocos pré-formatados que servem como base para a criação ou adaptação de um modelo de negócio.

Ainda sobre o Canvas, Matias (2023) declara ser uma excelente opção para que as empresas disponibilizem a suas equipes uma visão clara sobre os projetos da empresa. Assim, os gestores podem se guiar pelos quadros para alcançar os objetivos da empresa de forma mais clara e ágil.

A Figura 1 mostra o modelo de negócios canvas do projeto de *software*.

Figura 1 – Modelo de Negócios Canvas



Fonte: elaborado pelos autores.

Ainda segundo Matias (2023), cada um destes blocos do quadro Canvas detalha um aspecto específico do projeto, conforme apresentado brevemente a seguir:

- a) **Segmento de Clientes.** Neste primeiro bloco, é definido quais são os públicos que a empresa pretende atender. Essa definição é importante para os próximos passos, que vão estabelecer estratégias de comunicação, distribuição e desenvolvimento de produtos.
- b) **Proposta de Valor.** Definição do que a empresa vai entregar para os seus clientes, que vai muito além de definir quais produtos serão produzidos. Deve-se determinar quais problemas os produtos vão ajudar a resolver, e quais necessidades dos clientes precisam ser atendidas.
- c) **Canais.** No bloco de canais, são listados quais canais serão utilizados para fazer com que os produtos sejam entregues aos clientes. Aqui podem entrar elementos como distribuidores, loja física, *e-commerce*, revendedores, atacadistas, entre outros.
- d) **Relacionamento.** Bloco no qual são definidas quais estratégias serão adotadas pela empresa para conquistar e reter clientes. Para isso, é importante entender quais são as preferências dos seus públicos para que consiga ter uma comunicação mais eficaz.

- e) **Fontes de Receita.** Neste bloco, define-se como será a entrada de dinheiro na empresa, ou seja, quanto os clientes pagarão por cada produto ou serviço. Além dos custos de produção e de operação, deve-se considerar o poder aquisitivo dos seus públicos, os preços praticados pelos seus principais concorrentes bem como a média do mercado.
- f) **Recursos Chave.** Deverão ser listados os recursos necessários para o funcionamento da empresa. Abrangendo não somente os recursos financeiros, mas também de recursos humanos, intelectuais e físicos.
- g) **Atividades Chave.** Neste bloco, serão listadas todas as ações mais importantes para que o negócio funcione. Engloba tanto as atividades relacionadas diretamente com a produção como as tarefas administrativas.
- h) **Parcerias Chave.** Consideram-se os principais parceiros do seu negócio as empresas que vão ajudar no funcionamento, como fornecedores e outras empresas prestadoras de serviço.
- i) **Estrutura de Custos.** A estrutura de custos é um dos blocos mais relevantes do canvas. Aqui serão registrados todos os custos necessários para que todos os outros planos se concretizem. Ou seja, deve-se determinar os custos da operação, tanto diretos como indiretos.

2.2 Matriz SWOT

A Figura 2 exibe a matriz SWOT desenvolvida no projeto. Para Pedra (2023, *online*) a matriz SWOT é uma ferramenta muito utilizada em planejamentos estratégicos:

A matriz SWOT (ou matriz FOFA em português) é uma ferramenta que pode ser utilizada no planejamento estratégico ou também nos planos de negócios. Sua finalidade é avaliar os ambientes internos e externos de uma organização para apoiar na tomada de decisões.

Na prática, análise SWOT é a sigla em inglês para:

Forças (*Strengths*).

Oportunidades (*Opportunities*).

Fraquezas (*Weakness*).

Ameaças (*Threats*).

O conceito foi desenvolvido pelos professores Kenneth Andrews e Roland Christensen por volta dos anos 60. Seu objetivo sempre foi aprimorar o planejamento estratégico das empresas, com foco nas tomadas de decisão.

Figura 2 – Matriz SWOT

	AJUDA	ATRAPALHA
INTERNA (organização)	<p>FORÇAS</p> <ul style="list-style-type: none"> • Bom relacionamento com os clientes • Exclusividade de produtos • Logística eficiente • Disponibilidade e diversidade de produtos • Qualidade dos produtos 	<p>FRAQUEZAS</p> <ul style="list-style-type: none"> • Preço do produto maior que o da concorrência • Falta de capacitação dos representantes • Alto índice de clientes inadimplentes • Falta de conhecimento dos clientes sobre a utilização dos produtos
EXTERNA (ambiente)	<p>OPORTUNIDADES</p> <ul style="list-style-type: none"> • Expandir a área de atuação • Aumentar a variedade de produtos • Investir em suporte especializado ao cliente • Fidelizar clientes • Investir em novas tecnologias de logística 	<p>AMEAÇAS</p> <ul style="list-style-type: none"> • Aumento da concorrência • Aumento do dólar • Preço baixo dos concorrentes • Guerras em países produtores • Crise em países fornecedores

Fonte: elaborado pelos autores.

Entre as fraquezas identificadas durante o processo de entrevista com o cliente, duas refletem condições financeiras de mercado e estão fora do escopo proposto neste projeto. As fraquezas relacionadas a capacitação dos representantes de vendas e falta de conhecimento dos clientes sobre os produtos foram discutidas como possíveis problemas a serem resolvidos. Após uma análise feita pela equipe em conjunto com o cliente foi definido que os esforços seriam voltados para o desenvolvimento de um sistema para mitigar a falta de capacitação dos representantes de vendas, pois conclui-se que desta forma os colaboradores poderiam informar melhor os clientes sobre os produtos durante as negociações.

2.3 Plano de Ação 5W2H do Projeto

Para Sebrae-SC (2022), 5W2H é um *checklist* de ações preventivas e corretivas a serem realizadas por uma empresa de forma organizada, prática, eficiente e clara. Envolve líderes e colaboradores para identificar as necessidades e propor soluções aos objetivos da empresa.

Ainda segundo Sebrae-SC (2022), o 5W2H é a representação de um plano de ação que tem como objetivo definir os desdobramentos da estratégia em ações com monitoramento. Por meio dele, a empresa deve planejar tudo o que deve ser executado, criar uma metodologia, criar um cronograma e indicar os responsáveis para dar andamento a cada etapa. A Figura 3 demonstra a planilha 5W2H desenvolvida para o projeto.

Figura 3 - Planilha 5W2H

Gestão Participativa: 5W2H

Questão problema: Conscientizar os representantes (vendedores) da empresa.

Ação	O que? (What)?	Por quê? (Why)?	Onde? (Where)?	Quem? (Who)?	Quando? (When)?	Como? (How)	Quanto? (How Much)?
1	Criar uma plataforma de treinamentos e processos (não só dos clientes da empresa também).	Capacitar os vendedores a oferecer o melhor produto de acordo com o procedimento e necessidade do cliente.	Plataforma online (sistema web).	Desenvolvedores.	Média de 2 meses para implantação.	-Contratar empresa de software para implantar o sistema.	R\$ 8.000,00
2	Especialistas na área dar treinamentos na plataforma.	Melhorias de qualidade nas vendas.	Setor de vendas da empresa.	Vendedores especializados da empresa.	Média de 1 mês	-Especialistas gravar e postar vídeos na plataforma	Sem custos.
3	Disponibilizar notebooks aos vendedores.	Objeto de estudo para que os vendedores assistem os vídeos na plataforma.	Poderá assistir o treinamento em casa ou presencial.	Empresa	Assim que os especialistas acabarem de gravar os treinamentos (Média de 1 mês).	-Empresa irá adquirir mais notebooks.	R\$ 30.000,00
4	Vendedores utilizarem a plataforma	Capacitação nas vendas (ferramenta de	Plataforma online (sistema web)	Vendedores.	1 semana	-Assistir vídeos postados pelos especialistas	Sem custos.

Fonte: elaborado pelos autores.

Após uma análise realizada em conjunto com o cliente, definiu-se como prioridade a tratativa da falta de capacitação de seus vendedores, pois frequentemente aconteciam vendas de insumos inadequados para o tipo de produto manufaturado pelos clientes. Assim, havia um descontentamento por parte dos clientes e perdas financeiras de ambos os lados. No intuito de sanar este problema foi definido um plano de ação para mitigar tal fragilidade da empresa.

De acordo com a planilha a ação número 1 foi definida como escopo para o projeto, segue a descrição de cada campo do plano de ação definido no projeto:

- a) O que será feito? (*What?*) Criar uma plataforma de treinamentos *online*.
- b) Por que será feito? (*Why*) Capacitar os vendedores a oferecer o melhor produto de acordo com o procedimento e necessidade do cliente.

- c) Onde será feito? (*Where?*) Plataforma *online* (sistema web).
- d) Quando será feito? (*When?*) Desenvolvedores.
- e) Por quem será feito? (*Who?*) Média de 2 meses para implementação;
- f) Como será feito? (*How?*) Contratar empresa de desenvolvimento de *software* para implantar o sistema.
- g) Quanto vai custar? (*How Much?*) R\$ 4.510,08 (quatro mil, quinhentos e dez reais e oito centavos) para a implantação além de um contrato de manutenção com mensalidade no valor de R\$ 150,00 (cento e cinquenta reais), que incluirá atualizações de versão, suporte técnico limitado a assinatura para configurações e solução de problemas, acesso a um treinamento *online* sobre o sistema com vídeos, tutoriais, dicas e repostas de perguntas mais frequentes.

3 Levantamento de Requisitos

3.1 Elicitação e especificação dos Requisitos

A elicitação de requisitos é a fase do projeto que ajuda o analista a entender as reais necessidades do cliente, extraindo informações importantes sobre o que ele deseja que seja construído. Na elicitação de requisitos são percebidos os recursos e as características que esse sistema deve ter. Em muitos casos, o cliente não tem um entendimento claro sobre as suas necessidades e não consegue passar as informações de uma forma compreensível. É nesse momento que o profissional de TI utiliza técnicas de elicitação de requisitos para entender o que exatamente o cliente está precisando, buscando informações como funcionalidades que o sistema deve ter, regras de negócio, restrições, usabilidade do *software*, e assim por diante (Devmedia, 2014).

Por meio de diversas visitas realizadas na empresa para o levantamento dos requisitos, foram aplicadas as técnicas de conversação e entrevista com o gerente da empresa, onde foram levantadas as seguintes especificações:

Eu, enquanto cliente
Desejo criar treinamentos *online*
Para capacitar os colaboradores
Prioridade: Essencial

Eu, enquanto cliente
Desejo inserir arquivos de texto, imagens e vídeos nos treinamentos
Para transmitir conhecimentos da melhor forma possível
Prioridade: Importante

Eu, enquanto cliente
Desejo criar e aplicar avaliações sobre os treinamentos
Para avaliar o aproveitamento do conteúdo pelo funcionário
Prioridade: Essencial

Eu, enquanto cliente
Desejo enviar notificações sobre treinamentos pendentes
Para lembrar os funcionários de realizarem os treinamentos
Prioridade: Importante

Eu, enquanto cliente
Desejo inscrever funcionários nos treinamentos
Para permitir o acesso dos funcionários aos treinamentos
Prioridade: Essencial

Eu, enquanto cliente
Desejo cadastrar novos usuário no sistema
Para liberar o acesso de novos alunos/funcionários
Prioridade: Essencial

Eu, enquanto cliente
Desejo disponibilizar acesso aos treinamentos pelos usuários do sistema
Para permitir acesso do funcionário ao conteúdo e gravar seu progresso
Prioridade: Essencial

Eu, enquanto cliente
Desejo disponibilizar um campo para os funcionários registrarem suas dúvidas
Para que as dúvidas possam ser esclarecidas
Prioridade: Importante

Eu, enquanto cliente
Desejo disponibilizar recursos para esclarecimento de dúvidas sobre o treinamento
Para que as dúvidas de usuários sobre treinamentos possam ser esclarecidas
Prioridade: Importante

Eu, enquanto cliente
Desejo emitir certificados de conclusão para os funcionários que atingirem os requisitos
Para fornecer certificação e manter histórico sobre a capacitação dos funcionários
Prioridade: Importante

3.2 BPMN

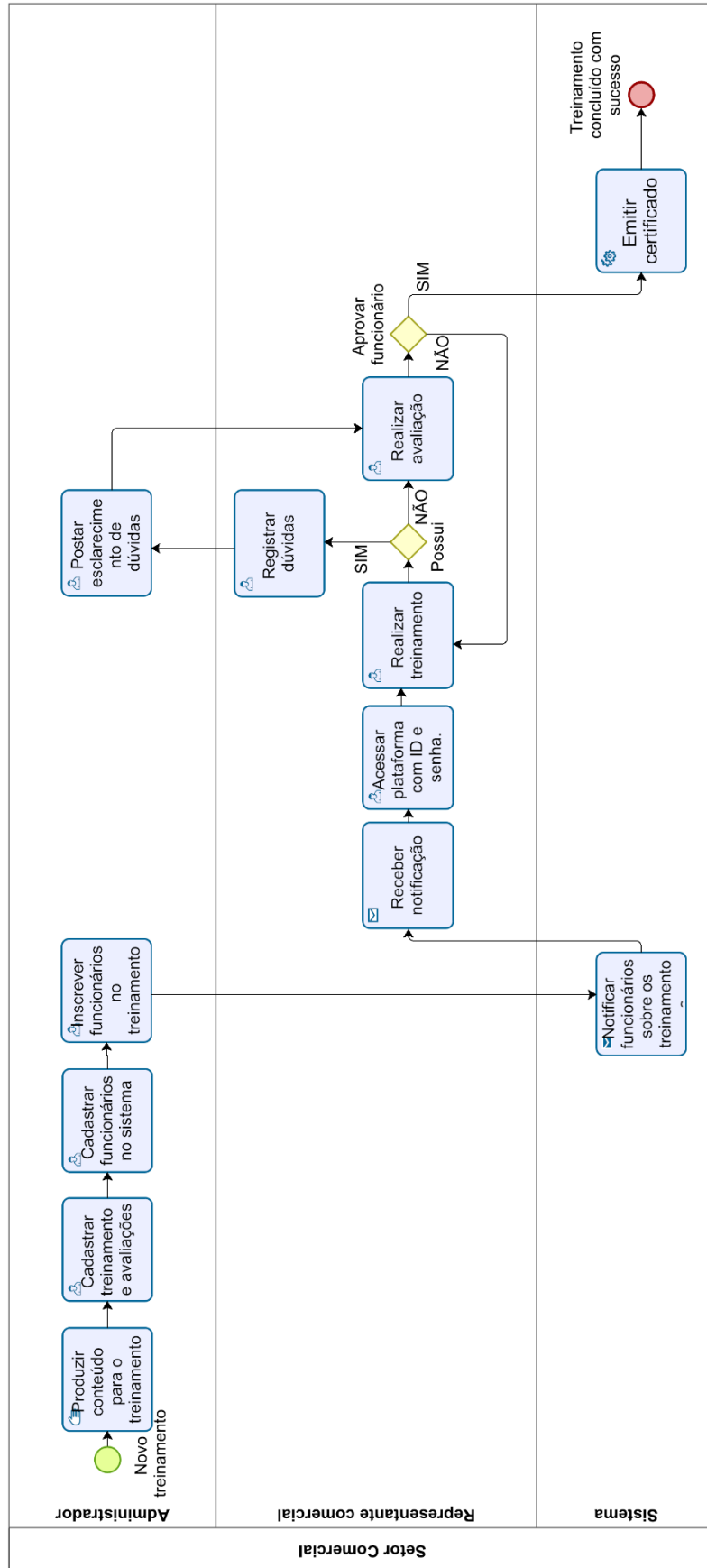
Segundo Silver (2019, p. 5), “BPMN, que significa *Business Process Model and Notation*, é uma linguagem de diagramação para modelos de processos de negócios.”

O principal objetivo da BPMN é oferecer uma notação de fácil entendimento por todos os usuários, desde os analistas que criam os primeiros rascunhos do projeto, os desenvolvedores que fazem a implementação da tecnologia que irá executar estes processos e as pessoas que irão administrar e monitorar os processos (OMG, 2011).

A utilização do BPMN ajuda a melhorar a eficiência do projeto, otimizando os resultados e reduzindo o consumo de recursos. Ela é uma metodologia importante para promover maior transparência estratégica e estrutural, possibilitando uma visão compreensiva de como as coisas funcionam dentro de uma organização.

A Figura 4 mostra o diagrama BPMN desenvolvido para o projeto.

Figura 4 – Diagrama BPMN



Fonte: elaborado pelos autores.

3.3 Requisitos Funcionais

“É a definição das funcionalidades que um sistema de *software* deve fornecer. Esse tipo de requisito informa como o sistema deve trabalhar os dados de entrada e quais informações devem ser geradas na saída” (Pierri, 2019, *online*).

Ainda segundo Pierri (2019), os requisitos funcionais servem para definir como o sistema irá se comportar em situações específicas. Eles também definem o escopo do sistema de *software* especificando claramente quais serão suas funcionalidades e o que deve ou não deve fazer.

O Quadro 1 demonstra os requisitos funcionais levantados para o projeto.

Quadro 1 – Requisitos Funcionais do sistema

ID: RF001	Nome do Requisito: Cadastrar treinamentos e avaliações
Descrição:	O sistema deverá permitir o cadastro de novos treinamentos e suas avaliações conforme a necessidade da empresa.
Categoria: Evidente	Prioridade: Essencial
Informações:	ID do treinamento, ID da avaliação, nome do treinamento, arquivos de digitais de conteúdo para os treinamentos (texto, imagens, áudio e vídeo), e formulários para as avaliações.
Regra de Negócio:	O sistema deverá impedir a criação de treinamento com mesmo nome e/ou ID. O sistema deverá permitir o gerenciamento de níveis dos acessos aos treinamentos. O sistema deverá permitir a edição dos treinamentos cadastrados (inserção e exclusão de arquivos e formulários). O sistema deverá permitir a exclusão dos treinamentos.
ID: RF002	Nome do Requisito: Cadastrar os funcionários na plataforma.
Descrição:	O sistema deverá permitir cadastrar os funcionários e seus níveis de acesso aos treinamentos.
Categoria: Evidente	Prioridade: Essencial
Informações:	ID de usuário, nome do usuário, nível de acesso ao treinamento, CPF do usuário, e-mail do usuário, número do telefone do usuário.
Regra de Negócio:	O sistema deverá permitir o cadastro de apenas um usuário por CPF. O sistema deverá permitir alterações nos usuários.
ID: RF003	Nome do Requisito: Inscrever funcionários nos treinamentos
Descrição:	O sistema deverá permitir a inscrição de usuários nos treinamentos.
Categoria: Evidente	Prioridade: Essencial
Informações:	ID do funcionário, ID do treinamento, nome do usuário, nível de acesso ao treinamento, CPF do usuário, e-mail do usuário, número do telefone do usuário.
Regra de Negócio:	O sistema deverá permitir a inscrição somente de usuários ativos nos treinamentos.
ID: RF004	Nome do Requisito: Notificar os funcionários sobre o treinamento
Descrição:	O sistema deverá enviar uma notificação para o funcionário via e-mail sempre que ele for inscrito em um treinamento.

Categoria: Oculto	Prioridade: Importante
Informações:	Nome do usuário, nome do treinamento, <i>link</i> para acesso ao treinamento, usuário e senha para acesso ao treinamento, mensagem de aviso de treinamento disponível.
Regra de Negócio:	O sistema deverá enviar e-mail de notificação a todos os usuários inseridos que forem cadastrados em novos treinamentos.
ID: RF005	Nome do Requisito: Realizar treinamento
Descrição:	O sistema deverá permitir o acesso dos usuários ao conteúdo dos treinamentos em que este estiver cadastrado e gravar seu progresso.
Categoria: Evidente	Prioridade: Essencial
Informações:	Usuário de sistema, senha, conteúdo do treinamento.
Regra de Negócio:	O sistema deverá permitir o acesso apenas de usuários cadastrados no treinamento
ID: RF006	Nome do Requisito: Postar esclarecimentos de dúvidas.
Descrição:	O sistema deverá permitir a postagem de esclarecimentos de dúvidas enviadas pelos usuários.
Categoria: Evidente	Prioridade: Importante
Informações:	ID do funcionário que enviou a dúvida, descrição da dúvida.
Regra de Negócio:	Permitir o envio de dúvidas apenas de usuários que estiverem iniciado o treinamento.
ID: RF007	Nome do Requisito: Registrar dúvidas
Descrição:	O sistema deverá apresentar um campo para inserção de dúvidas do usuário que será enviada ao gerente com um botão para registrar a dúvida.
Categoria: Oculto	Prioridade: Importante
Informações:	Nome do usuário, CPF do usuário, nome do treinamento, texto-pergunta.
Regra de Negócio:	
ID: RF008	Nome do Requisito: Realizar avaliação
Descrição:	O sistema deverá permitir a realização de avaliação através de preenchimento de formulários.
Categoria: Evidente	Prioridade: Essencial
Informações:	Id do funcionário, ID do treinamento, ID da avaliação, número da tentativa, questões da avaliação, respostas, nota, situação.
Regra de Negócio:	O sistema deverá permitir a realização da avaliação apenas dos usuários que concluírem todas as etapas do treinamento.
ID: RF009	Nome do Requisito: Emitir certificado
Descrição:	O sistema deverá emitir o certificado de conclusão de treinamento
Categoria: Evidente	Prioridade: Importante
Informações:	Nome do usuário, CPF do usuário, nome do treinamento, carga horária do treinamento, nota de aproveitamento obtida.
Regra de Negócio:	O sistema deverá emitir certificado apenas para usuários com nota de aproveitamento superior a 70%
ID: RF010	Nome do Requisito: Acessar a plataforma com ID e senha.
Descrição:	O sistema deverá permitir o acesso a plataforma mediante inserção de id e senha válidos
Categoria: Evidente	Prioridade: Importante
Informações:	ID e senha do usuário.
Regra de Negócio:	O sistema deverá restringir o acesso a plataforma apenas com ID e senha válidos.
ID: RF011	Nome do Requisito: Inserir arquivos de texto, imagens e vídeos nos treinamentos
Descrição:	O sistema deverá permitir a inserção de arquivos de texto, imagens e vídeos no conteúdo dos treinamentos.

Categoria: Evidente	Prioridade: Importante
Informações:	Nome do arquivo, tipo do arquivo, data de atualização

Fonte: elaborado pelos autores.

3.4 Casos de Uso

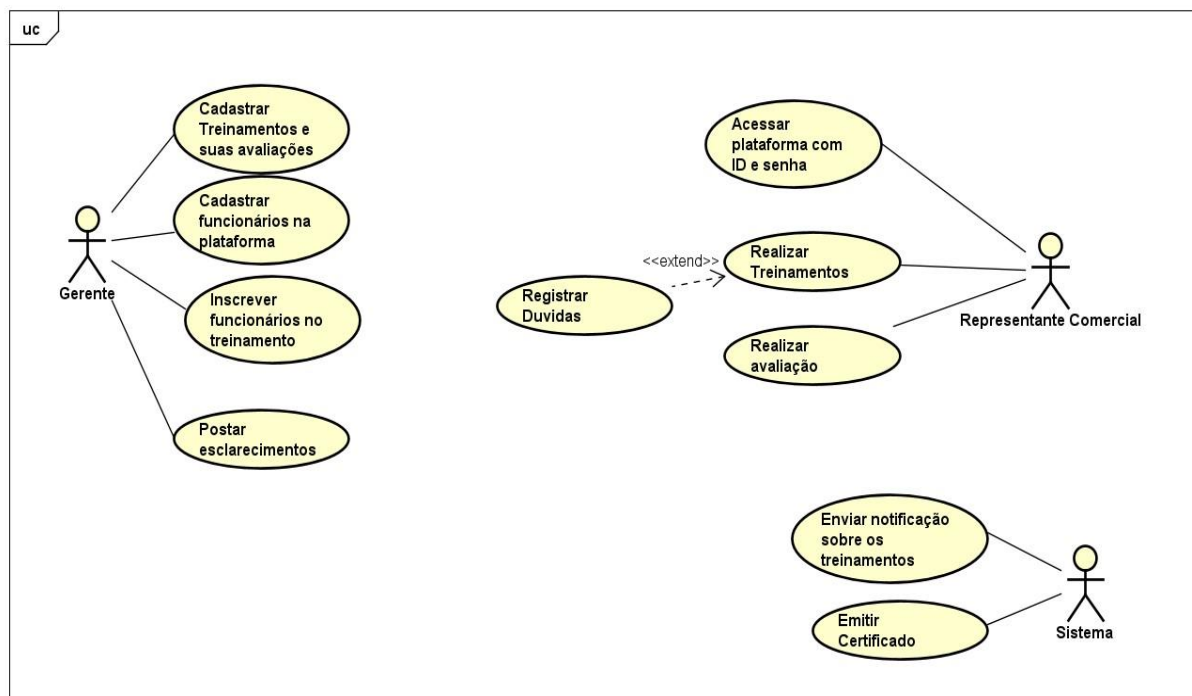
Um caso de uso descreve as interações entre os atores e o sistema na forma de um diálogo entre eles. Segundo a IBM (2021):

Um caso de uso é um artefato que define uma sequência de ações que produzem um resultado observável de valor. Os casos de uso fornecem uma estrutura para expressar requisitos funcionais no contexto de processos de negócios e de sistema. Os casos de uso podem ser representados como um elemento gráfico em um diagrama e como uma especificação de caso de uso em um documento textual.

Um caso de uso de negócios define uma sequência de ações executadas por um negócio que produzem um resultado de valor observável (uma saída de trabalho) para um agente de negócios específico ou que mostra como o negócio responde a um evento de negócio.

A ferramenta utilizada para a criação dos diagramas foi o *software* ASTAH UML versão 8.5.0/39c620 licença para estudantes, ambiente Windows desenvolvido pela Change Vision, Inc. A Figura 5 mostra o digrama de caso de uso relacionados nos Quadros 2 a 11.

Figura 5 – Diagrama de Caso de uso



Fonte: elaborado pelos autores.

Quadro 2 – Use Case Cadastrar Funcionário

Caso de Uso – Cadastrar Funcionário na plataforma	
ID	UC 002
Descrição	Este caso de uso tem por objetivo cadastrar um novo funcionário na plataforma atendendo o RF 002.
Ator Primário	Gerente do sistema
Pré-condição	Ter perfil de gerente no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o gerente seleciona o campo de cadastrar um novo usuário “funcionário”. 2. O sistema carrega o formulário para cadastrar o usuário “funcionário”. 3. O gerente informa os dados do usuário “funcionário”, como: Nome do perfil, nome do usuário, nível de acesso ao treinamento, CPF do usuário, e-mail do usuário, número do telefone do usuário. 4. O sistema consulta o banco de dados e verifica. 5. O sistema valida o CPF e nome de perfil do usuário. 6. O sistema grava o nome do gerente que cadastrou o usuário. 7. O sistema grava a data de inclusão. 8. O sistema gera o ID do usuário e salva no banco de dados. 9. Uma mensagem de confirmação do cadastro é mostrada.
Pós-condição	Não possui.
Cenário Alternativo	<p>a - A qualquer momento o gerente pode sair do sistema.</p> <p>6.a - Nome de perfil ou CPF já se encontra cadastrado.</p>

Fonte: elaborado pelos autores.

Quadro 3 – Use Case Cadastrar Treinamentos e Avaliações

Caso de Uso – Cadastrar Treinamentos e avaliações	
ID	UC 001
Descrição	Este caso de uso tem por objetivo cadastrar um novo treinamento no sistema atendendo o RF 001.
Ator Primário	Gerente
Pré-condição	Ter perfil de gerente no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o gerente seleciona o campo de cadastrar um novo treinamento. 2. O sistema carrega o formulário para cadastrar treinamento. (Arquivos: Imagens, Áudios, Vídeos, formulários e textos). 3. O gerente informa o nome do treinamento e anexa o arquivo de treinamento. 4. O sistema consulta o banco de dados e verifica. 5. O sistema valida o nome e ID do treinamento. 6. O sistema grava o nome do gerente que cadastrou o treinamento. 7. O sistema grava a data de inclusão. 8. O sistema gera o ID do treinamento e salva no banco de dados. 9. Uma mensagem de confirmação do cadastro é mostrada.
Pós-condição	Não possui.
Cenário Alternativo	<p>a - A qualquer momento o gerente pode sair do sistema.</p> <p>6.a - Nome e/ou ID já se encontra cadastrado.</p>

Fonte: elaborado pelos autores.

Quadro 4 – Use Case Inscrever Funcionários no Treinamento

Caso de Uso – Inscrever funcionários no treinamento	
ID	UC 003
Descrição	Este caso de uso tem por objetivo cadastrar um funcionário a um treinamento no sistema atendendo o RF 003.
Ator Primário	Gerente
Pré-condição	Ter perfil de gerente no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o gerente seleciona o campo de cadastrar funcionário a um novo treinamento. 2. O sistema carrega o formulário para cadastrar o funcionário. 3. O gerente informa o nome ou ID do treinamento em que deseja cadastrar o funcionário. 4. O sistema consulta o banco de dados e verifica. 5. O sistema valida o nome e ID do treinamento. 6. Gerente informa o ID do funcionário. 7. O sistema consulta o banco de dados e verifica. 8. O sistema valida ID do funcionário. 9. Sistema preenche automaticamente os dados do funcionário, como: Nome do perfil, nome do usuário, nível de acesso ao treinamento, CPF do usuário, e-mail do usuário, número do telefone do usuário. 10. O Gerente seleciona a opção de enviar e-mail para o funcionário para deixá-lo ciente sobre o treinamento em que foi incluído. 11. O sistema grava o nome do gerente que cadastrou o funcionário. 12. O sistema grava a data de inclusão. 13. O sistema grava o ID do funcionário cadastrado no treinamento e salva no banco de dados. 14. Uma mensagem de confirmação do cadastro é mostrada.
Pós-condição	Não possui.
Cenário Alternativo	<p>a, - Em qualquer momento o gerente pode sair do sistema.</p> <p>7.a - CPF ou ID já se encontra cadastrado.</p>

Fonte: elaborado pelos autores.

Quadro 5 – Use Case Enviar notificação sobre os treinamentos

Caso de Uso - Enviar notificação sobre os treinamentos	
ID	UC 004
Descrição	Este caso de uso tem objetivo de notificar via e-mail os funcionários que foram incluídos a treinamentos pelo gerente no sistema, atendendo o RF 004.
Ator Primário	Sistema
Pré-condição	Gerente inscrever um novo funcionário em um treinamento.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o gerente seleciona a opção de enviar e-mail assim que o funcionário for cadastrado no formulário do UC 002. 2. O sistema envia o e-mail com o <i>link</i> do treinamento. 3. Funcionário acessa o <i>link</i> informando seu <i>login</i> e senha.
Pós-condição	Não possui.
Cenário Alternativo	2.a - O e-mail não foi encontrado.

Fonte: elaborado pelos autores.

Quadro 6 – Use Case Postar Esclarecimentos

Caso de Uso – Postar Esclarecimentos	
ID	UC 005
Descrição	Este caso de uso tem o objetivo de esclarecer as dúvidas que o funcionário tiver sobre o treinamento, atendendo o RF007.
Ator Primário	Gerente
Pré-condição	Ter perfil de gerente no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o gerente seleciona o campo dúvidas de treinamentos. 2. O sistema carrega um chat com as dúvidas postadas pelos funcionários. 3. O Sistema mostra todas as dúvidas que ainda não foram respondidas. 4. Sistema mostra no chat o nome do funcionário, descrição da dúvida, data e horário que a dúvida foi registrada e de qual treinamento se trata. 5. O sistema possibilita o gerente enviar mensagens pelo chat. 6. O gerente posta o esclarecimento da dúvida no chat respondendo o funcionário. 7. Uma mensagem de resposta postada com sucesso é mostrada.
Pós-condição	Não possui.
Cenário Alternativo	2.a - Nenhuma dúvida registrada.

Fonte: elaborado pelos autores.

Quadro 7 – Use Case Realizar Treinamento

Caso de Uso – Realizar treinamento	
ID	UC 006
Descrição	Este caso de uso tem objetivo de disponibilizar o conteúdo do treinamento para o usuário atendendo o RF005.
Ator Primário	Funcionário
Pré-condição	Ter perfil de funcionário na plataforma.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário “funcionário” seleciona o campo de realizar treinamento. 2. A plataforma carrega uma página para cliente informar seus dados. 3. O usuário informa seus dados, como: ID e e-mail. 4. O sistema consulta o banco de dados e verifica. 5. Sistema valida os dados informados. 6. Sistema mostra todos os treinamentos em que o usuário está incluído. 7. O usuário seleciona o treinamento que deseja realizar. 8. O sistema abre o arquivo anexado para aquele treinamento. 9. O usuário realiza o treinamento. 10. O sistema mostra a porcentagem da conclusão do treinamento. 11. Sistema grava a porcentagem da conclusão no banco de dados.
Pós-condição	Não possui.
Cenário Alternativo	5.a- ID ou e-mail não encontrados. 8.a - Treinamento já realizado.

Fonte: elaborado pelos autores.

Quadro 8 – Use Case Registrar Dúvidas

Caso de Uso – Registrar dúvidas	
ID	UC 007

Descrição	Este caso de uso tem objetivo de disponibilizar ao usuário um campo para que ele possa registrar dúvidas sobre o treinamento no sistema, atendendo o RF006.
Ator Primário	Funcionário
Pré-condição	Ter perfil de funcionário na plataforma.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário “funcionário” seleciona o campo de registrar dúvidas. 2. A plataforma carrega uma página para cliente informar seus dados e do treinamento. 3. O usuário informa os dados, como: ID do usuário e ID de treinamento. 4. O sistema consulta o banco de dados e verifica. 5. Sistema valida os dados informados. 6. Sistema carrega um chat de bate-papo. 7. O sistema possibilita que o usuário envie mensagens pelo chat. 8. O usuário registra sua dúvida. 9. Uma mensagem de dúvida registrada com sucesso é mostrada. 10. O sistema grava os dados do usuário, a data e horário em que a dúvida foi gravada no banco de dados.
Pós-condição	Não possui.
Cenário Alternativo	5.a - ID do usuário ou ID do treinamento não encontrados. 5.b - ID do usuário informado não está incluso em um treinamento
Inclusão	Não possui
Extensão	UC 006 - Realizar treinamento

Fonte: elaborado pelos autores.

Quadro 9 – Use Case Realizar Avaliação

Caso de Uso – Realizar Avaliação	
ID	UC 008
Descrição	Este caso de uso tem como objetivo disponibilizar ao usuário um formulário avaliativo sobre o treinamento no sistema, atendendo o RF008.
Ator Primário	Funcionário
Pré-condição	Ter perfil de funcionário na plataforma.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o usuário “funcionário” seleciona o campo de realizar avaliação. 2. A plataforma carrega uma página para o cliente informar seus dados e os dados do treinamento. 3. O usuário informa os dados, como: ID do usuário e ID de treinamento. 4. O sistema consulta o banco de dados e verifica. 5. Sistema valida os dados informados. 6. Sistema carrega o formulário da avaliação para usuário começar a avaliação. 7. O sistema estabelece um tempo para finalização da avaliação. 8. O usuário finaliza a avaliação respondendo todas as alternativas do formulário. 9. Sistema mostra no final o total de acertos das questões. 10. Sistema grava os dados da avaliação no banco de dados. 11. Uma mensagem de avaliação efetuada com sucesso é mostrada.
Pós-condição	Não possui
Cenário Alternativo	5.a - ID do usuário ou ID do treinamento não encontrados. 5.b - Prova já realizada 8.a - Não foram respondidas todas as questões.

Fonte: elaborado pelos autores.

Quadro 10 – Use Case Emitir Certificado

Caso de Uso – Emitir Certificado	
ID	UC 009
Descrição	Este caso de uso tem objetivo de emitir certificado para aqueles que foram aprovados na avaliação do treinamento atendendo RF009.
Ator Primário	Sistema
Pré-condição	Ter concluído e ser aprovado no treinamento.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o funcionário seleciona a opção de finalizar a avaliação. 2. O Sistema verifica se a porcentagem de acerto da avaliação foi de 70% ou mais. 3. O Sistema emite o certificado automaticamente e apresenta uma opção para o usuário fazer download do certificado em arquivo formato PDF. 4. O usuário seleciona a opção faz o download do arquivo. 5. Uma mensagem de certificado baixado com sucesso é mostrada.
Pós-condição	Não possui
Cenário Alternativo	3.a - A porcentagem de acerto está abaixo de 70%, não foi possível emitir o certificado.

Fonte: elaborado pelos autores.

Quadro 11 – Use Case Acessar Plataforma com ID e Senha

Caso de Uso – Acessar Plataforma com ID e senha	
ID	UC 010
Descrição	Este caso de uso tem objetivo permitir o acesso a plataforma para realização dos treinamentos, atendendo RF010.
Ator Primário	Funcionário
Pré-condição	Estar cadastrado na plataforma.
Cenário Principal	<ol style="list-style-type: none"> 1. O <i>use case</i> inicia quando o funcionário seleciona a opção de acessar a plataforma. 2. O Sistema verifica se o ID e senha inseridos são validos no banco de dados. 3. O Sistema permite o acesso caso o ID e senha sejam válidos.
Pós-condição	Não possui
Cenário Alternativo	3.a - Caso o ID e/ou senha sejam inválidos o sistema não libera o acesso

Fonte: elaborado pelos autores.

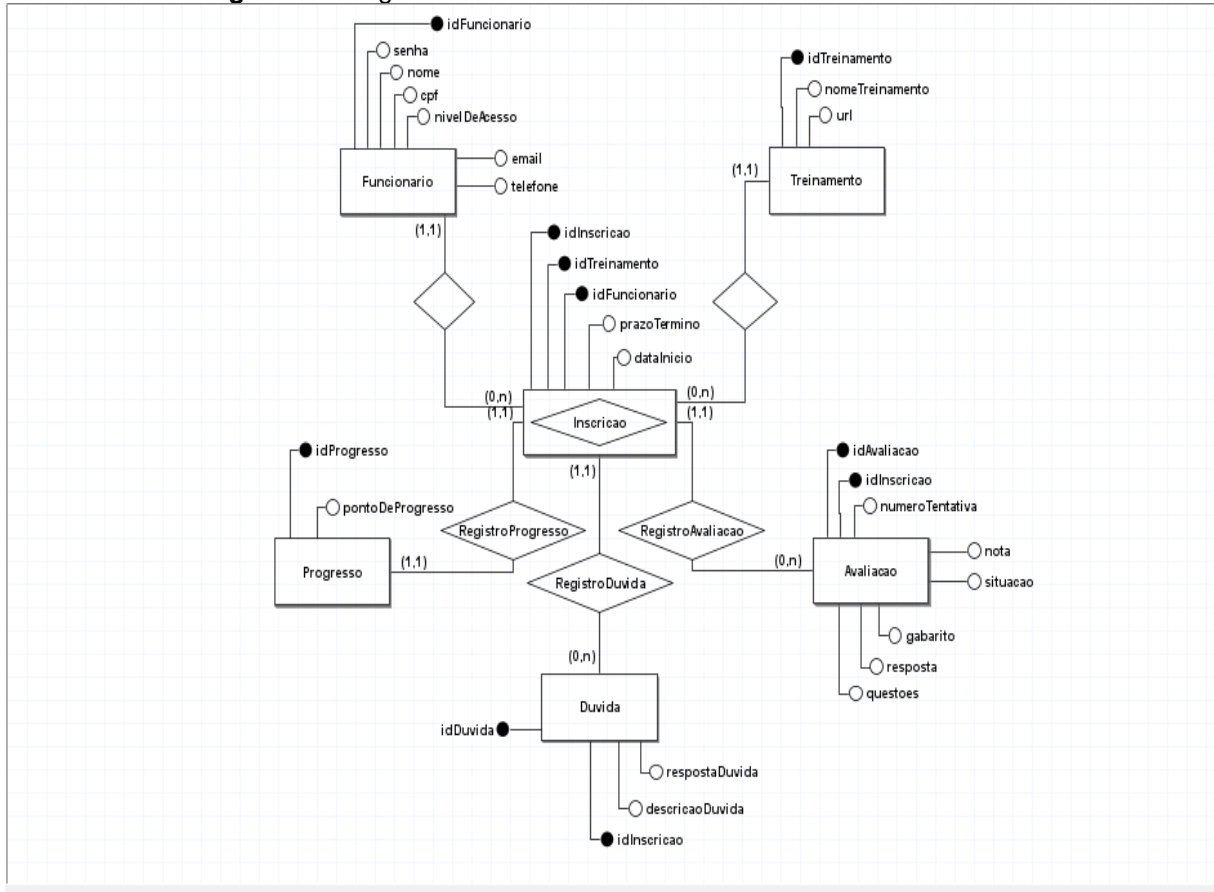
3.5 Diagrama Entidade-Relacionamento

Um diagrama de entidade relacionamento, também conhecido como DER's ou modelos ER, é um fluxograma que ilustra os relacionamentos entre entidades(objetos) de um sistema. Eles são mais utilizados para projetar bancos de dados relacionais e muitas vezes são utilizados junto com diagramas de fluxo de dados (DFDs), que mapeiam o fluxo de informações para processos ou sistemas (Lucidchart, 2023).

O DER é um modelo lógico fácil de ser interpretado. Ele mostra como os dados fluem de uma entidade para outra, facilitando a visualização da estrutura de um sistema pelos desenvolvedores e *designers* (Miro, 2023).

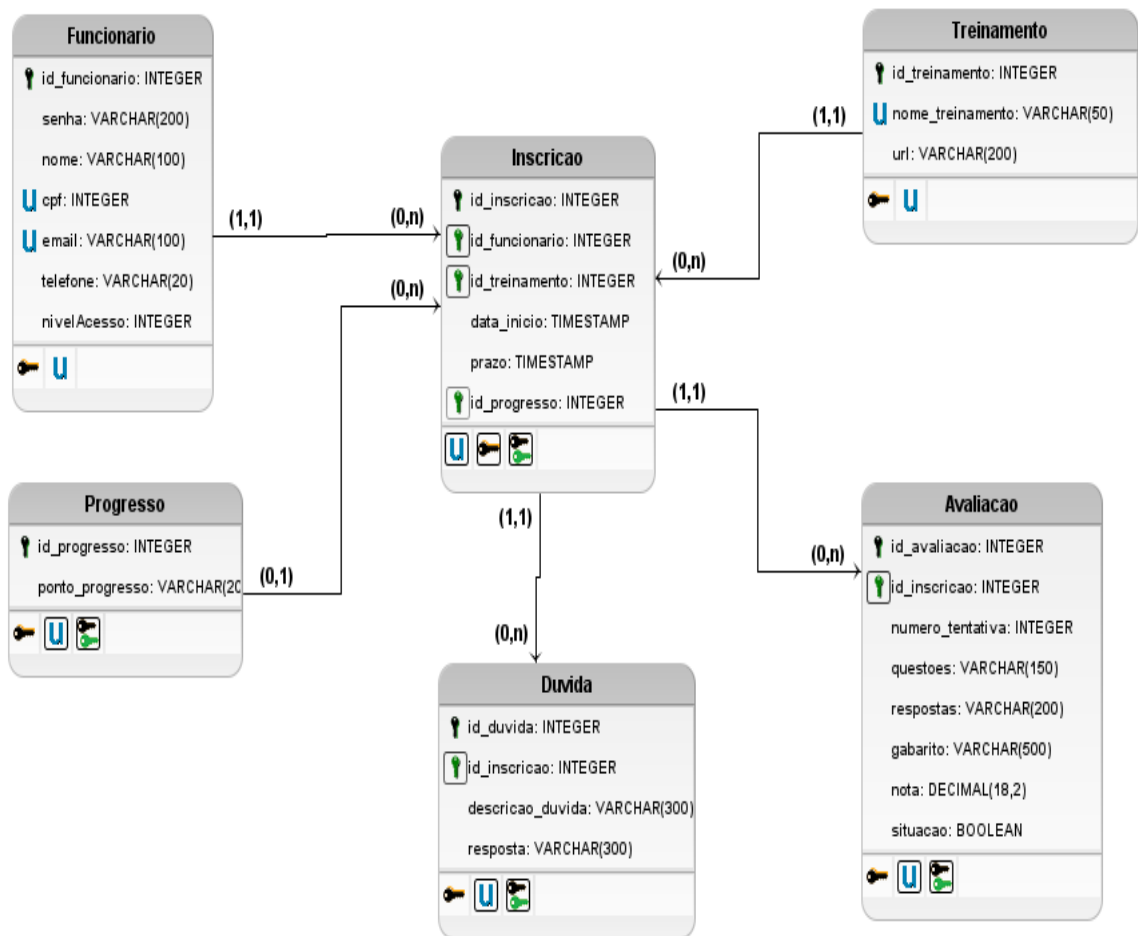
O diagrama de entidade de relacionamento do projeto está disposto em seu modelo conceitual na Figura 6 e no modelo lógico na Figura 7.

Figura 6 – Diagrama Entidade Relacionamento – Modelo Conceitual



Fonte: elaborado pelos autores.

Figura 7 – Diagrama Entidade Relacionamento – Modelo Lógico



Fonte: elaborado pelos autores.

4 Ferramentas e Métodos

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e suporte à comunidade. Além disso, essas ferramentas têm documentação abrangente, tutoriais e recursos disponíveis na comunidade de desenvolvedores, o que torna mais fácil para os desenvolvedores aprenderem e implementarem as soluções. A escolha dessas ferramentas também foi influenciada pela preferência pessoal da equipe de desenvolvimento e experiência prévia no uso delas. As licenças das ferramentas são de código aberto, o que significa que são gratuitas e podem ser usadas para fins comerciais e pessoais.

4.1 Ferramentas

4.1.1 Banco de dados

Foi escolhido o SGBD PostgreSQL na versão 11 por ser um *software* com licença de código aberto que oferece recursos necessários para o projeto, e por estar disponível em uma plataforma *online* e gratuita para estudantes no *site* da empresa Clever Cloud

Postgresql (2023) descreve o PostgreSQL como um poderoso sistema de banco de dados objeto-relacional de código abertos que combina a linguagem SQL com recursos capazes de lidar com cargas de trabalho mais complicadas. A origem do PostgreSQL como parte do projeto POSTGRES da Universidade da Califórnia em Berkeley no ano de 1986. Possui uma forte reputação por sua arquitetura comprovada, confiabilidade e empenho da comunidade de código aberto que mantém o software para fornecer consistentemente soluções inovadoras de alto desempenho.

Desde 2010, Clever-Cloud (2023) considera que a hospedagem industrializada permitirá que as empresas trabalhem mais rápido, e sejam mais ágeis em seus mercados, foquem em seu valor agregado e diminuam sua preocupação com tecnologias de hospedagem.

4.1.2 *Back-end*

Para o desenvolvimento do *back-end* foi escolhido o ambiente Node.js versão 18.12.1 em conjunto com o gerenciador de pacotes NPM por serem de código aberto e possuir recursos que facilitam o desenvolvimento de *software*. “Como um tempo de execução assíncrono conduzido por eventos de JavaScript, Node.js é projetado para construir aplicativos escaláveis e de rede” (Nodejs, 2023).

Bessa (2023, *online*), descreve o Node.js como um ambiente de execução do código JavaScript do lado do servidor, possibilitando a criação de aplicações *standalone* (autossuficientes) em uma máquina servidora, sem a necessidade de um navegador.

Ainda segundo Bessa (2023):

O Node.js nasceu em 2009 como uma resposta às tentativas de rodar códigos JavaScript em modo *server side*, uma vez que a linguagem tinha como meta principal a manipulação do DOM (Document Object Model) e deixar as aplicações web mais interativas e dinâmicas.

Soma-se a isso o fato de que pessoas desenvolvedoras de JavaScript trabalham com uma linguagem simples, interpretada e que não necessita da instalação de ferramentas complexas de desenvolvimento.

Esses são alguns dos fatores motivadores do criador do projeto do Node.js, o engenheiro de *software* Ryan Dahl, responsável por esse ambiente de execução do código JavaScript fora do navegador, no lado servidor.

Para Gaspar (2021, *online*) é necessário instalar e gerenciar ferramentas e pacotes Node.js para utilizá-los, sendo essa função do NPM. O NPM permite que os desenvolvedores instalem, atualizem e desinstalem dependências em um único comando.

Para a comunicação do *back-end* com o servidor do banco de dados foram utilizadas as dependências pg na versão 8.10.0, pg-hstore na versão 2.3.4, sequelize na versão 6.31.0 e sequelize-cli na versão 6.6.0.

A dependência pg, conforme descrito por Npmjs (2023), é um cliente PostgreSQL para o ambiente Node.js que realiza uma gama de conexões entre ambos, possibilitando a extensão de tipos de dados em ambas as direções entre o Javascript e o PostgreSQL.

A dependência pg-hstore, conforme descrito por Npmjs (2021), é um pacote para Node.js utilizado para lidar com serializações de dados em formato JSON para o formato hstore que, por sua vez, é um formato de dados que armazena pares chave-valor em um único valor.

De acordo com Sequelize (2023, *online*), Sequelize é um moderno ORM disponível para TypeScript e Node.js com suporte sólido a transações, relações, carregamento, replicação de leitura entre outras funções para SGBD como MySQL, MariaDB, SQLite e SQL Server entre outros.

Segundo Alura (2023, *online*), o Sequelize abstrai comandos de operações de SQL, e permite que o desenvolvedor utilize apenas a linguagem de programação empregada no back-end para conectar e operar o banco de dados. Tornando mais ágil e fácil o desenvolvimento do projeto.

Segundo Sequelize (2023), semelhante ao controle de versionamento disponibilizado pelo Git para códigos de desenvolvimento, as migrações permitem o controle de versionamento de estados do banco de dados, possibilitando transições entre estados do banco de dados. Para isto, utiliza-se a CLI (Sequelize Command-Line Interface) que oferece suporte para migrações e inicialização de projetos.

Uma migração no Sequelize é um arquivo JavaScript que exporta duas funções e, que regem como executar a migração de desfazê-la.

Foi escolhido o *framework* Express na versão 4.18.2 pois, de acordo com Mozilla (2023, *online*) oferece soluções de gerenciamento de requisições de diferentes verbos HTTP em diferentes URLs; integra "*view engines*" para inserir dados nos *templates*; define configurações comuns de aplicações web, como a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta; adicionar novos processos de requisição por meio de "*middleware*" em qualquer ponto da "fila" de requisições.

“O Express é uma estrutura de aplicação web Node.js mínima e flexível que fornece um conjunto robusto de recursos para aplicativos web e móveis” (Expressjs, 2023, *online*).

A dependência jsonwebtoken (versão 9.0.0) foi utilizada para tratar requisições HTTP, pois conforme Bito.ai (2023, *online*), é um meio compacto e seguro de URL para representar requisições entre duas partes. Geralmente é utilizado para autenticar usuários e é facilmente transmitido via URL.

A dependência dotenv versão 16.0.3 foi utilizada para armazenar valores de variáveis de sistema pois segundo Gomes (2023, *online*) “O Dotenv é um pacote que serve justamente para gerenciar as variáveis de ambiente dentro de um projeto

Node.js. Essa ferramenta armazena a configuração dessas variáveis em um ambiente separado do código da aplicação”.

Ainda segundo Gomes (2023, *online*), variáveis de ambiente armazenam os dados de configuração do sistema, informações sensíveis como chaves de API's, credenciais de acesso a banco de dados, porta HTTP entre outros. Portanto, uma boa prática é mantê-las separadas do código.

A dependência *cookie-parser* versão 1.4.4 é um analisador de *cookie* utilizado para analisar informações inseridas nas requisições HTTP geradas pelos usuários.

A dependência *bcrypt* versão 5.1.0 foi escolhida como método de criptografia para senhas de usuário. De acordo com Kerlyn (2019, *online*):

O BCrypt foi desenvolvido por Niels Provos e David Mazières (1999) com o propósito de esconder senhas criadas pelos usuários em forma de texto “puro” em dados indecifráveis, utilizando o algoritmo hash. Essa é uma ferramenta segura para armazenar senhas no banco de dados e pode ser utilizada por qualquer linguagem (C, C ++, C #, Go, Java, JavaScript, Elixir, Perl, PHP, Python, Ruby e outros).

O BCrypt possui dois critérios importantes:

O salt é uma das vantagens do BCrypt, pois acrescenta aleatoriamente sequências de caracteres a senha, projetando resultados criptográficos complexos e aumentando a segurança contra-ataques de força bruta, como o rainbow tables, ou seja, um hash sempre será diferente, mesmo que a senha seja igual.

A outra vantagem do BCrypt é a possibilidade de alterar valores do *saltRounds* (relacionado a custos), onde quanto maior o número fornecido, mais lento será o processamento para calcular o hash associado a senha. Por padrão, o valor utilizado é 10, mas é importante salientar que um valor mais alto demandará mais tempo para encontrar as possíveis senhas nos casos de ataque de força bruta. Então esse valor deve ser pequeno o suficiente para não demorar na verificação no *login* do usuário.

4.1.3 *Front-end*

Foi utilizado o *framework* React versão 18.2.0 em conjunto com a ferramenta de desenvolvimento Vite versão 4.2.0 para facilitar a construção da interface do usuário.

“O React permite criar interfaces de usuário a partir de peças individuais chamadas componentes. Crie seus próprios componentes do React e, em seguida, combine-os em telas, páginas e aplicativos inteiros” (React, 2023, *online*).

Segundo Neves (2023, *online*), um *framework* é um conjunto de ferramentas e bibliotecas que fornecem recursos e funcionalidades prontas para desenvolvedores

de *software*. Permitindo que construam aplicativos rapidamente, com menos esforço e código, diminuindo as chances de erros.

Ainda de acordo com Neves (2023, *online*), React é um *framework* JavaScript criado pelo Facebook (atual Meta) usado para criar interfaces de usuário (UI) em aplicativos web. Popular por ser fácil de usar, flexível e escalável, é usado por muitas empresas de tecnologia.

Segundo Vite (2023, *online*):

...é uma ferramenta de construção de projetos de *frontend* que se destina a oferecer uma experiência de desenvolvimento mais rápida e leve para projetos de web modernos. Ela consiste em duas partes principais:

Um servidor de desenvolvimento que oferece melhorias de funcionalidade ricas sobre módulos de ECMAScript nativo, por exemplo Substituição de Módulo Instantânea extremamente rápida.

Um comando de construção que empacota o nosso código com a *Rollup*, pré-configurado para produzir recursos estáticos altamente otimizados para produção.

Foram utilizados componentes da biblioteca Material UI que implementa o estilo do Google. Em sua página oficial MUI (2023, *online*) declara “A MUI oferece um conjunto abrangente de ferramentas de interface do usuário gratuitas para ajudá-lo a enviar novos recursos mais rapidamente”.

A dependência *joi* versão 17.9.2 foi utilizada para realizar validações de dados de entrada no *front-end* pois conforme *Joi* (2023, *online*) descreve em sua página oficial: “O *Joi* permite que você descreva seus dados usando uma linguagem simples, intuitiva e legível”.

4.2 Métodos

O primeiro passo do desenvolvimento do projeto de *software* foi a criação do banco dados no servidor da plataforma *Clever-Cloud*. Em seguida foi criada a API *back-end* utilizando o *Node.js* com os dados de conexão armazenados em variáveis de sistema. Foram instaladas as dependências necessárias para estabelecer a conexão com o servidor do banco dados e para realizar as conversões entre as linguagens *Javascript* e *PostgreSQL*.

Continuando, foram criadas as tabelas no banco de dados por intermédio da API gerando arquivos *migrations*, que permitem a manipulação de estado do banco de dados, e arquivos *models*, que servem com o modelo para gravação de dados no

banco de dados. Em seguida foram criados arquivos *controllers* que armazenam a lógica de dados criação, leitura, alteração e exclusão de arquivos do banco de dados.

No passo seguinte, foram criados arquivos *routes* que estabelecem as rotas de acesso dos métodos de criados anteriormente ao banco de dados.

Finalizado o *back-end*, foi criado o projeto do *front-end* com o Vite, utilizando componentes do MUI para construir a interface de usuário, do React para realizar a captura dos dados de entrada, e o Joi para validação de formato de dados de entrada.

5 Desenvolvimento

A implementação parcial do projeto contempla os casos de uso 001 (cadastro de treinamento) e 002(cadastro de funcionários).

O computador utilizado para o desenvolvimento possui uma configuração semelhante a computadores de escritório e pessoais, com sistema operacional Windows 11, *hardware* com processador Intel Core i5-10210U a 1.6 GHz, 8 GB de memória RAM e um disco rígido SSD com 240 GB, sendo essa configuração suficiente para que o *software* apresente uma boa *performance*.

A aplicação foi desenvolvida na linguagem de programação *Javascript* por meio do editor de código Visual Studio Code da Microsoft.

O *back-end* da aplicação foi desenvolvido em Node.js conectado a um banco de dados por meio do SGBD PostgreSQL hospedado na plataforma *online* da empresa Clever Cloud. Para criação e gerenciamento das tabelas foi utilizado o ORM Sequelize do Node.js. Também foi utilizado o *framework* Express para gerenciamento das rotas da API.

A ferramenta Vite foi utilizada para agilizar e facilitar o processo de desenvolvimento da interface do usuário no *front-end*.

5.1 Banco de dados

Para Oracle (2023, *online*):

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

SQL é uma linguagem de programação usada por quase todos os bancos de dados relacionais para consultar, manipular e definir dados e fornecer controle de acesso. O SQL foi desenvolvido pela primeira vez na IBM nos anos 1970, com a Oracle como principal contribuinte, o que levou à implementação do padrão SQL ANSI; o SQL estimulou muitas extensões de empresas como IBM, Oracle e Microsoft. Embora o SQL ainda seja amplamente usado hoje em dia, novas linguagens de programação estão começando a aparecer.

A base de dados da aplicação foi nomeada como “treinamento”, utilizando o SBGD PostgreSQL e hospedado na plataforma *online* da empresa Clever Cloud conforme demonstrado pela Figura 8.

Figura 8 - Banco de Dados

The screenshot shows the PostgreSQL Studio interface. On the left, there is a sidebar with a navigation menu including 'projeto', 'Contact support team', 'Create...', 'Filter apps, add-ons...', and 'treinamento'. The main area is titled 'PostgreSQL by Clever Cloud' and shows a table with columns: TYPE, PLAN, CLUSTER, VERSION, REGION, STATUS, CREATED, and ID. Below this, there is a table with one row: PostgreSQL Dev par-postgresql-11 par ACTIVE 2023-04-15 postgresql_167cdb45-1483-4d01-917d-f177413cbf54. The interface also shows the PostgreSQL logo and 'STUDIO' text. Below the logo, there are tabs for 'Schema', 'Columns', 'Indexes', 'Constraints', 'Triggers', and 'Rules'. The 'Schema' tab is selected, showing a list of tables: SequelizeMeta, avaliacoes, duvidas, funcionarios, inscricoes, progressos, spatial_ref_sys, and treinamentos. The 'treinamentos' table is selected, and its columns are displayed in a table view:

Column Name	DK	PK	Data Type
id			integer
nome_treinamento			character varying(50)
url			character varying(200)
createdAt			timestamp with time zone
updatedAt			timestamp with time zone

Fonte: elaborado pelos autores.

As tabelas do banco de dados foram criadas via terminal por meio de comandos executados no editor de código Visual Studio Code, conforme demonstrado na Figura 9. “Editores de código são editores de texto com poderosos recursos integrados e funcionalidades especializadas para simplificar e acelerar o processo de edição de código” (Lima, 2022, *online*).

Figura 9 – Exemplo de *script* de criação de tabelas

```
1  criar model + migrations de tabelas
2  npx sequelize-cli model:generate --name 'Nome da tabela em pascal case'
3  --attributes 'campos da tabela ex: name:string,email:string...'
4
5  Tabela Funcionarios
6  npx sequelize-cli model:generate --name Funcionario
7  --attributes name:string,senha:string,cpf:integer,email:string,
8  telefone:string,nivel_acesso:integer
9
10 Tabela Progressos
11 npx sequelize-cli model:generate --name Progresso
12 --attributes id_inscricao:integer,ponto_progresso:string
13
```

Fonte: elaborado pelos autores.

A Figura 10 exemplifica parte de um destes arquivos criados pelo *script*. Estes *scripts* utilizam a ORM Sequelize para criar as tabelas no banco de dados e geram dois arquivos. ORM (*Object Relational Mapper*) é uma técnica de mapeamento objeto relacional que permite fazer uma relação dos objetos com os dados que representam, em outras palavras é uma ferramenta que abstrai os comandos SQL para os desenvolvedores. O primeiro arquivo é gerado na pasta denominada “*migrations*” e registra as alterações realizadas na base de dados e pode ser utilizado como ponto

Figura 10 - Migration

```

1  'use strict';
2  /** @type {import('sequelize-cli').Migration} */
3  module.exports = {
4    async up(queryInterface, Sequelize) {
5      await queryInterface.createTable('treinamentos', {
6        id: {
7          allowNull: false,
8          autoIncrement: true,
9          primaryKey: true,
10         type: Sequelize.INTEGER
11       },
12       nome_treinamento: {
13         type: Sequelize.STRING(50),
14         allowNull: false,
15       },
16       url: {
17         type: Sequelize.STRING(200),
18         allowNull: false,
19       },
20       createdAt: {
21         allowNull: false,
22         type: Sequelize.DATE
23       },
24       updatedAt: {
25         allowNull: false,
26         type: Sequelize.DATE
27       }
28     });
29   },
30   async down(queryInterface, Sequelize) {
31     await queryInterface.dropTable('treinamentos');
32   }
33 };

```

Fonte: elaborado pelos autores.
restauração para recuperar o estado anterior a alteração

realizada no banco de dados. Pode ainda ser utilizado para recriar a tabela caso seja necessário mudança do *hardware* de hospedagem do banco de dados.

Esse arquivo contém a lógica de criação de uma tabela e estabelece os parâmetros das colunas presentes nela. A lógica utiliza duas funções assíncronas para acessar o banco de dados, sendo a primeira para criação da tabela e a segunda para deletar a tabela.

A Figura 11 mostra um exemplo de arquivo criado na pasta *models*. Estes arquivos são utilizados como modelo para inserção de instâncias em uma tabela do banco de dados. Também é neles que são inseridas as associações entre as tabelas do banco de dados, conforme descrito no diagrama de entidade de relacionamento.

Figura 11 - Models

```

1  'use strict';
2  const { Model } = require('sequelize'); 1.6M (gzipped: 272.8k)
3  You, há 1 segundo • Uncommitted changes
4  module.exports = (sequelize, DataTypes) => {
5      You, há 5 segundos | 2 authors (lucrc and others)
6      class Treinamento extends Model {
7          static associate(models) {
8              this.belongsToMany(models.Funcionario, {
9                  through: 'inscricoes', //tabela intermediaria
10                 foreignKey: 'id_treinamento', // chave estrangeira
11                 otherKey: 'id_funcionario', // outra chave da tabeoa intermediaria
12                 as: 'funcionarios' // Nome do campo de associação (plural)
13             })
14         }
15     }
16     Treinamento.init({
17         id: {
18             allowNull: false,
19             autoIncrement: true,
20             primaryKey: true,
21             type: DataTypes.INTEGER
22         },
23         nome_treinamento: {
24             type: DataTypes.STRING(50),
25             allowNull: false,
26         },
27         url: {
28             type: DataTypes.STRING(200),
29             allowNull: false,
30         },
31     }, {
32         sequelize,
33         modelName: 'Treinamento',
34         tableName: 'treinamentos'
35     });
36     return Treinamento;
};

```

Fonte: elaborado pelos autores.

A configuração da conexão com o banco de dados é realizada utilizando os dados gravados nos arquivos “config.js” (Figura12) e “.env” que possui o mesmo padrão descrito no arquivo “.env.example”. (Figura 13).

Figura 12 – Arquivo config.js

```

1  require('dotenv').config()  2.9k (gzipped: 1.4k)
2
3  module.exports = {
4    "development": {
5      "username": process.env.DB_USERNAME,
6      "password": process.env.DB_PASSWORD,
7      "database": process.env.DB_DATABASE,
8      "host": process.env.DB_HOST,
9      "dialect": process.env.DB_DIALECT,
10   },
11   "test": {
12     "username": "root",
13     "password": null,
14     "database": "database_test",
15     "host": "127.0.0.1",
16     "dialect": "mysql",
17   },
18   "production": {
19     "username": "root",
20     "password": null,
21     "database": "database_production",
22     "host": "127.0.0.1",
23     "dialect": "mysql",
24   }
25 }

```

Fonte: elaborado pelos autores.

Figura 13 – Arquivo .env.example

```

1  DB_USERNAME=
2  DB_PASSWORD=
3  DB_DATABASE=
4  DB_HOST=
5  DB_DIALECT=
6  TOKEN_SECRET=
7

```

Fonte: elaborado pelos autores.

5.2 Back-end

O *back-end* da aplicação foi desenvolvido utilizando o *framework* Express, que segundo HostGator (2023, *online*):

...é uma espécie de *template* que conta com diversas funcionalidades que podem ser utilizados pelo desenvolvedor em seus projetos.

Ele conta com ferramentas, sistemas, componentes e guias que agilizam o processo de criação de soluções, sendo, portanto, um artifício essencial na vida de um profissional de TI.

É importante destacar que *framework* não é a mesma coisa que biblioteca. Em outras palavras, uma biblioteca é feita para trabalhar de forma independente de onde ela será implementada. No entanto, um *framework*, ao ser adotado, irá reger, conduzir e guiar todo o desenvolvimento da aplicação que está sendo construída com ele.

Foram utilizadas as dependências listadas na Figura 14, para facilitar a construção do sistema.

Figura 14 - Dependências

```

1  {
2  |   "name": "back-end",
3  |   "version": "0.0.0",
4  |   "private": true,
5  |   "scripts": {
6  |     | "start": "node ./bin/www"
7  |   },
8  |   "dependencies": {
9  |     | "bcrypt": "^5.1.0",
10 |     | "cookie-parser": "~1.4.4",
11 |     | "cors": "^2.8.5",
12 |     | "debug": "~2.6.9",
13 |     | "dotenv": "^16.0.3",
14 |     | "express": "^4.18.2",
15 |     | "jsonwebtoken": "^9.0.0",
16 |     | "morgan": "~1.9.1",
17 |     | "multer": "^1.4.5-lts.1",
18 |     | "pg": "^8.10.0",
19 |     | "pg-hstore": "^2.3.4",
20 |     | "sequelize": "^6.31.0",
21 |     | "sequelize-cli": "^6.6.0"
22 |   }
23 | }

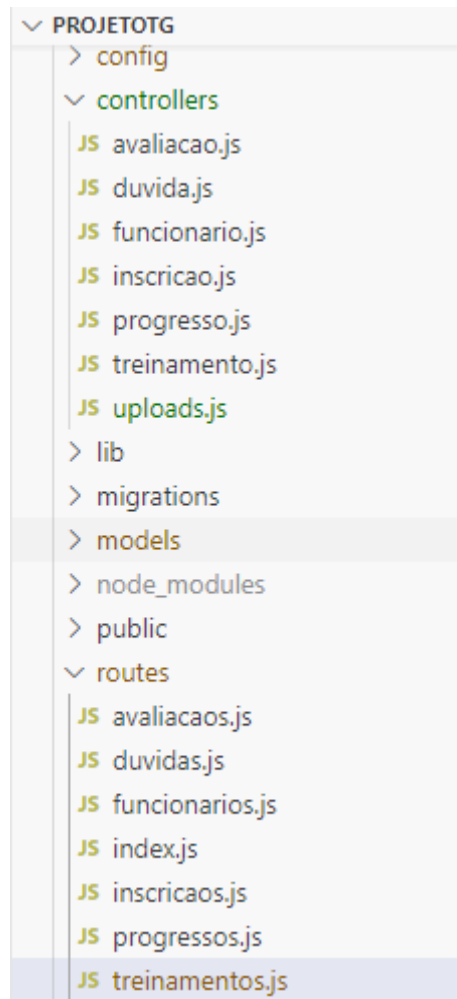
```

Fonte: elaborado pelos autores.

O acesso da aplicação ao banco de dados foi estruturado por meio da criação de rotas de acesso a diretórios correspondentes as tabelas do banco de dados. Em

seguida, são utilizadas funções assíncronas descritas nos arquivos *controllers* para realizar a comunicação entre o *back-end* e o banco de dados conforme ilustrado pelas Figuras 15 a 18.

Figura 15 – Diretório de arquivos do sistema



Fonte: elaborado pelos autores.

Figura 16 – Simplificação de rotas de acesso

```

const funcionarios=require('./routes/funcionarios')
app.use('/funcionarios', funcionarios)

const progressos=require('./routes/progressos')
app.use('/progressos', progressos)

const duvidas=require('./routes/duvidas')
app.use('/duvidas', duvidas)

const avaliacaos=require('./routes/avaliacaos')
app.use('/avaliacaos', avaliacaos)

const treinamentos=require('./routes/treinamentos')
app.use('/treinamentos', treinamentos)

const inscricaos=require('./routes/inscricaos')
app.use('/inscricaos', inscricaos)

const uploads=require('./routes/uploads')
app.use('/uploads', uploads)

module.exports = app;

```

Fonte: elaborado pelos autores.

Figura 17 – Direcionamento do acesso conforme método selecionado pelo usuário

```

1  const express = require('express');
2  const router = express.Router();
3  // const multer = require('multer')
4  // const multerConfig = require('../config/multer')
5  const controller = require('../controllers/treinamento')
6
7
8  router.post('/', controller.create)
9  router.get('/', controller.retrieve)
10 router.get('/:id', controller.retrieveOne)
11 router.put('/:id', controller.update)
12 router.delete('/:id', controller.delete)
13
14 module.exports = router;

```

Fonte: elaborado pelos autores.

Figura 18 – Exemplo de lógica de um *controller* com método de post de dados no banco

```

1  //Importar o model correspondente ao controller
2  const { ConnectionTimedOutError } = require('sequelize') 1
3  const { Treinamento, Funcionario } = require('../models')
4
5  const controller = {} // objeto vazio
6
7  /*
8   Métodos CRUD do controller
9   create: cria um novo registro
10  retrieve: lista (recupera) todos os registros
11  retrieveOne: lista (recupera) apenas um registro
12  update: atualiza um registro
13  delete: exclui um registro
14  */
15
16  controller.create = async(req, res) => {
17    try{
18      await Treinamento.create(req.body)
19      //HTTP 201: Created
20      res.status(201).end()
21    }
22    catch(error){
23      console.error(error)
24    }
25  }
26

```

Fonte: elaborado pelos autores.

A lógica presente no arquivo *controller* utiliza funções assíncronas para realizar as operações de criação, leitura, alteração e exclusão (CRUD na sigla em inglês) conforme a operação escolhida pelo usuário através das rotas de acesso. Recebendo como parâmetro os dados a serem inseridos, recuperados, alterados ou excluídos do banco de dados, é feito o envio da solicitação e aguardado a resposta do servidor do banco de dados. Caso a operação não seja realizada com sucesso é retornado um erro para o usuário.

5.3 Front-end

O *front-end* da aplicação foi desenvolvido com a ferramenta Vite, que conforme Seguins (2023, *online*):

O Vite é uma ferramenta que resolve os problemas comuns enfrentados no desenvolvimento JavaScript e TypeScript. Desenvolvido por Evan You, o criador do Vue.js, o Vite introduz o conceito de "carregamento rápido" (*fast refresh*), que permite um desenvolvimento mais ágil e uma experiência de desenvolvimento mais suave.

Ele se destaca por sua velocidade e eficiência quando comparado ao Webpack, especialmente em projetos maiores. Conforme os projetos JavaScript cresceram em complexidade, o Webpack, embora poderoso, começou a demonstrar um desempenho inferior ao recarregar alterações e iniciar o servidor de desenvolvimento. Isso resultou em experiências frustrantes para as pessoas desenvolvedoras e tempos de construção mais longos.

A Figura 19 exibe a lógica de renderização da página inicial do sistema e a Figura 20 exibe a renderização da página na tela do usuário.

Figura 19 – Lógica da tela inicial

```

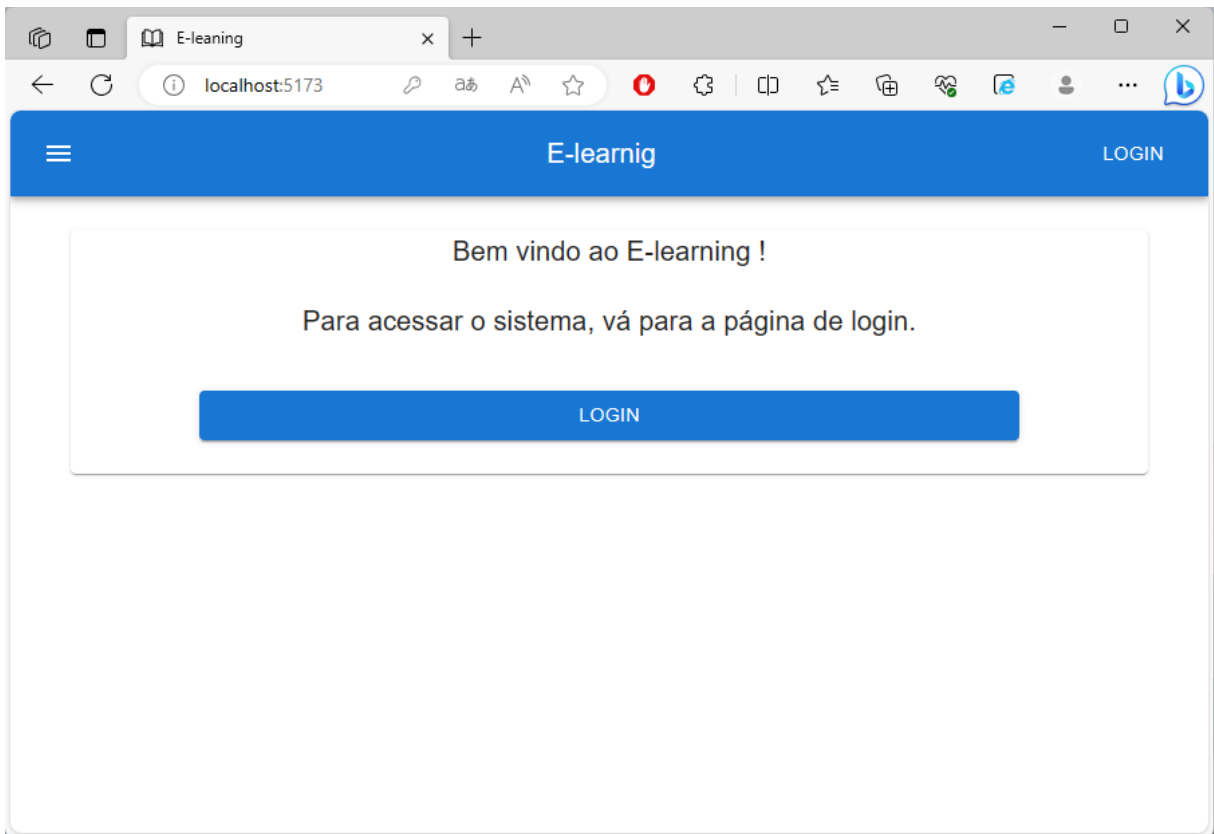
1  import React from "react" 6.9k (gzipped: 2.7k)
2  import Typography from "@mui/material/Typography" 30.5k (gzipped: 11k)
3  import Paper from '@mui/material/Paper' 30.1k (gzipped: 10.9k)
4  import Button from '@mui/material/Button' 48.1k (gzipped: 16.2k)
5  import {Link} from 'react-router-dom' 7.3k (gzipped: 3k)
6
7  export default function Home(){
8      return (
9          <>
10         <Paper sx={{width: '512', maxWidth: '90%', margin: '24px auto 0 auto', p: '12'}} >
11             <Typography variant="h6" textAlign='center'>
12                 <p>Bem vindo ao E-learning ! </p>
13                 <p>Para acessar o sistema, vá para a página de login. </p>
14             </Typography>
15             <Typography variant="h5" component="div">
16                 <form >
17                     <Button sx={{ m: 2 }} className= 'form-field' variant='contained'
18                         component={Link} to='/Login' fullWidth> Login </Button>
19                 </form>
20             </Typography>
21         </Paper>
22     </>
23 )
24 }
25

```

Fonte: elaborado pelos autores.

Na linha 18 da Figura 19 vê-se que é feito um redirecionamento para página de *login* quando o usuário clica no botão descrito como “*Login*”.

Figura 20 – Tela inicial renderizada



Fonte: elaborado pelos autores.

A Figura 21 mostra a lógica de coleta dados e de comunicação com o *back-end* da página de *login*, complementada pela lógica de renderização mostrada na Figura 22. São utilizadas variáveis de estado do React para armazenar os dados inseridos pelo usuário no sistema. Eles serão transmitidos para o banco de dados com a utilização de funções assíncronas de acesso ao *back-end*. Em caso de falha durante este processo, é exibido um alerta de erro para o usuário. A mesma lógica de captura de dados e comunicação com *back-end* para acessar o banco de dados é comum para todos os elementos do sistema.

Figura 21 – Lógica de interatividade da página de *login*.

```

11 export default function Login(){
12   const [email, setEmail] = React.useState('')
13   const [senha, setSenha] = React.useState('')
14   const [showWaiting, setShowWaiting] = React.useState(false)
15   const [notif, setNotif] = React.useState({show: false, message: '', severity: 'success' })
16   function handleChange(event){
17     if(event.target.name === 'email') setEmail(event.target.value)
18     else setSenha(event.target.value)
19   }
20   async function handleSubmit(event){event.preventDefault()
21     setShowWaiting(true)
22     try {
23       const result = await myfetch.post('/funcionarios/login', { email, senha })
24       window.localStorage.setItem('token', result.token)
25       setNotif({show:true,message:'Autenticação realizada com sucesso!',severity:'success'})
26     }
27     catch(error) {
28       console.error(error)
29       window.localStorage.removeItem('token')
30       setNotif({show: true, message: error.message, severity: 'error'})
31     }
32     finally {setShowWaiting(false)}
33   }
34   function handleNotifClose(event, reason) {
35     if (reason === 'clickaway') {
36       return;
37     }
38     setNotif({ show: false })
39   };
40 }

```

Fonte: elaborado pelos autores.

Figura 22 – Lógica de renderização da página de *login*.

```

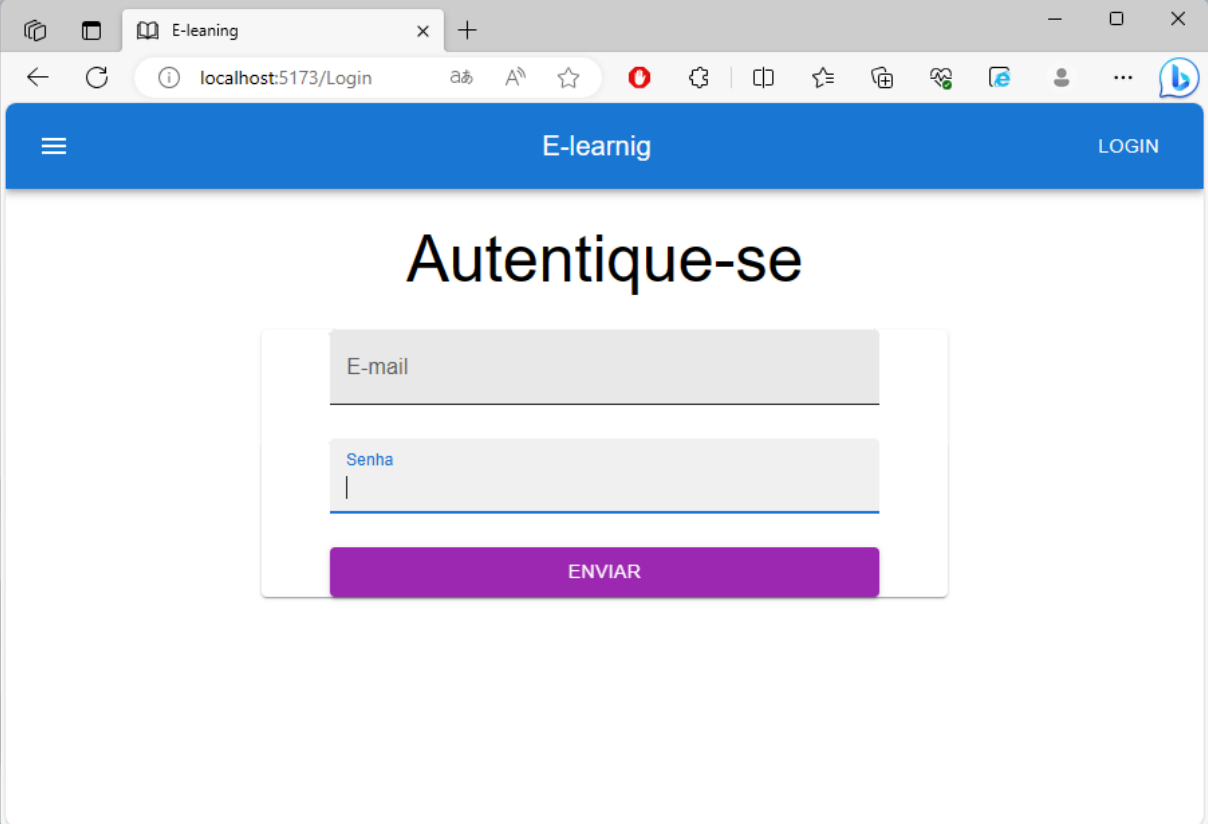
41 return (
42   <>
43     <Backdrop sx={{ color: '#fff', zIndex: (theme) => theme.zIndex.drawer + 1 }}
44     | open={showWaiting}> <CircularProgress color="inherit" /> </Backdrop>
45     <Notification show={notif.show} severity={notif.severity} onClose={handleNotifClose}>
46     | {notif.message} </Notification>
47     <Typography variant="h3" component="h1" sx={{textAlign: 'center'}}> Autentique-se </
48     | Typography>
49     <Paper sx={{width: '512px', maxWidth: '90%', margin: '24px auto 0 auto',p: '12'}} >
50     | <Typography variant="h5" component="div">
51     |   <form onSubmit={handleSubmit}>
52     |     <TextField id="email" className= 'form-field' name= 'email' label="E-mail"
53     |     | variant="filled" fullWidth onChange={handleChange} value={email} />
54     |     <TextField id="senha" className= 'form-field' name= 'senha' label="Senha"
55     |     | variant="filled" type= "password" fullWidth onChange={handleChange} value={senha}/>
56     |     <Button variant="contained" type="submit" color="secondary" fullWidth>
57     |     | Enviar
58     |     </Button>
59     |   </form>
60     | </Typography>
61   </Paper>
62 </>
63 )

```

Fonte: elaborado pelos autores.

Na Figura 23 é exibida a tela de *login* renderizada.

Figura 23 – Tela de *login* de usuário.



The image shows a web browser window with the following elements:

- Browser tab: E-learning
- Address bar: localhost:5173/Login
- Page header: E-learnig (left), LOGIN (right)
- Main heading: Autentique-se
- Form fields:
 - E-mail
 - Senha
- Submit button: ENVIAR

Fonte: elaborado pelos autores.

As Figuras 24 e 25 exibem a lógica de interatividade da página de cadastro de treinamentos, e a Figura 26 mostra a tela de cadastro que é renderizada para os usuários. A lógica de renderização segue a mesma técnica em todas as páginas. Portanto, optou-se por mostrar apenas os exemplos anteriores nas Figuras 19 e 22 sendo da tela inicial e de *login* respectivamente.

Figura 24 – Lógica de interatividade da página de cadastro de treinamentos – parte 1

```

16 export default function TreinamentoForm () {
17   const API_PATH = '/treinamentos'
18   const navigate = useNavigate()
19   const params = useParams()
20   const [state, setState] = React.useState({
21     treinamento:{nome_treinamento:'',url: ''},
22     showWaiting: false,
23     errors:{},
24     notif:{show: false, severity: 'success',message: ''}
25   })
26   const {treinamento, showWaiting, notif, errors} = state
27   function handleFormFieldChange(event) {
28     const treinamentoCopy = {...treinamento}
29     treinamentoCopy[event.target.name] = event.target.value
30     setState({...state, treinamento: treinamentoCopy})
31   }
32   function handleFormSubmit(event){
33     event.preventDefault()
34     sendData()
35   }
36   React.useEffect(()=>{if(params.id) fetchData()}, [])
37   async function fetchData(){setState({...state,showWaiting:true, errors:{}})
38     try {
39       const result = await myfetch.get(`${API_PATH}/${params.id}`)
40       setState({...state, treinamento: result, showWaiting: false})
41     }
42     catch (error){
43       setState({...state, showWaiting: false, errors: errorMessages, notif:{
44         severity: 'error', show: true,message: 'ERRO ' + error.message}})
45     }
46   }

```

Fonte: elaborado pelos autores.

Figura 25 – Lógica de interatividade da página de cadastro de treinamentos – parte 2

```

47   async function sendData() {
48     console.log('sendData')
49     setState({...state, showWaiting: true, errors:{}})
50     try{
51       await TreinamentoMethod.validateAsync(treinamento, {abortEarly: false})
52       if(params.id) await myfetch.put(`${API_PATH}/${params.id}`, treinamento)
53       else await myfetch.post(API_PATH, treinamento)
54       setState({...state, showWaiting: false, notif: {severity: 'success', show: true,
55         message: 'Novo item salvo com sucesso'}})
56     }
57     catch(error){
58       const { validationError, errorMessages } = getValidationMessages(error)
59       setState({...state, showWaiting: false, errors: errorMessages, notif: {severity:
60         'error',show: !validationError, message: 'ERRO: ' + error.message}})
61     }
62   }
63   function handleNotifClose (event, reason) {
64     if (reason === 'clickaway') {
65       return;
66     }
67     if(notif.severity === 'success') navigate(-1)
68     setState({...state, notif: {...notif, show: false}});
69   };

```

Fonte: elaborado pelos autores.

Figura 26 -Página de cadastro de treinamentos renderizada.

A imagem mostra uma interface web para o cadastro de treinamentos. No topo, há uma barra azul com o nome 'E-learnig' e um link 'LOGIN'. Abaixo, o formulário principal tem o título 'Cadastrar novo Treinamento'. O formulário contém dois campos de entrada: 'Nome do treinamento *' e 'Caminho *'. À direita dos campos, há um botão verde com o texto 'ENVIAR' e um ícone de seta para a direita.

Fonte: elaborado pelos autores.

As Figuras 27 e 28 mostram a lógica de interatividade da página de listagem dos treinamentos cadastrados no sistema. A Figura 29 ilustra a renderização da página que mostra a lista de treinamentos cadastrados no sistema.

Figura 27—Lógica de interatividade da página de listagem de treinamentos cadastrados – Parte 1.

```

19 export default function TreinamentoList () {
20   const API_PATH = '/treinamentos'
21   const [state, setState] = React.useState({
22     treinamentos:[],
23     deleteId: null,
24     showWaiting: false,
25     showDialog: false,
26     notif: {show: false,message: '',severity: 'success' }
27   })
28   const {treinamentos, showWaiting, showDialog, deleteId, notif} = state
29   async function fetchData(){setState({...state, showWaiting:true})
30     try{
31       const result = await myfetch.get(API_PATH)
32       setState({...state, treinamentos: result, showWaiting: false, showDialog: false})
33     }
34     catch{console.error(error)}
35   }
36   React.useEffect(() => {fetchData()}, [])
37   const columns = [
38     { field: 'id', headerName: 'Cód.', width: 90 },
39     {field: 'nome_treinamento', headerName: 'Nome', width: 250},
40     {field: 'url', headerName: 'Link', width: 250},
41     {field: 'edit', headerName: 'Editar',headerAlign: 'center', width: 90,
42       renderCell: params => (<Link to='.'/' + params.id><IconButton aria-label="Editar">
43         <EditIcon /></IconButton></Link>)},
44     {field: 'delete', headerName: 'Excluir', headerAlign: 'center', align: 'center', width: 90,
45       renderCell: params => (<IconButton aria-label="excluir" onClick={() => setState({
46         ...state, deleteId: params.id, showDialog: true})}><DeleteForeverIcon color="error" />
47         </IconButton>)}
48   ];

```

Fonte: elaborado pelos autores.

Figura 28—Lógica de interatividade da página de listagem de treinamentos cadastrados – Parte 2.

```

49   async function handleDialogClose(answer){
50     if(answer){setState({...state, showWaiting: true, showDialog: false})
51       try{
52         const result = await myfetch.delete(`${API_PATH}/${deleteId}`)
53         setState({...state, showWaiting: false, showDialog: false, inscritos: result,
54           |   notif:{show: true, message: 'Listado com sucesso', severity: 'success'}})
55         fetchData()
56       }
57       catch{console.error(error)
58         setState({...state, showWaiting: false, showDialog: false,
59           |   notif: {show: true, message: 'Erro: ' + error.message, severity: 'error'}})
60       }
61     }
62     else{setState({...state, showDialog:false})}
63   }
64   function handleNotifClose (event, reason) {
65     if (reason === 'clickaway') {return;}
66     setState({...state, notif: {show: false}})
67   }

```

Fonte: elaborado pelos autores.

Figura 29 – Página de listagem de treinamentos cadastrados renderizada.

Cód.	Nome	Link	Editar	Excluir
1	Treinamento 1	www.youtube.com/E-learning-1		
2	Treinamento 2	www.youtube.com/E-learning-2		
15	Canvas 2.0	www.youtube.com/E-learning-Ca...		
16	Treinamento 3	www.youtube.com/E-learning-3		

Fonte: elaborado pelos autores.

6 Resultados e Discussão

6.1 Análise de Portabilidade

6.1.1 Dados da Empresa

Empresa atuante no ramo de distribuição de produtos alimentícios.

6.1.2 Infraestrutura

Hardware – Rede de Dados

Pontos de Acesso: 1

Meios e transmissão: Rede Cabeada e Wi-Fi.

Velocidade: 600mb operadora Algar Telecom

Equipamento: 01 Switch, 01 Hub.

Hardware – Rede Elétrica

Estabilizadores: 2

Nobreaks: 4

Dispositivos - Computadores

Notebooks: 1

(Arquitetura: Windows 7 Professional 64bits, 8gb RAM, I3 4ºgen, 500gb memória interna)

Servidores: Físico (1)

(Arquitetura: Windows Server 2019 64bits, 4gb RAM, I5 7ºgen, 500gb memória interna)

Hardware - Periféricos

Impressora: 2

Scanner: 1

Outros: 2 Telefones com ramal

Linhas de telefonia móvel: 7 (plano empresarial operadora Tim)

Licenças

Softwares: 2

Sistemas Operacionais: Windows Server (original), Windows 7 Professional (descontinuado).

Aplicativos: Andes Sistemas (ERP *online*), Nextar (ERP *offline*).

Recursos Humanos

Quantidade de Usuários: 7

Nível de Alfabetização digital: Conhecimento do uso essencial

Existe mudança constante de usuário: não

Configuração Mínima

Sistema operacional: Linux Server utilizando Apache.

Espaço mínimo em disco: 160 MB livres, recomendado 5GB de espaço.

Memória RAM: mínimo 256MB, recomendado 1 GB.

6.2 Métricas

Para gerenciar um processo é necessário medi-lo. Portanto, é necessário entender o que está acontecendo para definir parâmetros que possibilitem avaliar o processo.

Neste projeto foi utilizado o conjunto de parâmetros estabelecidos como Métricas Orientadas ao Tamanho, que são medidas diretas do *software* e do processo por meio do qual o *software* será desenvolvido. O Quadro 12 apresenta as métricas aplicadas para o desenvolvimento do sistema proposto neste documento.

Quadro 12 – Métricas

MÉTRICAS						
Função	Nº de ocorrência	Complexidade	Peso	Resultado	Nível de Influência do Sistema (0 a 5)	
entradas		Simple	3	0	Comunicação de dados	5
		Médio	4	0	Performance	5
	6	complexo	6	36	Volume de transações	5
saídas		Simple	4	0	Eficiência do usuário final	5
	1	Médio	5	5	Processamento complexo	5
	6	complexo	7	42	Facilidade de implantação	5
consultas		Simple	3	0	Múltiplos locais	5
	1	Médio	4	4	Processamento distribuído	5
	6	complexo	6	36	Utilização de equipamento	5
arquivos		Simple	7	0	Entrada de dados <i>online</i>	5
	6	Médio	10	60	Atualização <i>online</i>	5
		complexo	15	0	Reutilização de código	5
interfaces		Simple	5	0	Facilidade operacional	5
	7	Médio	7	49	Facilidade de mudanças	5
		complexo	10	0		
Total de FP'b				232	Total de NI	70
					FA = multiplicar o NI pela taxa real =0,65+(0,01*119)	1,35
					FP'r = multiplicar o FP'b pelo FA	313

		Informe o nº de LOC da Linguagem	20
		KLOC = Multiplicar o FP'r pelo tipo de linguagem	
Estimativas do número médio de LOC por FP		Total de KLOC	6264
Cobol	100		
Pascal	90		
Linguagens Orientadas a Objeto (C++)	30	Informe o tipo de sistema	
Java / Delphi / Visual Basic	20	3.300	
Geradores de Código (SQL + HTML)	15	PRAZO (dividir o KLOC pelo tipo de sistema)	
		(RESULTADO DA DIVISÃO) QTDD DE MESES	1,90
		QTDD DE DIAS	41,80
		QTDD DE HORAS	4,80
		QTDD DE MINUTOS	48
		CUSTO - INVESTIMENTO	
		Informe o valor da hora de trabalho	R\$ 18,00
		ISO (NORMA INTERNACIONAL) HORAS/MÊS =	132
		FÓRMULA = 132 * RESULTADO DA DIVISÃO * VALOR DA HORA	
		VALOR TOTAL DO PROJETO =	R\$ 4.510,08
		PROVA DE CONFIRMAÇÃO (DIAS * HORAS * VALOR DA HORA)	

PRAZO	
DIAS/MÊS	22
HORAS/DIA	6
MINUTOS/HORA	60

Tipo de Sistema	Produtividade - Kloc/Loc /mês
Sistema Comercial	2.500
Comércio Eletrônico	3.600
Sistema Web	3.300

Fonte: elaborado pelos autores.

6.3 Proposta Comercial

6.3.1 Introdução

O objetivo inicial é desenvolver um projeto de sistema para gerenciar o fluxo de treinamentos de processos e produtos da empresa para todos os seus funcionários, além de armazenar dados históricos sobre os treinamentos realizados pelos funcionários.

O treinamento e capacitação de funcionários é uma prática que permite o desenvolvimento contínuo dos colaboradores de uma empresa, impactando

diretamente na qualidade e eficiência dos processos, na relação com clientes e na lucratividade. Permite aperfeiçoar habilidades técnicas dos colaboradores de uma empresa, trazendo novas metodologias a serem implementadas nas rotinas diárias e na forma de resolver problemas.

6.3.2 Solução Proposta

Elaborar um sistema *web* para disponibilizar e permitir o gerenciamento dos materiais de treinamentos produzidos pela empresa, buscando flexibilizar, agilizar e padronizar a divulgação de novos produtos e procedimentos implementados pela empresa.

Possibilitar que os gerentes possam acompanhar o progresso dos treinamentos, responder dúvidas e disponibilizar certificados de conclusão para os funcionários.

A plataforma terá um *layout* intuitivo e responsivo para facilitar a realização dos treinamentos em vários tipos de dispositivos utilizados pelos funcionários.

6.3.3 Visão Geral da Solução

A plataforma deverá possuir as seguintes funcionalidades:

- Criação de tipos de usuários
- Cadastro de usuários
- Cadastro de treinamento e avaliações
- Armazenamento de arquivos de áudio, vídeo, formulários e texto
- Inscrição de funcionários nos treinamentos
- Notificação dos funcionários sobre treinamentos em que foram inscritos
- Gravar o progresso de cada funcionário em cada treinamento
- Possibilitar cadastro e envio de dúvidas pelos funcionários
- Possibilitar recurso destinado a resposta de dúvidas enviadas pelos funcionários
- Aplicação de avaliações sobre o conteúdo dos treinamentos através de formulários
- Emissão de certificados de conclusão para os funcionários aprovados nos treinamentos

6.3.4 Escopo da Solução

O projeto foi dividido em etapas conforme descrito a seguir:

- a) Iniciação: Elaborar questões de elicitação de requisitos, antecipando possíveis dificuldades que a empresa enfrenta. Realizar e gravar, com a permissão dos envolvidos, entrevistas com os *stakeholders* da empresa.
- b) Levantamento dos requisitos: Realizar etnografia no ambiente da empresa para visualizar a dinâmica dos processos realizados. Realizar brainstorming para coletar as expectativas dos *stakeholders* a respeito das funcionalidades que o sistema deve possuir.
- c) Criar protótipo do sistema: Seguir normas da engenharia de *software* para elaboração dos diagramas e documentações (BPMN, documentação de requisitos, diagrama de caso de uso, documentação de caso de uso, diagrama de classe, diagrama de sequência, diagrama de atividade, diagrama de máquina de estado, matriz de rastreabilidade), elaborar métricas para cálculo de prazo e custos de desenvolvimento.
- d) Implantação do sistema: Instalação dos módulos desenvolvidos, realizar treinamento dos usuários apoiando em sua adaptação de uso do sistema.

6.3.5 Prazos e Investimentos

Aplicadas as métricas, calculou-se os valores para o desenvolvimento do sistema, baseado no diagrama de classes e quantidade de tabelas da solução.

O custo total de desenvolvimento do sistema é de R\$ 4.510,08 (quatro mil, quinhentos e dez reais e oito centavos), sendo necessários 41 dias 4 horas e 48 minutos úteis para o desenvolvimento.

A entrega do *software* está prevista para 60 dias após o início do desenvolvimento e cada licença de uso terá um custo de R\$ 1.504,00 (um mil, quinhentos e quatro reais) divididos em doze parcelas no valor de R\$ 125,33 (cento e vinte e cinco reais e trinta e três centavos) cada, além de um contrato de manutenção com mensalidade no valor de R\$ 150,00 (cento e cinquenta reais), que incluirá atualizações de versão, suporte técnico limitado a assinatura para configurações e solução de problemas, acesso a um treinamento *online* sobre o sistema com vídeos, tutoriais, dicas e repostas de perguntas mais frequentes.

A manutenção do banco de dados utilizado pelo sistema será de responsabilidade da empresa contratante.

Considerações finais

A proposta inicial deste trabalho foi compreender os problemas e identificar os pontos fracos que afetavam o setor de vendas da empresa case. Para isso foram utilizados alguns métodos e ferramentas para analisar o atual cenário da empresa, identificar suas fraquezas, dentre elas escolher aquela que poderia ser utilizada como o escopo principal do projeto e partindo desta premissa propor uma solução sistêmica capaz de auxiliar a empresa na mitigação de tal problema.

Com base neste contexto, após a identificação de uma grande deficiência na capacitação profissional dos colaboradores, foi desenvolvido um projeto de *software* de treinamentos *online* que oferece um método de treinamento a distância, uma alternativa que possibilita a disponibilização de treinamentos através de diferentes tipos de arquivos e permite o acesso dos conteúdos através de vários tipos de dispositivos. O projeto também possui recursos que possibilitam um gerenciamento efetivo dos treinamentos por parte da gerência da empresa. Desta maneira, o resultado obtido pode ser considerado bastante satisfatório, pois foi possível alcançar o objetivo inicial do projeto, propondo uma alternativa que auxiliará a empresa na capacitação profissional de seus funcionários e trazendo uma solução eficaz para o problema em questão.

A realização deste trabalho nos possibilitou conhecer um pouco da história da empresa e seus processos na comercialização de seus produtos. Também nos proporcionou um grande conhecimento na utilização de técnicas e ferramentas de modelagem e desenvolvimento de *softwares*. Os principais desafios enfrentados foram na fase de coleta de informações com o cliente e na elaboração do BPMN devido à falta de padronização nos processos de venda da empresa.

Como trabalho futuro poderá ser realizado a implementação de recursos para a criação e edição de videoaulas integrado no sistema, como forma de facilitar a criação de conteúdo dos treinamentos. Também pode se pensar em uma funcionalidade de criação de relatórios com dados de desempenho e metas de aprendizagem dos funcionários da empresa.

Referências

- ALURA. **ORM com NodeJS: API com Sequelize e MySQL**, 2023. Disponível em: <https://www.alura.com.br/conteudo/orm-nodejs-api-sequelize-mysql>. Acesso em: 30 set. 2023.
- BESSA, André. Node.JS: o que é, como funciona esse ambiente de execução JavaScript e um Guia para iniciar. **Alura**, 18.set.2023. Disponível em: <https://site.alura.com.br/artigos/node-js>. Acesso em: 13 out. 2023.
- BITO.AI. **jsonwebtoken npm: npm explicado**, 2023. Disponível em: <https://bito.ai/resources/jsonwebtoken-npm-npm-explained/>. Acesso em: 14 out. 2023.
- CLEVER-CLOUD. **Por que Nuvem inteligente?**. Disponível em: <https://www.clever-cloud.com/presentation/>. Acesso em: 13 out. 2023.
- CORDEIRO, Luiz *et al.* **Modelo de desenvolvimento de negócios (canvas) da incubadora de empresas de base tecnológica da UFGD (GDTEC)**, 2017. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/181136/101_00249.pdf. Acesso em: 12 set. 2023.
- DEVMEDIA. **Elicitação de Requisitos**: Levantamento de requisitos e técnicas de Elicitação, 2014. Disponível em: <https://www.devmedia.com.br/elicitacao-de-requisitos-levantamento-de-requisitos-e-tecnicas-de-elicitacao/31872>. Acesso em: 9 set. 2023.
- EXPRESSJS. **Express**, 2023. Disponível em: <https://expressjs.com/>. Acesso em: 14 out. 2023.
- GASPAR, Larissa. Aprenda o que é NPM e como utilizá-lo. **Hostgator**, 19 fev. 2021. Disponível em: <https://www.hostgator.com.br/blog/o-que-e-npm/>. Acesso em: 14 out. 2023.
- GOMES, Gabrielle. Dotenv: gerenciando variáveis de ambiente. **Alura**, 20 jan. 2023. Disponível em: <https://www.alura.com.br/artigos/dotenv-gerenciando-variaveis-ambiente#:~:text=O%20Dotenv%20%C3%A9%20um%20pacote,separado%20do%20c%C3%B3digo%20da%20aplica%C3%A7%C3%A3o>. Acesso em: 14 out. 2023.
- HOSTGATOR. **Framework**: o que é, quais utilizar e como eles funcionam!, 9 jan. 2023. Disponível em: <https://www.hostgator.com.br/blog/frameworks-na-programacao/>. Acesso em: 30 set. 2023.
- IBM. **Diagramas de Caso de Uso**, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/engineering-lifecycle-management-suite/lifecycle-management/6.0?topic=cases-use>. Acesso em: 5 set. 2023.
- JOI. **API v17.9.x Introdução**, 2023. Disponível em: <https://joi.dev/api/?v=17.9.1#general-usage>. Acesso em: 14 out. 2023.

KERLYN. Uma breve introdução sobre BCrypt. **Medium**, 10 dez. 2019. Disponível em: <https://medium.com/reprogramabr/uma-breve-introdu%C3%A7%C3%A3o-sobre-bcrypt-f2fad91a7420>. Acesso em: 14 out. 2023.

LIMA, Guilherme. Saiba tudo sobre o IDE - Integrated Development Environment. **Alura**, 9 set. 2022. Disponível em: <https://www.alura.com.br/artigos/o-que-e-uma-ide#editor-de-codigo-e-ide>. Acesso em: 30 set. 2023.

LOURENÇO, Hugo. Termo de Abertura do Projeto: o que é, como funciona e como podemos utilizá-lo na agilidade. **Objective**, 10 out. 2021. Disponível em: <https://www.objective.com.br/insights/termo-de-abertura-do-projeto-o-que-e-como-funciona-e-como-podemos-utiliza-lo-na-agilidade/>. Acesso em: 11 out. 2023.

LUCIDCHART. **O que é um Diagrama Entidade Relacionamento?**, 2023. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>. Acesso em: 5 set. 2023.

MATIAS, Sanon. **Canvas: O Que É, Como Funciona E Vantagens**, 2023. Disponível em: <https://www.webmaissistemas.com.br/blog/canvas-o-que-e/>. Acesso em: 12 set. 2023.

MIRO. **Diagrama de Entidade e Relacionamento**, 2023. Disponível em: <https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/>. Acesso em: 5 set. 2023.

MOZILLA. **Introdução Express/Node**, 3 ago. 2023. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction. Acesso em: 30 set. 2023.

MUI. **Mova-se mais rápido com as ferramentas intuitivas do React UI**, 2023. Disponível em: <https://mui.com/>. Acesso em: 14 out. 2023.

NEVES, Vinícios. React: o que é, como funciona e um guia dessa popular ferramenta JS. **Alura**, 17 jan. 2023. Disponível em: <https://www.alura.com.br/artigos/react-js>. Acesso em: 14 out. 2023.

NODEJS. **Sobre Node.js**. Disponível em: <https://nodejs.org/pt-br/about>. Acesso em: 13 out. 2023.

NPMJS. **node-postgres**, 2023. Disponível em: <https://www.npmjs.com/package/pg>. Acesso em: 14 out. 2023.

NPMJS. **pg-hstore**, 2021. Disponível em: <https://www.npmjs.com/package/pg-hstore>. Acesso em: 14 out. 2023.

OMG [Object Management Group]. **Business Process Model and Notation (BPMN): Version 2.0**. OMG, 2011.

ORACLE. **O que é um Banco de Dados?**, 2023. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 30 set. 2023.

PEDRA, David. **Matriz SWOT: entenda como enriquecer seu planejamento**, 2023. Disponível em: <https://www.siteware.com.br/metodologias/Planejamento-estrategico-analise-matriz-swot/>. Acesso em: 14 set. 2023.

PIERRI, Renato. Qual é a diferença entre requisitos funcionais e requisitos não funcionais na engenharia de *software*? **Nets & Nuts**, 1 nov. 2019. Disponível em: <https://nets-nuts.com.br/qual-e-a-diferenca-entre-requisitos-funcionais-e-requisitos-nao-funcionais-na-engenharia-de-software/>. Acesso em: 1 set. 2023.

POSTGRESQL. **Sobre**. Disponível em: <https://www.postgresql.org/about/>. Acesso em: 13 out. 2023.

PROJ4ME, **Saiba o que é e para que serve o Termo de Abertura do Projeto (TAP)**, 2022. Disponível em: <https://www.proj4.me/blog/termo-de-abertura-do-projeto>. Acesso em: 11 out. 2023.

RABELLO, Guilherme. **Modelo Canvas: saiba o que é, para que serve e como fazer!**, 01.set.2023. Disponível em: <https://www.siteware.com.br/metodologias/modelo-canvas/>. Acesso em: 12 set. 2023.

REACT. **Create user interfaces from components**, 2023. Disponível em: <https://react.dev/>. Acesso em: 14 out. 2023.

SEBRAE-SC. **5W2H: o que é, para que serve e por que usar na sua empresa**, 30 dez. 2022. Disponível em: <https://www.sebrae-sc.com.br/blog/5w2h-o-que-e-para-que-serve-e-por-que-usar-na-sua-empresa>. Acesso em: 12 set. 2023.

SEGUINS, Neilton. Utilizando o Vite para criar uma aplicação React com TypeScript. **Alura**, 19 set. 2023. Disponível em: <https://www.alura.com.br/artigos/vite-criar-aplicacao-react-typescript>. Acesso em: 14 out. 2023.

SEQUELIZE. **Sequelize**, 2023. Disponível em: <https://sequelize.org/>. Acesso em: 14 out. 2023.

SILVER, Bruce. **BPMN Method And Style: With BPMN Implementer's Guide**. 2 ed. Altadena: Cody-Cassidy Press, 2019.

UCJ [UFMG CONSULTORIA JÚNIOR]. **Análise de Viabilidade de Projetos: saiba como elaborá-lo e aplicá-lo no seu negócio!**, 2021. Disponível em: <https://ucj.com.br/blog/analise-de-viabilidade-de-projetos/>. Acesso em: 12 set. 2023.

VITE. **Começar**, 2023. Disponível em: <https://pt.vitejs.dev/guide/features.html>. Acesso em: 30 set. 2023.