**SENAC - Criciúma**

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

**Bruno Albuquerque Vaz Jucoski**

**Leonardo Rodrigues da Costa**

**Patrick Fermino**

**Mateus Pires**

**Luiz Gustavo de Moraes José**

**APPOLO:**

Conectando artistas

**Criciúma**

**2025**

**Bruno Albuquerque Vaz Jucoski**

**Leonardo Rodrigues da Costa**

**Patrick Fermino**

**Mateus Pires**

**Luiz Gustavo de Moraes José**

**APPOLO:**

Conectando artistas

Trabalho apresentado à Faculdade Senac de Criciúma como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Lucas Bonfante Rabelo

**Criciúma**

**2025**

Ficha de identificação da obra

|  |
| --- |
|  |
| A ficha catalográfica deve ser elaborada pelo bibliotecário, conforme o Código de Catalogação Anglo-Americano |
|  |
|  |
|  |
|  |
|  |

**Bruno Albuquerque Vaz Jucoski**

**Leonardo Rodrigues da Costa**

**Patrick Fermino**

**Mateus Pires**

**Luiz Gustavo de Moraes José**

**APPOLO:**

Conectando artistas

Trabalho apresentado à Faculdade Senac de Criciúma como requisito parcial para obtenção do título Análise e Desenvolvimento de Sistemas.

Orientador: Nome completo do orientador

**Banca:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lucas Bonfante Rebelo (Orientador)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roberto Fermino Medeiros - Senac

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dayana Gomes Ricken Miranda- Senac

Criciúma, 03 junho de 2025

**RESUMO**

As dificuldades enfrentadas por artistas em geral no cenário profissional, como a informalidade, baixa visibilidade e poucas oportunidades, fomentam um ambiente profissional muito inseguro para quem busca construir uma carreira nesse meio. A partir desses preceitos trouxemos a proposta de conectar artistas a potenciais contratantes, promovendo a cultura e desenvolvendo o crescimento econômico de forma inclusiva e sustentável. Por meio de pesquisas e análises sobre o contexto sociocultural, aplicamos questionários com perguntas fechadas e abertas, identificou-se a ausência de ferramentas digitais específicas que dêem suporte a esse público, como ocorre em outras áreas profissionais. Com base nessas percepções, elaboramos um sistema com recursos como perfis sob medida, categorização dos meios artísticos, filtro de pesquisa por região, e ainda a opção de avaliação de ambas partes envolvidas para fortalecer a credibilidade. A *interface* foi pensada para ser simples e fácil de usar, servindo tanto artistas menos acostumados com tecnologia, quanto pessoas que querem algo prático. Dessa forma, o projeto concretiza a meta principal de desenvolver uma ferramenta digital eficaz, enquanto simultaneamente alcança propósitos focados em ampliar o acesso do público geral a experiências culturais e à redução dos problemas encarados pelo setor artístico.

Palavras-chave: Artistas locais. Cultura. Empregabilidade.

**ABSTRACT**

The difficulties faced by artists in general in the professional scene, such as informality, low visibility and few opportunities, foster a very insecure professional environment for those seeking to build a career in this field. Based on these principles, we proposed connecting artists with potential employers, promoting culture and developing economic growth in an inclusive and sustainable way. Through research and analysis on the sociocultural context, we applied questionnaires with closed and open questions, and identified the lack of specific digital tools that support this audience, as occurs in other professional areas. Based on these perceptions, we developed a system with resources such as customized profiles, categorization of artistic media, search filter by region, and even the option for evaluation by both parties involved to strengthen credibility. The interface was designed to be simple and easy to use, serving both artists less familiar with technology and people who want something practical. In this way, the project achieves the main goal of developing an effective digital tool, while simultaneously achieving purposes focused on expanding the general public's access to cultural experiences and reducing the problems faced by the artistic sector.

Keywords: Local artists. Culture. Employability.

**LISTAS DE ILUSTRAÇÕES**

[**Figura 1 – *Project Model Canvas* – PM Canvas. 47**](#_sy4tequfctgt)

[**Figura 2 - Diagrama de caso de uso (Resumo). 51**](#_n5ie47y2aphh)

[**Figura 3 - Diagrama de caso de uso (Realizar *login*) 52**](#_n5ie47y2aphh)

[**Figura 4 - Diagrama de caso de uso (Interações do artista) 53**](#_n5ie47y2aphh)

[**Figura 5 - Diagrama entidades relacionamentos 54**](#_cdbdtbn6pggd)

[**Figura 6 - Fluxo para contratação de um artista 55**](#_a91jku3qv0ga)

[**Gráfico 1 - Pergunta 1 56**](#_zc8rzern9cbd)

[**Gráfico 2 - Pergunta 2 57**](#_cjvo9n1dn7ro)

[**Gráfico 3 - Pergunta 3 57**](#_4u34oimi8ir8)

[**Gráfico 4 - Pergunta 4 58**](#_f7den5x5w7jp)

[**Gráfico 5 - Pergunta 5 58**](#_99u9ed54hbb1)

[**Gráfico 6 - Pergunta 6 59**](#_emj8wqhlv63i)

[**Gráfico 7 - Pergunta 7 59**](#_m1e0w4jgnl7l)

[**Gráfico 8 - Pergunta 8 60**](#_auwy6aec6iwg)

[**Gráfico 9 - Pergunta 9 60**](#_ii9i2gfd1jcb)

[**Gráfico 10 - Pergunta 10 61**](#_4gaypav4d6s)

[**Gráfico 11 - Pergunta 11 61**](#_scp915b0tkdw)

[**Gráfico 12 - Pergunta 12 62**](#_gdwj9g95jlet)

[**Gráfico 13 - Pergunta 13 62**](#_sswz7qeeurrf)

**LISTA DE SIGLAS**

MEI – Microempreendedor Individual  
OOA – Análise Orientada a Objetos  
ODS – Objetivos de Desenvolvimento Sustentável  
ONU – Organização das Nações Unidas  
ER – Entidade-Relacionamento  
MCU – Modelo de Casos de Uso  
DCU – Diagrama de Casos de Uso  
UML – Linguagem de Modelagem Unificada  
BPMN – Notação de Modelagem de Processos de Negócio  
ABPMP – Association of Business Process Management Professionals  
PCO – Planejamento e Controle da Produção  
HTTP – Protocolo de Transferência de Hipertexto  
CERN – Organização Europeia para a Pesquisa Nuclear  
PHP – Hypertext Preprocessor  
FI – Forms Interpreter  
PDF – Formato Portátil de Documento  
XML – Linguagem de Marcação Extensível  
SGBD – Sistema de Gerenciamento de Banco de Dados  
SQL – Linguagem de Consulta Estruturada  
DER – Diagrama Entidade-Relacionamento  
CNPJ – Cadastro Nacional da Pessoa Jurídica  
CEP – Código de Endereçamento Post

**SUMÁRIO**

[**1 INTRODUÇÃO 8**](#_rmz5gdg5bitj)

[1.1 OBJETIVOS 9](#_7006914aq9zs)

[**1.1.1 Objetivo geral 9**](#_znr7m1qhi0w)

[**1.1.2 Objetivos específicos 9**](#_wdmlfwwqcj50)

[**2 REVISÃO DE LITERATURA 10**](#_tyjcwt)

[2.1 RAMO CULTURAL 10](#_y994ztb5ylf7)

[2.2 OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL 10](#_bztigutw72da)

[2.3 OBJETIVO DE DESENVOLVIMENTO SUSTENTÁVEL DE ATUAÇÃO 11](#_c8awrk9ub9ex)

[2.4 ARTISTAS MODERNOS (MEI) 11](#_2meawmjg10ik)

[2.5 ANÁLISE ORIENTADA A OBJETOS 12](#_b8ngp0isqx4t)

[2.6 SOLICITAÇÃO DE SERVIÇOS 12](#_vp9x6dsuvuwe)

[2.7 SISTEMA DE SOLICITAÇÃO DE SERVIÇOS 13](#_yt5rcx8gz0vg)

[**3 GERENCIAMENTO DE PROJETOS 14**](#_f8u9dfn2w61u)

[3.1 PROTOTIPAÇÃO 15](#_tou6levax9uh)

[3.2 MODELO ENTIDADE-RELACIONAMENTO 15](#_f0g4elty2s54)

[3.3 DIAGRAMA DE CASO DE USO 16](#_7upugcbpsyrk)

[3.5 *PROJECT MODEL CANVAS* 17](#_ktknx3vxspkp)

[3.6 OS 5 PORQUÊS 18](#_onm8mq3w0tgl)

[3.7 *BUSINESS PROCESS MODELING NOTATION* 18](#_b1d3qt3w9fpd)

[**4 ENGENHARIA DE *SOFTWARE* 20**](#_kpczhyny3gr4)

[4.1 PROGRAMAÇÃO ORIENTADA A OBJETOS 21](#_kpuyw5udqh9l)

[4.2 *WEB* 22](#_b7om81ww612s)

[4.3 LINGUAGENS DE PROGRAMAÇÃO 22](#_iwpfvh1yyjnj)

[4.3.1 *Model-View-Controller* (MVC) 23](#_4ezwt3upwl6c)

[4.3.2 *Personal Home Page/Forms Interpreter* (PHP) 24](#_pz39hclvolbi)

[4.3.3 *Laravel* 24](#_xe9a5wy7deg5)

[4.3.4 *Javascript* 25](#_xwh6hj2emvi9)

[**5 SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS 27**](#_witnw3xxys56)

[5.1 *STRUCTURED QUERY LANGUAGE* 28](#_d2cnhraxnsz5)

[**6 METODOLOGIA 29**](#_3dy6vkm)

[6.1 CARACTERIZAÇÃO DA PESQUISA 30](#_jhno9jhorrb4)

[6.2 TÉCNICAS DE COLETA DE DADOS 31](#_hcjwwwfju4na)

[6.3 FORMA DE ANÁLISE DE DADOS 31](#_4mmi6xxy92ib)

[**7 CONTEXTUALIZAÇÃO E RESULTADOS 33**](#_glk6z0wurguy)

[7.1 CONTEXTUALIZAÇÃO 33](#_845okkg7043i)

[7.2 RESULTADOS 34](#_9cebukvtfza8)

[**REFERÊNCIAS 36**](#_iheyuo3femmd)

[APÊNDICE A – Identificação da causa raiz do problema 45](#_bfosm49k74vf)

[APÊNDICE B – Estruturação do projeto 46](#_lthoi4ipm8zz)

[APÊNDICE C – Requisitos funcionais e não funcionais 47](#_1i3rsucnpudj)

[APÊNDICE D – Diagrama de caso de Uso 51](#_z8le7u8pqy5l)

[APÊNDICE E – Modelo E&R Lógico - DER 54](#_9o3x7iyzggm2)

[APÊNDICE F – Fluxo BPMN 55](#_kgh4rnayfrdl)

[APÊNDICE G – Questionário 56](#_rkvun1dbldrq)

## 

## 

## 

## 

## 

## 

## 1 INTRODUÇÃO

O meio cultural remete ao conjunto de valores, crenças, tradições, práticas artísticas, linguagem que caracterizam uma determinada sociedade ou grupo. Todos esses aspectos são importantes para assegurar um ambiente rico na formação de uma sociedade coesa que promove a diversidade e preservação de patrimônios culturais, influenciando a maneira como as pessoas vislumbram o mundo, se relacionam entre si e com o meio ambiente.

Muitos profissionais do ambiente cultural têm grande dificuldade de encontrar oportunidades de trabalho atualmente, por conta de uma alta informalidade para desenvolver-se portfólios, a maioria dos artistas locais depende da divulgação em redes sociais para ascenderem profissionalmente e conseguir fechar contratos para garantir o seu sustento. Carecendo de uma plataforma que conecte contratantes com os serviços ofertados pelos artistas locais, a defasagem do meio segue crescente, sendo que artistas iniciantes acabam desistindo de seguir carreira nos primeiros anos de profissão. Além disso, muitas empresas e cidadãos têm dificuldade de encontrar os profissionais, recorrendo comumente a indicações para buscar os artistas que desejam contratar, muitas vezes não sendo encontrados.  
 Os artistas do meio cultural têm a necessidade de encontrar oportunidades de um trabalho digno, carecem de uma plataforma como *linkedin* que conecta empresas com profissionais qualificados. O meio cultural divide-se em múltiplos segmentos como por exemplo teatro, música e artes visuais.

Visando oferecer um trabalho decente e crescimento econômico conforme a ODS 8, é essencial criar meios que valorizem e interajam com artistas informais, como os artistas de baixa fidelidade ao mercado formal.

A finalidade é criar uma plataforma que valorize e conecte artistas locais com públicos interessados, oferecendo visibilidade e oportunidades para divulgação de artistas, permitindo configurar portfólio, endereço e categoria de atuação (artes visuais, teatro, entre outras), cadastro de artistas e de consumidores.

Usuários podem buscar os artistas que desejam, podendo filtrar por categoria, localidade e outras características desejadas. Desenvolver *feedbacks* e avaliações na plataforma, destacando os artistas mais bem avaliados e promovendo a conexão entre consumidores e artistas.

Por fim, a plataforma visa atuar facilitando a empregabilidade no meio artístico, gerando desenvolvimento econômico notável num meio que se encontra fragmentado, diminuindo a informalidade e promovendo o contato entre a população que deseja contratar artistas qualificados com os mesmos.

## 1.1 OBJETIVOS

## 1.1.1 Objetivo geral

Desenvolver uma plataforma *web* para conectar e divulgar artistas locais ao público em âmbito nacional.

## 1.1.2 Objetivos específicos

* Identificar a causa raiz do problema, validando a necessidade junto aos *stakeholders*;
* Elicitar e documentar os requisitos funcionais e não funcionais;
* Desenvolver a análise do sistema e banco de dados;
* Apresentar um protótipo de alta fidelidade para o *software*;
* Realizar testes de unidade para garantir a correta implementação do sistema;

## 2 REVISÃO DE LITERATURA

## 

Neste capítulo será abordado sobre o objeto de estudo deste trabalho, o ramo cultural, microempreendedores individuais, gerenciamento de portfólio, solicitação de serviços, sistema de solicitação de serviço e logo abaixo as ferramentas de análises usadas para construir o nosso projeto.

## 2.1 RAMO CULTURAL

Bourdieu (1984) descreve o ramo cultural como um campo de lutas simbólicas, onde se negociam legitimidade, poder e reconhecimento das práticas criativas. Hall (1966) por sua vez, define o ramo cultural como o espaço onde as experiências humanas são organizadas por meio de símbolos, valores e práticas compartilhadas. Geertz (1973) reforça que o ramo cultural funciona como uma teia de significados, na qual as práticas artísticas refletem e reinterpretam contextos sociais. Contudo Hesmondhalgh (2007) caracteriza o ramo cultural como um setor que combina produção econômica e simbólica, marcado por tensões entre criatividade e comercialização.

## 2.2 ARTISTAS MODERNOS (MEI)

De acordo com Canclini (1990), os artistas modernos operam em um contexto de hibridismo, combinando influências globais e locais para criar formas inovadoras de expressão. Já Banks (2010) afirma que os artistas modernos se caracterizam pela autonomia criativa, mas enfrentam desafios como a instabilidade financeira e a precariedade no mercado cultural. Appadurai (1996) destaca que os artistas modernos aproveitam a globalização para utilizar plataformas digitais, alcançando públicos diversos e ampliando o impacto de suas criações. Florida (2002) argumenta que os artistas modernos, como parte da classe criativa, transformam ideias em produtos e serviços inovadores, impulsionando a economia cultural.

## 

## 2.3 ANÁLISE ORIENTADA A OBJETOS

## 

A análise orientada a objetos (OOA) constitui uma metodologia de modelagem que prioriza a definição de objetos, suas classes e inter-relações como componentes essenciais na análise de sistemas complexos. Segundo Pereira (2018), esse tipo de modelagem engloba a construção de diagramas como os de classes e de sequência os quais ajudam a representar a estrutura e o comportamento dos sistemas, favorecendo a comunicação entre os envolvidos no processo de desenvolvimento.

Santos (2019) destaca que a OOA permite uma modelagem mais aderente ao contexto do problema, o que facilita a interpretação dos requisitos por parte dos analistas e desenvolvedores. Já Pressman (2015) afirma que a abordagem orientada a objetos no desenvolvimento de *software* foca na identificação e modelagem de entidades (objetos) que representam elementos do mundo real.

Larman (2007) complementa que a OOA utiliza princípios da orientação a objetos para capturar e refinar os requisitos, promovendo um alinhamento natural entre análise e *design*. Além disso, Booch et al. (2007) salientam que a modelagem orientada a objetos oferece uma base sólida para representar abstrações do domínio do problema, tornando o desenvolvimento mais intuitivo e sustentável.

## 2.4 SOLICITAÇÃO DE SERVIÇOS

De acordo com Fitzsimmons e Fitzsimmons (2006), a solicitação de serviço é o ponto inicial do processo de atendimento, onde as necessidades do cliente são formalizadas para garantir clareza e alinhamento com os objetivos do serviço. Zeithaml et al. (2009) destacam que a solicitação de serviço atua como uma interface crítica entre o cliente e o provedor, exigindo comunicação eficaz para captar expectativas e definir escopo.

Slack et al. (2010) reforçam que a solicitação de serviço é essencial para o planejamento operacional, permitindo que recursos sejam alocados com base em prioridades e prazos estabelecidos. Johnston e Clark (2008) argumentam que a solicitação de serviço deve seguir processos padronizados para reduzir ambiguidades, garantindo que as demandas sejam compreendidas e atendidas com eficiência.

Kerzner (2013) caracteriza a solicitação de serviço como um componente chave na gestão de projetos, pois define os requisitos iniciais que orientam o planejamento e a execução das atividades. Davenport (2005) destaca que a solicitação de serviço, quando apoiada por ferramentas digitais, aumenta a eficiência, permitindo rastreamento e respostas rápidas às necessidades dos clientes.

Voss e Tsikriktsis (2007) argumentam que um sistema de solicitação de serviços eficaz proporciona rastreabilidade, permitindo que clientes e gestores monitorem o progresso das demandas em tempo real.

## 2.5 SISTEMA DE SOLICITAÇÃO DE SERVIÇOS

De acordo com Grönroos (2007), o sistema de solicitação de serviços é um mecanismo estruturado que formaliza as demandas dos clientes, garantindo que sejam processadas com clareza e precisão.

Kotler e Keller (2012) destacam que um sistema de solicitação de serviços bem projetado melhora a experiência do usuário, oferecendo *interfaces* intuitivas para registrar e acompanhar solicitações.

Laudon e Laudon (2014) definem o sistema de solicitação de serviços como uma ferramenta tecnológica que automatiza o fluxo de demandas, reduzindo erros e agilizando o atendimento.

Heizer e Render (2011) reforçam que sistemas de solicitação de serviços integram diferentes áreas da organização, alinhando recursos e prioridades para atender às necessidades dos clientes.

Berry (1995) afirma que sistemas de solicitação de serviços contribuem para a gestão de qualidade, ao captar *feedback* diretamente das demandas, que pode ser usado para aprimorar processos, Berry (1995) enfatiza que o sistema de solicitação de serviços deve captar *feedback* implícito nas demandas dos clientes, transformando-o em oportunidades para aprimorar a experiência do serviço.

## 2.6 OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL

Os Objetivos de Desenvolvimento Sustentável (ODS) foram instituídos pela Organização das Nações Unidas (ONU) em 2015 como um compromisso global para erradicar a pobreza, proteger o meio ambiente e promover a prosperidade até 2030. De acordo com Ferreira e Boechat (2014), os ODS representam uma tentativa de construir metas mensuráveis capazes de eliminar a pobreza no contexto do desenvolvimento sustentável, abordando desafios sociais, econômicos e ambientais de maneira integrada. Já Trevisam e Cruciol Júnior (2022) destacam que a implementação dos ODS é essencial para assegurar a continuidade da vida no planeta, sendo um instrumento de efetivação dos direitos humanos em sua totalidade. A própria ONU (2015) ressalta que os ODS mesclam de forma equilibrada as três dimensões do desenvolvimento sustentável, econômica, social e ambiental, buscando resultados amplos e interdependentes. Para Buss et al. (2012), a Agenda 2030 estabelece uma nova lógica de ação internacional que conecta saúde, justiça social e proteção ambiental como bases para o progresso sustentável e inclusivo.

## 2.7 OBJETIVO DE DESENVOLVIMENTO SUSTENTÁVEL DE ATUAÇÃO

ONU (2015) diz que a ODS 8 busca promover o crescimento econômico sustentado, inclusivo e sustentável, o emprego pleno e produtivo, e o trabalho decente para todos. Isto implica melhorar a produtividade, promover a inovação e fomentar políticas públicas que incentivem a criação de empregos dignos. Sachs, (2015) diz que a promoção do trabalho decente está no coração da ODS 8, pois visa garantir condições dignas de trabalho e a inclusão social de todos, com ênfase no combate às desigualdades de gênero, raciais e sociais, a mesma complementa dizendo A ODS 8 visa não só melhorar as condições de trabalho, mas também promover uma transição para uma economia de baixo carbono, incentivando a sustentabilidade e a responsabilidade social nas empresas e nas práticas de trabalho. Buss (2012) complementa a ODS 8 requer políticas integradas que incentivem a inovação tecnológica, o empreendedorismo e a sustentabilidade nos setores produtivos.

## 3 GERENCIAMENTO DE PROJETOS

O início de um projeto de sistema parte de uma necessidade de negócio (Martins, 2007), o que destaca a relevância do gerenciamento de projetos na implantação de *softwares.*

Segundo Sommerville (2018), esse gerenciamento é essencial, já que projetos de *software* enfrentam limitações de tempo e orçamento.

Para apoiar esse processo, Finocchi Júnior (2013) criou o *Project Model Canvas*, uma abordagem ágil que permite o planejamento colaborativo, facilitando a compreensão das áreas do projeto. Complementando, Pinheiro et al. (2015) descrevem o *Canvas* como uma técnica visual que resume o modelo de negócio em uma única página.

Reis (2018) reforça a importância do uso de conhecimentos, habilidades e ferramentas no gerenciamento. Oliveira (2022) define essa prática como um conjunto de ações focadas no planejamento, execução e controle dos projetos, visando seu sucesso. Da mesma forma, Catarino (2020) afirma que a gestão de projetos é essencial para garantir prazos, custos e requisitos. Silva (2021) acrescenta que se trata de um processo sistemático, que contribui para a conclusão eficaz dos projetos.

De acordo com Mendes (2015), um projeto é um esforço com duração limitada voltado à criação de um produto ou serviço. Espinha (2023) amplia esse conceito ao afirmar que a gestão de projetos envolve metodologias e ferramentas que asseguram o cumprimento de prazos, custos e qualidade, mantendo o alinhamento com os objetivos estabelecidos. Para ele, essa prática é vital para coordenar ações de forma eficiente e gerar resultados positivos dentro da organização.

Maximiano (2021) enxerga o gerenciamento de projetos como uma disciplina que transforma ideias em resultados concretos, exigindo planejamento e controle eficazes para atingir metas estratégicas. Já Pressman (2021) ressalta a importância de conhecer os possíveis erros durante a execução de um projeto de *software*, a fim de preveni-los e garantir o sucesso do desenvolvimento.

## 3.1 PROTOTIPAÇÃO

Segundo Sommerville e Sawyer (1997), os protótipos servem como meio de comunicação eficaz entre os membros da equipe.

Silva (2022) entende o protótipo como um modelo inicial usado para testar funcionalidades ou ideias, podendo ser elaborado com materiais físicos ou ferramentas digitais.

Bardin (2021) complementa ao afirmar que protótipos são úteis em todas as fases do desenvolvimento, permitindo testes e coleta de *feedbacks*. Para Ferreira (2020), essa abordagem ajuda a reduzir custos e riscos ao possibilitar a identificação de falhas antes do investimento total no desenvolvimento.

Com a prototipação de alta fidelidade, o *software Figma* foi utilizado por sua praticidade de funcionamento direto no navegador. Martin e Bierly (2019) reforçam a eficiência do *Figma* para a criação de interfaces de alta qualidade.

O termo “protótipo” vem do grego “*protos*” (primeiro) e “*typos*” (modelo), significando o primeiro modelo de um produto (Vale, 2020).

Santos (2024) define os protótipos como representações visuais que simulam as telas de uma aplicação, permitindo ao usuário visualizar o sistema antes mesmo de ele ser construído.

De acordo com Pressman (2021), os protótipos auxiliam na identificação dos requisitos do *software*. O autor também explica que, no paradigma da prototipação, após a definição dos objetivos do *software*, é criado um projeto rápido, gerando um protótipo que será avaliado pelos envolvidos, possibilitando ajustes nos requisitos conforme o *feedback* recebido.

## 3.2 MODELO ENTIDADE-RELACIONAMENTO

Segundo Heuser (2010), o modelo Entidade-Relacionamento é fundamental na modelagem conceitual, pois permite representar graficamente entidades, atributos e relacionamentos, o que facilita a compreensão do domínio de dados antes de sua implementação no banco de dados.

De acordo com Silberschatz, Korth e Sudarshan (2006), o modelo ER proporciona uma visão de alto nível dos dados e seus relacionamentos, servindo como ponto de partida para o projeto de banco de dados relacional.

Elmasri e Navathe (2005) afirmam que o modelo Entidade-Relacionamento oferece uma abordagem intuitiva para capturar os requisitos informacionais de uma organização, sendo essencial para gerar um esquema lógico bem estruturado.

Conforme Pressman (2021), os diagramas ER são amplamente utilizados na análise de requisitos, contribuindo para a modelagem das informações que o sistema precisa manipular e auxiliando no processo de engenharia de *software*.

## 3.3 DIAGRAMA DE CASO DE USO

Após a elicitação dos requisitos, organizados por tipo, foi necessário estruturá-los em casos de uso, conforme orienta Martin (2018), que define esse tipo de diagrama como uma forma de representar como o sistema será utilizado, identificando as entradas fornecidas pelos usuários. Sommerville (2018) acrescenta que esse diagrama mostra as interações entre o sistema e seu ambiente. Assim, os diagramas de caso de uso foram utilizados tanto para visualizar o comportamento do sistema quanto como apoio à equipe no desenvolvimento, validação e implementação das funcionalidades.

Segundo Soares (2022), o diagrama de caso de uso é uma ferramenta visual essencial para representar as interações entre usuários e sistemas.

Wazlawick (2011) enfatiza que sua elaboração permite identificar consultas e transformações importantes, colaborando para a organização dos requisitos.

Oliveira (2022) aponta que esses diagramas favorecem a comunicação entre os membros da equipe, facilitando o desenvolvimento.

Saliba Júnior (2020) reforça que o diagrama de caso de uso é eficaz para definir e documentar os serviços do sistema.

Bezerra (2020) explica que o Modelo de Casos de Uso (MCU) representa as funcionalidades observáveis do sistema e os agentes que interagem com ele. Cada caso de uso descreve uma sequência completa de interações e, conforme a complexidade do sistema, o número de casos pode variar.

O Diagrama de Casos de Uso (DCU) é um dos componentes da UML, oferecendo uma visão externa e abstrata do sistema. Ele representa graficamente os atores (com a figura de um boneco), os casos de uso (em elipses) e as conexões entre eles (linhas retas), funcionando como uma espécie de “diagrama de contexto” (Bezerra, 2020).

## 3.5 *PROJECT MODEL CANVAS*

O *Project Model Canvas*, criado por José Finocchio Júnior, é um modelo que visa simplificar o planejamento de projetos, condensando os principais elementos em um formato visual e acessível (FINOCCHIO JÚNIOR, 2013).

Segundo Prikladnicki, Graeml e Soares (2020), o *PM Canvas* pode ser particularmente útil em projetos que exigem adaptação rápida, pois oferece uma visão geral e integrada das dimensões essenciais do projeto, sem a rigidez dos modelos tradicionais.

Além disso, a ferramenta valoriza a colaboração e o entendimento mútuo entre *stakeholders*, substituindo documentos extensos por uma representação mais visual e direta dos objetivos e ações do projeto (FINOCCHIO JÚNIOR, 2013).

*“O Project Model Canvas* é uma ferramenta visual de planejamento de projetos que permite, em uma única página, alinhar todos os envolvidos em relação ao escopo, cronograma, riscos, recursos e entregas do projeto.”  
 (FINOCCHIO JÚNIOR, 2013, p. 19).

De acordo com Barcaui (2014), o *PM Canvas* funciona como uma ponte entre abordagens ágeis e tradicionais, proporcionando uma estrutura visual que favorece a adaptação dos processos às necessidades específicas de cada projeto.

## 3.6 OS 5 PORQUÊS

O método dos 5 porquês foi desenvolvido no Sistema *Toyota* de Produção como uma ferramenta prática para identificar causas raízes de falhas, promovendo uma cultura de resolução estruturada de problemas (OHNO, 1997).

“Para encontrar a verdadeira causa de um problema, você deve perguntar ‘por quê? ’ cinco vezes” (RIES, 2012, p. 128).

De acordo com Jeffrey Liker (2005), o uso dos cinco porquês é parte do pensamento sistêmico da *Toyota*, sendo uma prática incorporada no processo decisório diário.

“A técnica dos 5 porquês força a equipe a olhar além dos sintomas e resolver o verdadeiro problema” (GEORGE et al., 2005, p. 44).

Vicente Falconi (1996) destaca que a análise de causas deve ser sistemática e ir além da superfície do problema, sendo os 5 porquês uma ferramenta eficiente nesse processo.

## 3.7 *BUSINESS PROCESS MODELING NOTATION*

Para Silver (2011), o valor doBPMNestá não apenas na notação em si, mas na aplicação de estilo e padronização para garantir clareza nos modelos.

“*BPMN* é simples o suficiente para ser entendida por analistas de negócios e técnicos, ao mesmo tempo em que fornece a precisão necessária para a automação de processos” (FREUND; RÜCKER, 2015, p. 17).

“A principal vantagem do BPMN é permitir que diferentes *stakeholders* compreendam e validem os processos de negócio de maneira padronizada” (PESSOA, 2017, p. 24).

O guia da ABPMP (2013) reforça o papel do BPMN como facilitador da comunicação e padronização na modelagem de processos.

*“* BPMN é a linguagem ou notação de modelagem de processos mais amplamente utilizada. É fácil de aprender, é aplicável às mais diversas áreas de atuação das organizações – e melhor ainda, é um padrão aberto” (CAMPOS, 2014, p. 14).

Campos (2014) destaca que a modelagem de processos é essencial para o autoconhecimento organizacional, permitindo que as empresas profissionalizem seus métodos de trabalho, aumentem o sucesso de seus projetos e melhorem a eficiência operacional.

## 

## 3.8 TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

"A elicitação dos requisitos envolve encontros com *stakeholders* de diferentes tipos para descobrir informações sobre o sistema proposto" (SOMMERVILLE, 2019, p. 131).

A elicitação de requisitos depende de reuniões com *stakeholders*, sendo fundamental compreender como eles trabalham e os sistemas que utilizam atualmente, o que pode indicar a necessidade de adaptações no novo sistema (SOMMERVILLE, 2019).

"Diversas técnicas podem ser utilizadas para a elicitação de requisitos, incluindo entrevistas com usuários, grupos de discussão, questionários, e técnicas baseadas em modelos como a prototipagem" (PRESSMAN, 2016, p. 158).

"As técnicas de levantamento de requisitos podem variar desde simples entrevistas e questionários até métodos mais estruturados, como *workshops* e análise de documentos" (PFLEEGER; ATLEE, 2010, p. 150).

## 

## 3.9 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

"Requisitos funcionais descrevem o que o sistema deve fazer, enquanto requisitos não funcionais estabelecem restrições sobre os serviços ou funções oferecidas pelo sistema, como desempenho, segurança e usabilidade" (SOMMERVILLE, 2019, p. 142).

"Os requisitos funcionais descrevem as funções específicas que o *software* deve executar. Já os requisitos não funcionais são aqueles que impõem restrições ao produto ou ao processo, como requisitos de desempenho ou qualidade" (PRESSMAN, 2016, p. 152).

"Requisitos funcionais definem os serviços que o sistema deve fornecer, enquanto requisitos não funcionais especificam propriedades de qualidade, como confiabilidade, desempenho, segurança e usabilidade" (PFLEEGER; ATLEE, 2010, p. 160).

"Requisitos funcionais descrevem comportamentos específicos do sistema em resposta a estímulos, enquanto requisitos não funcionais estabelecem características globais de qualidade ou restrições de projeto" (WIEGERS; BEATTY, 2013, p. 110).

## 4 ENGENHARIA DE *SOFTWARE*

De acordo com Sommerville (2018), a engenharia de *software* disponibiliza métodos que dão suporte à definição, concepção e aprimoramento de programas, indo além da mera fase de desenvolvimento.

Conforme destaca Soares (2022), essa área envolve a utilização coordenada de princípios, práticas e ferramentas ao longo de todas as etapas do ciclo de vida do *software*, incluindo desde a definição até a manutenção do sistema.

Para Pressman e Maxim (2021), a engenharia de *software* oferece uma gama de metodologias, estratégias e instrumentos que visam melhorar o desenvolvimento de sistemas, sendo que avanços técnicos continuam surgindo, contribuindo para práticas mais promissoras.

A engenharia de *software* é considerada um dos pilares no processo de criação de sistemas, como observa Araújo (2021), ao afirmar que ela é desenvolvida com foco nos diferentes envolvidos no projeto de *software*. Já Sommerville (2018) ressalta que essa engenharia é indispensável para o funcionamento adequado de múltiplos setores da sociedade.

A elicitação de requisitos, sendo uma área essencial dentro da engenharia de *software*, é definida por Sommerville (2018) como o conjunto de especificações dos serviços que o sistema deve oferecer, bem como as limitações para seu funcionamento. Em complemento, Pfleeger (2004) descreve um requisito como uma funcionalidade ou capacidade que o sistema deve possuir para atingir seus propósitos. “O processo responsável por identificar, analisar, registrar e validar esses requisitos é conhecido como engenharia de requisitos (ER)” (Sommerville , 2018).

Para Pressman (2021), engenharia de *software* é a implementação de uma abordagem organizada, disciplinada e mensurável aplicada ao desenvolvimento, operação e manutenção de *software*. Nesse contexto, inspirado no livro *How to Solve It* de George Polya, Pressman (2021) explica que esse processo se dá em quatro fases principais: identificar o problema, planejar uma solução, colocar o plano em prática e verificar o resultado obtido.

Conforme Valente (2020), o *software* é parte integrante do cotidiano contemporâneo, presente nas corporações, no setor público, no comércio digital, em dispositivos inteligentes e até mesmo na modernização de indústrias tradicionais, evidenciando sua presença constante na sociedade.

Pressman (2021) observa que projetos de *software* têm como objetivo solucionar problemas específicos, corrigir falhas, atualizar sistemas existentes, desenvolver novos produtos ou incrementar funcionalidades em programas já em uso.

Valente (2020) define a Engenharia de *software* como um campo da Computação que se debruça sobre os desafios do desenvolvimento de sistemas complexos e em grande escala, buscando soluções eficientes e de qualidade.

Segundo Carvalho (2024), trata-se de uma área multidisciplinar que une conhecimentos de Engenharia, Arquitetura, Administração e Desenvolvimento com o objetivo de conceber, criar, projetar e administrar sistemas tecnológicos, assegurando sua eficácia e segurança desde a concepção até a entrega final.

## 4.1 PROGRAMAÇÃO ORIENTADA A OBJETOS

A Perspectiva da Programação Centrada em Objetos (PCO) representa um modelo de criação onde o *software* é construído em torno de "objetos" — entidades independentes que agrupam informações e ações. Conforme Tucker (2010), entender um sistema PCO significa vê-lo como um conjunto de objetos que se comunicam, atuando como mini-sistemas individuais.

Essa tática promove a divisão em módulos, o reaproveitamento do código e a simplicidade na representação de sistemas intrincados.

Conforme Alves (2014), a PCO organiza o *software* em categorias, das quais os objetos são criados e passam a interagir uns com os outros durante o funcionamento do programa. Um caso comum é o de uma cadeira como representação da categoria móvel, recebendo características como cor, massa e tamanho (Pressman, 2021). Essa transmissão de características é um dos alicerces desse modelo, junto com a representação simplificada, a diversidade de formas e a proteção dos dados internos dos objetos (Ricarte, 2001).

  Especialistas como Booch (2007) e Oliveira (2022) enfatizam que a PCO vai além de um método de programação, sendo uma maneira eficiente de conceber e organizar sistemas atuais. Ela incentiva a criação de aplicativos mais bem estruturados, adaptáveis e fáceis de manter, sendo muito usada no desenvolvimento moderno de *software*.

## 4.2 *WEB*

A *WEB*, ou *World Wide Web*, representa uma rede global de dados que une arquivos, ilustrações, filmes e outros materiais através da internet, empregando ferramentas como *URLs*, hipertexto e o protocolo *HTTP*. Desenvolvida por Tim Berners-Lee em 1989 e colocada em prática em 1990 no *CERN*, ela viabiliza a exploração entre conteúdos associados (Berners-Lee, 1994). Na visão de Pressman (2021, p. 115), a *Web* é “um conjunto de elementos ligados entre si, exploráveis por meio de hiperligações”, o que simplifica o acesso a dados de maneira prática e dinâmica.

Conforme Kurose e Ross (2012), a *Web* configura um sistema distribuído de dados que proporciona acesso simples e universal a documentos, sendo basicamente “um agrupamento de recursos de dados e serviços, conectados por uma rede de comunicação” (p. 6). Loudon (2010) ressalta que aplicações *web* extensas necessitam de organização em partes isoladas, auxiliando na conservação e na administração da complexidade do sistema.

Hoje, as aplicações *web* estão bastante inseridas no dia a dia, sendo empregadas em setores como ensino, comércio, saúde e lazer. Pruteanu (2019) salienta que essas aplicações se tornaram cruciais à estrutura da sociedade atual, por serem de fácil desenvolvimento, rápida implementação e extremamente acessíveis. Desse modo, a *web* se desenvolveu além de um arquivo de documentos, convertendo-se em uma plataforma essencial para soluções digitais interativas e de colaboração.

## 4.3 LINGUAGENS DE PROGRAMAÇÃO

# 

Conforme Sebesta (2010) explica, essas linguagens estabelecem uma maneira estruturada de explicitar algoritmos através de padrões sintáticos e semânticos próprios. Sua divisão ocorre em modelos variados, como o imperativo, funcional, orientado a objetos e lógico, apresentando cada qual métodos diferentes para solucionar desafios computacionais.

Schildt (2015) salienta que a linguagem Java exerceu um efeito revolucionário no desenvolvimento de *software*, notabilizando-se por sua capacidade de rodar em diferentes plataformas e rápida popularização. Do mesmo modo, o *JavaScript* se firmou como uma linguagem universal crucial para a *internet.* Mendes (2021) informa que essa linguagem opera em todos os principais navegadores, e Silva (2023) enfatiza sua flexibilidade na produção de conteúdos interativos para sites, aplicativos móveis e até jogos.

A seleção de uma linguagem de programação está ligada a elementos como objetivo, performance, simplicidade na manutenção e auxílio da comunidade. Tanenbaum (2009) observa que linguagens mais sofisticadas concedem maior abstração, auxiliando no desenvolvimento de sistemas integrados de forma mais organizada. O conhecimento de linguagens como *Java*, *JavaScript*, *Python* ou *C* é, desse modo, essencial para os profissionais da área, visto que afeta diretamente a qualidade, o potencial de expansão e o rendimento das soluções criadas.

## *4.3.1 Model-View-Controller* (MVC)

O modelo arquitetural MVC, ou *Model-View-Controller*, é um padrão de projeto consagrado, apresentado inicialmente por Krasner em 1988 e, mais tarde, popularizado por várias tecnologias atuais. Conforme explica Felipe Miranda em 2016, o MVC divide os elementos de um sistema em três partes fundamentais: o modelo, que cuida da lógica da aplicação e do gerenciamento dos dados; a visualização, que mostra ao utilizador como está a aplicação naquele momento; e o controlador, que serve de ligação, processando as ações do utilizador e coordenando as mudanças na visualização e no modelo. Essa divisão de tarefas ajuda a organizar melhor o código, a reaproveita-lo e a simplificar a sua manutenção.

De acordo com Deacon em 2009, o Modelo é a parte mais consistente da aplicação, sem ligação direta com a interface que o utilizador vê, enquanto a visualização tem a função de exibir as informações de forma visual, como janelas e imagens. O Controlador, por outro lado, age diretamente sobre a visualização, conhecendo-a, mas o contrário não acontece. Essa organização em níveis melhora a independência do sistema, permitindo que mudanças numa parte não afetem as outras, sendo muito usada em plataformas como *Angular*, *Django*, *Laravel*, entre outras.

# 

## *4.3.2 Personal Home Page/Forms Interpreter* (PHP)

Criado primeiramente em 1995 por Rasmus Lerdorf como PHP/FI (*Personal Home Page/Forms Interpreter*), o PHP extrapolou rapidamente suas metas iniciais, convertendo-se em uma das linguagens mais empregadas no desenvolvimento para a *web* (Niederauer, 2017).

Conforme Moraz (2005), o PHP viabiliza tanto a programação estruturada quanto a focada em objetos, conseguindo produzir não só HTML dinâmico, mas também imagens, documentos PDF, animações em *Flash* e arquivos em formatos como XML e XHTML, tornando-se uma alternativa versátil para a criação de aplicações *web*. Ademais de sua simplicidade de utilização, o PHP possui uma comunidade atuante que coopera para seu desenvolvimento constante.

De acordo com Milani (2016), a linguagem é particularmente recomendada para *websites* e portais, oferecendo uma performance rápida e amparo a diversas bibliotecas e módulos já preparados, direcionados sobretudo a aplicações *web* e de comércio virtual. Esse apoio colaborativo, associado à sua essência de código aberto, tem fomentado a progressão da linguagem e assegurado sua posição de relevo no panorama do desenvolvimento contemporâneo.

## *4.3.3 Laravel*

# 

Com base no padrão *MVC* (*Model-View-Controller*) e nas práticas de programação orientada a objetos, o *Laravel* é um *framework* PHP projetado para tornar o desenvolvimento mais agradável, inventivo e eficiente, conforme aponta Gabardo (2017). Ele foi construído para simplificar as atividades comuns no desenvolvimento *web*, como autenticação, roteamento, gerenciamento de sessões e filas, e implementação de *cache*. O *Laravel* oferece uma estrutura organizada, clara e com um código bem escrito, o que explica o uso da analogia dos “artesãos” em seu lema (Gabardo, 2017).

Na visão de Douglas e Marabesi (2017), o lançamento do *Laravel* causou uma mudança importante no mercado, que era amplamente influenciado por alternativas como *Zend Framework* e *Symfony*. A facilidade com que ele permite implementar recursos complexos atraiu a atenção da comunidade de desenvolvedores. Joukovski (2021) acrescenta que sua característica de ser gratuito, de código aberto e a simplicidade na construção e clareza do código o transformaram em um dos *frameworks* PHP mais utilizados hoje em dia.

## *4.3.4 Javascript*

A linguagem *JavaScript* foi criada na década de 1990, fruto de uma colaboração entre a *Netscape* e a *Sun Microsystems,* com o propósito de tornar as páginas *web* mais interativas (SILVA, 2010). Grillo e Fortes (2008) mencionam que a linguagem inicialmente recebeu o nome de *Mocha*, sendo rebatizada como *JavaScript* em 1995. Embora o nome tenha sido escolhido por sua semelhança com a sintaxe da linguagem *Java*, não existe relação direta entre ambas. Ainda assim, segundo Flanagan (2013), a associação nominal persistiu e *JavaScript* ganhou ampla aceitação entre os desenvolvedores.

A distinção entre *Java* e *JavaScript* é claramente observada em seu funcionamento: enquanto o *Java* opera, em geral, no lado do servidor, o *JavaScript* é executado no lado do cliente (SILVA, 2010). De acordo com Grillo e Fortes (2008), o *JavaScript* possibilita a inserção de pequenos *scripts* diretamente nas páginas HTML. Flanagan (2013) ressalta que a linguagem já ultrapassou sua concepção inicial como um simples *script*, tornando-se uma linguagem de programação versátil e poderosa.

Josh Goldberg (2022, p. 25) informa que Brendan Eich desenvolveu o *JavaScript* em apenas dez dias, em 1995, enquanto trabalhava na *Netscape*, com o objetivo de criar uma linguagem acessível e de fácil uso em ambientes *web*.

Atualmente, o *JavaScript* é considerado a linguagem padrão da *web*, presente na maioria absoluta dos sites e compatível com todos os navegadores modernos, seja em computadores, dispositivos móveis, consoles ou *tablets*. De acordo com Flanagan (2013, p. 1), essa abrangência faz do *JavaScript* a linguagem mais utilizada na história da computação.

Além do uso tradicional no desenvolvimento *web*, o *JavaScript* também tem ampliado sua aplicação no desenvolvimento de aplicativos móveis, impulsionado por ferramentas como o *React Native* e o *Ionic*. Estas tecnologias possibilitam a criação de aplicações nativas para plataformas como *Android* e *iOS*, utilizando *JavaScript* e bibliotecas relacionadas (FLANAGAN, 2013).

Prescott (2016, p. 10) observa que o *JavaScript* é amplamente reconhecido como a principal linguagem de *script* utilizada no lado do cliente, destacando-se por sua flexibilidade e funcionalidade dinâmica. Trata-se de uma linguagem amplamente empregada para adicionar elementos interativos, animações e responsividade às páginas da *web*. Sua utilização se estende ao controle de navegadores, comunicação assíncrona com servidores, modificações dinâmicas de conteúdo, além de possibilitar o desenvolvimento de jogos e aplicações para *desktop* e dispositivos móveis.

Conforme enfatiza Flanagan (2013), o *JavaScript* exerce um papel central na *web* contemporânea, fornecendo uma gama extensa de recursos que favorecem tanto a experiência do usuário quanto a interatividade de aplicações *web* em escala global.

## 

## 

## 5 SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS

## 

Uma fase importante no processo de preparação de um sistema é a escolha do banco de dados a ser utilizado. De acordo com Elmasri & Navathe (2011), podemos considerar o banco de dados, afinal, como um grupo de dados inter-relacionados, armazenados e organizados para um número de tarefas. Date (2004) mantém o mesmo ponto de vista, ao indicar que o banco de dados é o responsável pela boa organização das grandes quantidades de informações unidas. O mesmo autor reafirma que uma boa organização dos dados tem um papel fundamental para a operação de qualquer sistema informático atualmente.

Para organizar esses dados, de maneira coerente e aproveitá-los adequadamente, é preciso um Sistema de Gerenciamento de Banco de Dados (SGBD). De acordo com Silberschatz, Korth e Sudarshan (2011), a SGBD é um conjunto de programas que auxilia nos cuidados necessários para armazenar, recuperar e modificar os diversos dados em bancos organizados. Coronel e Morris (2016) complementam observando que os SGBDs são intermediários entre os dados brutos e os usuários e/ou aplicações, proporcionando uma maior interação, eficiência, segurança e coerência com os dados armazenados.

Para além do aumento na rapidez de acesso e manutenção dos dados, os SGBDs possibilitam vantagens como a independência dos dados, gerenciamento da concorrência, integridade referencial e segurança (SILBERSCHATZ; KORTH; SUDARSHAN, 2011). Rob e Coronel (2007) afirmam que tais vantagens transformam os SGBDs assim, em um dos pilares da infraestrutura da tecnologia, ajudando significativamente na confiabilidade dos sistemas e

## 5.1 *STRUCTURED QUERY LANGUAGE*

Thomas Nield (2016) destaca que o domínio da linguagem *SQL* pode ser rapidamente alcançado por meio da prática com exercícios simples, permitindo que o usuário crie e gerencie seus próprios bancos de dados mesmo sem um ambiente de servidor.

Segundo Alice Zhao (2022), a linguagem SQL possui variações significativas entre diferentes sistemas gerenciadores, como *SQL Server*, *MySQL* e *PostgreSQL*, sendo essencial compreender suas diferenças para consultas e alterações eficazes.

Ben Forta (2013) defende uma abordagem direta e segmentada para o ensino de SQL, onde o aprendizado é dividido em pequenas lições voltadas à aplicação prática e imediata da linguagem.

Para Alan Beaulieu (2009), a melhor maneira de aprender SQL é por meio de exemplos claros e exercícios práticos, que auxiliam na consolidação dos fundamentos da linguagem desde suas aplicações mais simples até as mais complexas.

Allen e Owen enfatizam (2020) a importância do domínio das operações básicas da linguagem SQL, como inserção, atualização e recuperação de dados, como pré-requisito para a manipulação eficiente de bancos de dados relacionais.

Cathy McClain (2021) sugere que o uso avançado de SQL vai além de simples consultas, sendo possível aplicá-lo a análises complexas como séries temporais, cortes e detecção de anomalias, tornando-se uma ferramenta poderosa para cientistas de dados.

## 5.2 *phpMyAdmin*

O *phpMyAdmin* é uma ferramenta de administração de bancos de dados *MySQL* e MariaDB se utilizando de uma interface gráfica acessível por navegador, que se popularizou pelo seu conforto na administração dos dados, principalmente no desenvolvimento web. Segundo Oliveira (2021), o *phpMyAdmin* facilita o trabalho em tarefas comuns, como a criação de tabelas, inserção e edição de registros, execução de comandos SQL e a realização de *backups*, sem a necessidade de entrar diretamente na linha de comando.

Criado em 1998, o *phpMyAdmin* é um *software* livre desenvolvido em PHP e compatível com diferentes sistemas operacionais e navegadores (MACEDO; NASCIMENTO, 2018). Conforme Silveira e Carvalho (2020), a interface gráfica intuitiva do *phpMyAdmin* é um dos seus diferenciais, uma vez que é especialmente útil para os iniciantes na administração dos bancos de dados, pelo oferecimento de um ambiente visual de fácil navegação.

E por fim, oferece recursos mais avançados, como exportação de dados em múltiplos formatos, gerenciamento de usuários e permissões, execução de rotinas armazenadas, visualização de índices e estatísticas de desempenho (FERREIRA; LIMA, 2019). Fazendo dessa ferramenta uma alternativa viável não apenas para a educação, mas também para pequenos e médios projetos em ambientes reais.

## 6 METODOLOGIA

## 

Identificando a causa raiz do problema, adotando uma abordagem abrangente e uma análise aprofundada validando a necessidade junto aos *stakeholders* será realizado entrevistas e questionários com artistas locais e possíveis contratantes para identificar as principais dificuldades e necessidades, utilizando ferramentas como o *PM Canvas* (*Project Model Canvas*), questionários realizado por meio do *Google Forms,* que será divulgado por meio de redes sociais que contenham um público alto de artistas locais.

Para elicitar e documentar os requisitos funcionais e não funcionais, foi realizada a análise de documentos relevantes como os dados coletados por meio de entrevistas e questionários, entendendo as necessidades de negócios e as expectativas sobre o produto, organizando sessões colaborativas com os *stakeholders*, esclarecendo pontos e validando as expectativas, garantindo que todos os aspectos sejam abordados, identificando funcionalidades e características do sistema.

A análise do sistema e banco de dados iniciada com a modelagem, começando pela elaboração de um diagrama de casos de uso que descreve todas as interações entre os usuários e o *software*. Utilizando fluxo de dados e funcionalidades do sistema, por meio de diagramas de casos de uso, diagrama de entidade e relacionamentos (DERs), um diagrama de classe também foi desenvolvido para representar a estrutura e a interação entre as entidades presentes no sistema, garantindo que o sistema seja construído de maneira eficiente, atendendo aos requisitos de desempenho e escalabilidade. Além disso, as tabelas do banco de dados foram modeladas e estruturadas utilizando a ferramenta *PostgreSQL*.

O processo de prototipação das telas do sistema foi realizado com artefatos de alta fidelidade, com o objetivo de facilitar o entendimento coletivo do que estava sendo desenvolvido.

Utilizando-se do *FIGMA*, um *software* para desenvolvimento, conseguimos chegar a um protótipo de alta fidelidade e apresentar de maneira eficaz, demonstrando o *design* do *software* de forma visual, interativa e clara, conseguindo uma apresentação bem-sucedida e a garantia que o protótipo seja completo, interativo e organizado de forma intuitiva, explicando as decisões de *design* e coletando *feedback* dos *stakeholders*, garantindo a coesão entre os diferentes componentes do sistema para que os desenvolvedores possam implementar com precisão.

Através de ferramentas como o *Qase*, realizamos testes de unidade para garantir a correta implementação do sistema e foram executados testes unitários de todos os requisitos sejam eles funcionais ou não funcionais. Isso garantiu que cada parte do sistema funcione como esperado e identificando erros que precisam ser corrigidos antes da fase de testes mais ampla, garantindo a implementação do sistema de forma segura e coesa.

## 6.1 CARACTERIZAÇÃO DA PESQUISA

Quanto aos fins, este trabalho irá utilizar as seguintes formas: exploratória, explicativa, metodológica e intervencionista

De acordo com Gil (2017), a pesquisa exploratória tem como objetivo principal o aprimoramento de idéias ou descoberta de intuições, proporcionando maior familiaridade com o problema, tornando-o mais explícito e constituindo hipóteses.

A pesquisa explicativa é o mais complexo e delicado, pois o risco de cometer erros é alto. Esta metodologia preocupa-se principalmente identificar fatores que determinam ou contribuem para a ocorrência de fenômenos, aprofundando-se no conhecimento da realidade, explicando a razão, o porquê das coisas (Gil, 2012).

"A pesquisa metodológica tem como propósito o estudo, o desenvolvimento e a avaliação de métodos e técnicas aplicáveis à investigação científica". (Gil, 2019, p. 17).

"A pesquisa intervencionista não apenas descreve e analisa atividades, mas também participa ativamente na sua transformação” (Engestrom, 2015, p. 141).

Quanto aos meios a pesquisa é classificada como, campo, experimental, bibliográfico, documental e participante.

Para Gil (2002), a pesquisa de campo é conjugada com procedimentos, tais como a análise de documentos, filmagem e fotografias e ressalta uma comunidade, observando diretamente as atividades do grupo a ser estudado e entrevistas com informantes para captar suas explicações e interpretações do que ocorre no grupo.

Segundo Gil (2008), o método experimental submete os objetos de estudo à influência de certas variáveis, em condições controladas e conhecidas pelo investigador, que observa os resultados que a variável produz no objeto.

“Resenha, recensão de livros ou análise bibliográfica é uma síntese ou um comentário dos livros publicados feito em revistas especializadas nas várias áreas da ciência, das artes e da filosofia” (Severino, 2000, p.131).

Para Gil (1989), a pesquisa documental utiliza documentos de segunda mão, que já foram analisados.

“A pesquisa participante, assim como a pesquisa-ação, caracteriza-se pela interação entre pesquisadores e membros das situações investigadas” (Gil, 1898, p.56).

## 6.2 TÉCNICAS DE COLETA DE DADOS

## 

No campo os dados da pesquisa foram coletados por meio de questionário, incluindo perguntas abertas e fechadas.

Segundo Vergara (2004) o mesmo destaca que o questionário é realizado por uma série de perguntas ao público alvo de maneira escrita incluindo perguntas abertas e também fechadas. É um método muito eficaz para análise e coleta de dados. O questionário tem a vantagem de aplicar um padrão nas respostas, com isso facilita bastante a análise e observação dos dados. Contudo essa forma de pesquisa utilizando questionário é vastamente utilizada em pesquisas quantitativas, garantindo uma maneira planejada e precisa de obter dados.

De acordo com Severino (2013), o questionário caracteriza-se como um instrumento de coleta de dados formado por um conjunto de questões aplicadas a um determinado grupo de respondentes. Essa ferramenta tem como propósito principal captar informações que expressem as opiniões dos participantes em relação ao objeto de estudo. Essa abordagem oferece um procedimento sistematizado para a obtenção de dados quantitativos, viabilizando uma análise objetiva e mensurável das respostas coletadas.

Fora do campo os dados foram obtidos por meio experimental e pesquisa-ação.

## 6.3 FORMA DE ANÁLISE DE DADOS

Quanto à forma de abordagem, a pesquisa é quantitativa e qualitativa, quantitativa para avaliar melhor as necessidades, trazendo números e dados sólidos e qualitativa contendo maiores informações e somando ao projeto.

Para Gil (2006), a pesquisa qualitativa explora fenômenos complexos por meio de métodos como observação, entrevistas e relatos, buscando compreender significados e interpretações, trazendo subjetividade e contextualizando a investigação. Ainda assim, aliada a pesquisa qualitativa a pesquisa quantitativa vem de forma a somar os resultados.  
 Segundo Gil (2006) as pesquisas quantitativas consideram que tudo pode ser contável, ou seja, tudo que é gerado a partir de números para classificar e analisar informações.  
 Os dados foram analisados por meio de gráficos, trazendo respostas e a validação de necessidade.

De acordo com Oliveira (2018) gráficos são figuras que servem para representação de dados, e utilizados com habilidade evidenciam aspectos visuais dos dados, dando compreensão de forma clara e fácil, empregados para dar clareza a certas relações significativas.

## 7 CONTEXTUALIZAÇÃO E RESULTADOS

O projeto desenvolveu uma plataforma *web* para conectar artistas locais a contratantes, enfrentando desafios como informalidade, baixa visibilidade e falta de oportunidades. A solução busca promover inclusão, diversidade e sustentabilidade no setor cultural, alinhada à ODS 8 da ONU.

A plataforma permite perfis personalizados, busca por localização e área de atuação, entre outros, além de avaliações que fortalecem a profissionalização dos usuários. O sistema foi pensado para ser acessível, intuitivo e adaptável à realidade dos artistas, incluindo MEIs.

Como parte da pesquisa, foi aplicado um questionário com 13 perguntas a 98 participantes, cujas respostas ajudaram a entender melhor o cenário atual dos artistas e guiaram o desenvolvimento da plataforma.

## 7.1 CONTEXTUALIZAÇÃO

A proposta teve como propósito central o desenvolvimento de uma plataforma *web* voltada para a valorização e conexão de artistas locais com potenciais contratantes, visando mitigar os desafios enfrentados por profissionais do setor cultural, como a informalidade, a escassez de oportunidades e a dificuldade de visibilidade no mercado. Ao longo do projeto, evidenciou-se que a criação da ferramenta não se restringe à dimensão técnica, mas abarca também aspectos sociais e culturais relevantes para a promoção da diversidade, inclusão e sustentabilidade econômica no meio artístico.

A análise do contexto sócio-cultural revelou que muitos artistas locais, sobretudo os iniciantes ou informais, enfrentam barreiras significativas para se inserirem no mercado de trabalho formal. A ausência de plataformas específicas, como aquelas disponíveis para outros profissionais, a exemplo do *LinkedIn*, contribui para a fragilização das trajetórias artísticas e para o abandono precoce de carreiras promissoras. Ao mesmo tempo, contratantes também encontram dificuldades em localizar artistas qualificados, recorrendo frequentemente a redes informais ou indicações pouco acessíveis a novos talentos.

Nesse cenário, o sistema proposto surge como resposta direta à lacuna identificada entre artistas e contratantes, com base nos princípios estabelecidos pela ODS 8 da ONU, que propõe o trabalho decente e o crescimento econômico. A plataforma permite a criação de perfis personalizados, categorização por segmentos artísticos como música, teatro e artes visuais, mecanismos de busca por localização e área de atuação, além de funcionalidades para avaliações e *feedbacks* que fortalecem a confiabilidade e a profissionalização dos usuários.

Do ponto de vista técnico, o desenvolvimento priorizou uma arquitetura acessível, escalável e intuitiva, de forma a contemplar a diversidade de perfis de usuários, desde artistas com baixa familiaridade tecnológica até consumidores em busca de praticidade e eficiência em sua contratação. O sistema foi pensado para ser inclusivo e adaptável às dinâmicas regionais, promovendo a descentralização do acesso às oportunidades culturais.

Para implementar essas funções, e garantir uma experiência agradável e acessível pros usuários, montamos grupos de situações, que demonstram como o sistema age a cada instante. Um desses grupos é o de registro do usuário, que detalha o trajeto de um dos personagens mais importantes da plataforma: o Microempreendedor Individual MEI. Esse caminho nós dividimos em três partes principais: o cadastro da empresa, onde o CNPJ é verificado automaticamente pela regra RN004; o preenchimento do endereço, que usa o CEP; e a criação da conta do usuário, onde ele define nome e senha, verificando a igualdade das senhas digitadas, seguindo a regra RN001. Essa sequência segue o que o sistema precisa fazer e a forma como ele deve fazer, o que garante que ele seja confiável, simples de usar e dentro das leis e regras técnicas do projeto.

Conclui-se, portanto, que o projeto não apenas oferece uma solução digital inovadora, mas também se caracteriza como uma iniciativa de impacto social significativo, ao fortalecer os vínculos entre cultura, economia e tecnologia.

Acredita-se que, com a consolidação da plataforma, será possível fomentar uma cadeia produtiva cultural mais justa, acessível e sustentável, contribuindo para a valorização da arte local, a redução da informalidade e o estímulo à empregabilidade no setor artístico.

Para representar a interação do usuário com o sistema, foram criados pacotes de casos de uso, sendo cada um deles vinculado a uma funcionalidade específica. Ao todo, foram produzidos dez diagramas, abrangendo as telas de *login*, cadastro de usuário, tela inicial, cadastro de orçamento, clientes, agendamento, ordem de serviço, relógio e cursos — todos apresentados no Apêndice D.

## 7.2 RESULTADOS

Realizamos 1 (um) questionário onde obtivemos 98 (noventa e oito) respostas com o objetivo de trazer melhor em forma de perguntas a realidade dos artistas, foram realizadas no total 13 (treze) perguntas, contendo 1 (uma) pergunta fechada.

**REFERÊNCIAS**

BRASIL. **Constituição da República Federativa do Brasil**. Brasília, DF: Senado Federal, 2008.

CORRÊA, Henrique Luiz. **Administração de produção e operações**: manufatura e serviços. 2. ed. São Paulo: Atlas, 2009. *E-book*.

GOUVEIA JÚNIOR, Mário; SANTOS, Raimundo Nonato Macedo. Mudança de paradigma e sua ruptura: um estudo de caso na Museologia e a pluralidade paradigmática da Ciência da Informação. **TransInformação**, Campinas, v. 24, n. 2, p. 117-126, maio/ago. 2012. Disponível em: http://www.scielo.br/scielo.php?pid. Acesso em: 20 set. 2023.

ALVES, D. F. *Programação orientada a objetos: conceitos e práticas*. São Paulo: Érica, 2014.

BOOCH, G. *Object-oriented analysis and design with applications*. 3. ed. Boston: Addison-Wesley, 2007.

DEITEL, H. M.; DEITEL, P. J. *Java: como programar*. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

LARMAN, C. *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao processo unificado*. 3. ed. Porto Alegre: Bookman, 2007.

FERREIRA, Marília C.; BOECHAT, Cláudio. *Objetivos do Desenvolvimento Sustentável Pós-2015*. 1. ed. Belo Horizonte: Fundação Dom Cabral, 2014.

TREVISAM, Elisaide; CRUCIOL JÚNIOR, Jessé. Objetivos do Desenvolvimento Sustentável: o direito humano e o suporte fático da rede da vida. *Revista de Direito da Unicuritiba*, Curitiba, v. 6, n. 2, 2022.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS (ONU). *Objetivos de Desenvolvimento Sustentável: a Agenda 2030*. 1. ed. Nova York: ONU, 2015.

BUSS, Paulo Marcos; FERREIRA, José Roberto; HOIRISCH, Célia; MATIDA, Áurea Helena. Desenvolvimento sustentável e governança global em saúde: a contribuição da Agenda 2030. *Revista Epidemiologia e Serviços de Saúde*, Brasília, v. 31, n. esp. 1, e20211047, 2022.

OBERLEITNER, R.; MASIERO, P. C. *Engenharia de software orientada a objetos*. São Paulo: Pearson, 2011.

OLIVEIRA, L. F. *Paradigmas da programação: uma abordagem prática*. Rio de Janeiro: Ciência Moderna, 2022.

RICARTE, I. L. M. *Projeto de sistemas com UML*. São Paulo: Pearson Makron Books, 2001.

TUCKER, A. B. *Fundamentos de ciência da computação*. Rio de Janeiro: LTC, 2010.

BERNERS-LEE, T. *Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor*. New York: HarperSanFrancisco, 1994.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. 5. ed. São Paulo: Pearson Addison Wesley, 2012.

LOUDON, K. C.; LOUDON, J. P. *Sistemas de informação gerenciais*. 9. ed. São Paulo: Pearson Prentice Hall, 2010.

PRUTEANO, M. *Desenvolvimento de aplicações web modernas*. São Paulo: Novatec, 2019.

MENDES, F. R. *Desenvolvimento web com JavaScript moderno*. São Paulo: Novatec, 2021.

SCHILDT, H. *Java: a complete reference*. 9. ed. *New York*: McGraw-Hill Education, 2015.

SEBESTA, R. W. *Conceitos de linguagens de programação*. 9. ed. Porto Alegre: Bookman, 2010.

SILVA, A. M. *JavaScript aplicado: do básico ao avançado*. Rio de Janeiro: Ciência Moderna, 2023.

TANENBAUM, A. S. *Organização Estruturada de Computadores*. 5. ed. São Paulo: Pearson Prentice Hall, 2009.

DEACON, J. *Model View Controller (MVC)*. 2009. Disponível em: http://www.dcs.bbk.ac.uk/~roger/mvc.html. Acesso em: 9 abr. 2025.

KRASNER, G. E.; POPE, S. T. *A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80*. Journal of Object-Oriented Programming, v. 1, n. 3, p. 26–49, 1988.

MIRANDA, F. *Model View Controller: padrões e boas práticas de arquitetura de software*. São Paulo: Casa do Código, 2016.

MILANI, A. *PHP: Programando com orientação a objetos*. São Paulo: Novatec, 2016.

MORAZ, R. *Aprendendo PHP: do básico ao profissional*. São Paulo: Érica, 2005.

NIEDERAUER, R. *PHP moderno: programação orientada a objetos*. São Paulo: Novatec, 2017.

DOUGLAS, R.; MARABESI, A. *Laravel: um framework para artesãos da web*. São Paulo: Novatec, 2017.

GABARDO, C. *Laravel para ninjas*. São Paulo: Casa do Código, 2017.

JOUKOVSKI, J. *Desenvolvimento web com Laravel para iniciantes*. São Paulo: Érica, 2021.

OLIVEIRA, L. M. *Métodos e técnicas de pesquisa científica*. São Paulo: Atlas, 2018.

BARDIN, L. Prototipagemno *design* de produtos e serviços. 2021.

BITTENCOURT, L. *Interfaces* modernas com *Figma*. 2023.

FERREIRA, D. *Design* de produtos*: abordagens práticas*. 2020.

MARTIN, R.; BIERLY, M. *Design colaborativo com Figma*. 2019.

SANTOS, T. *Prototipação em projetos de software*. 2024.

SILVA, J. *Fundamentos da prototipagem digital*. 2022.

VALENÇA, J. *Modelagem de processos com BPMN*. 2012.

VALE, S. *Origem e aplicação dos protótipos no desenvolvimento de sistemas*. 2020.

VIRGENS, A. *Prototipação de interfaces: práticas e ferramentas*. 2009.

MARTIN, R. *Análise e projeto orientados a objetos com aplicações UML*. São Paulo: Makron Books, 1996.

WAZLAWICK, R. S. *Análise e projeto de sistemas de informação orientados a objetos*. Rio de Janeiro: Elsevier, 2011.

HEUSER, Carlos A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2010.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistemas de banco de dados*. 5. ed. Rio de Janeiro: Elsevier, 2006.

ELMASRI, Ramez; NAVATHE, Shamkant B. *Fundamentos de sistemas de banco de dados*. 5. ed. São Paulo: Pearson Addison Wesley, 2005.

CAMPOS, André L. N. *Modelagem de processos de negócios com BPMN*. [S.l.]: Brasport, 2012.

CAPOTE, Gart. *Fuja do fluxograma: guia de modelagem da verdade com BPMN*. 2. ed. São Paulo: FutureThink, 2015.

SABARÁ, Saulo; VALLE, Rogério. *Análise e modelagem de processos de negócio: notação BPMN*. Rio de Janeiro: Elsevier, 2018.

ABPMP. *Guia para o Gerenciamento de Processos de Negócio – BPM CBOK V3.0*. 2. ed. São Paulo: ABPMP Brasil, 2013.

DUMAS, Marlon et al. *Fundamentos de BPM*. Porto Alegre: Bookman, 2014.

FREUND, Jakob; RÜCKER, Bernd. *BPMN na prática: usando a notação de processos de negócio de forma eficaz*. Rio de Janeiro: Brasport, 2015.

PESSOA, Osvaldo. *Dominando BPMN: da teoria à prática*. São Paulo: Editora Érica, 2017.

SILVER, Bruce. *BPMN – Método e Estilo*. Porto Alegre: Bookman, 2011.

CAMPOS, André L. N. *Modelagem de Processos com BPMN*. 2. ed. Rio de Janeiro: Brasport, 2014.

FALCONI, Vicente. *Gestão da Qualidade Total: no estilo japonês*. 3. ed. Nova Lima: Falconi Consultores, 1996.

GEORGE, Michael L. et al. *Lean Six Sigma: guia prático*. Rio de Janeiro: Elsevier, 2005.

LIKER, Jeffrey K. *O modelo Toyota: 14 princípios de gestão do maior fabricante do mundo*. Porto Alegre: Bookman, 2005.

OHNO, Taiichi. *O Sistema Toyota de Produção: além da produção em larga escala*. Porto Alegre: Bookman, 1997.

RIES, Eric. *A startup enxuta: como os empreendedores atuais utilizam a inovação contínua para criar empresas extremamente bem-sucedidas*. São Paulo: Leya, 2012.

FINOCCHIO JÚNIOR, José. *Project Model Canvas: gerenciamento de projetos sem burocracia*. São Paulo: Évora, 2013.

FINOCCHIO JÚNIOR, J. *Project Model Canvas: gerenciamento de projetos de forma visual*. São Paulo: Saraiva, 2013.

MAXIMIANO, A. C. A. *Introdução à administração*. 9. ed. São Paulo: Atlas, 2021.

MENDES, J. *Fundamentos de gerenciamento de projetos*. Rio de Janeiro: FGV Editora, 2015.

PINHEIRO, P. et al. *Modelos de negócio e o uso do Canvas*. Ponta Grossa: AYA Editora, 2015.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software: uma abordagem profissional*. 9. ed. Porto Alegre: AMGH, 2021.

REIS, F. *Competências em gestão de projetos*. [S.l.]: Lulu.com, 2018.

BEZERRA, Eduardo. *Princípios de Análise e Projeto de Sistemas com UML*. 3. ed. Rio de Janeiro: Elsevier, 2020.

MARTIN, Robert C. *Código limpo: habilidades práticas do Agile software*. Rio de Janeiro: Alta Books, 2018.

OLIVEIRA, Ricardo. *Engenharia de software: fundamentos, métodos e aplicações*. São Paulo: Atlas, 2022.

SALIBA JÚNIOR, Walter de Abreu. *Engenharia de requisitos com UML: uma abordagem prática*. São Paulo: Érica, 2020.

SOARES, Tatiane. *Modelagem de sistemas com UML: fundamentos e práticas*. São Paulo: Ciência Moderna, 2022.

SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson Prentice Hall, 2018.

WAZLAWICK, Raul Sidnei. *Análise e projeto de sistemas de informação orientados a objetos*. Rio de Janeiro: Elsevier, 2011.

GRILLO, A. A.; FO

RTES, R. P. M. *Desenvolvimento web com JavaScript*. Rio de Janeiro: LTC, 2008.

FLANAGAN, D. *JavaScript: o guia definitivo*. 6. ed. São Paulo: O’Reilly Media, 2013.

GOLDBERG, J. *Learning TypeScript*. Sebastopol: O’Reilly Media, 2022.

PRESCOTT, T. *JavaScript para desenvolvedores web*. São Paulo: Novatec, 2016.

SILVA, R. F. *Introdução ao JavaScript moderno*. Porto Alegre: Bookman, 2010.

ARAÚJO, J. A. *Fundamentos da engenharia de software*. São Paulo: Editora Exemplo, 2021.

CARVALHO, R. F. *Engenharia de software moderna: fundamentos e práticas integradas*. 2. ed. Rio de Janeiro: Ciência Moderna, 2024.

PFLEEGER, S. L. *Engenharia de software: teoria e prática*. São Paulo: Pearson Addison Wesley, 2004.

SOARES, L. M. *Ciclo de vida de software e práticas ágeis na engenharia moderna*. Belo Horizonte: Editora SoftTech, 2022.

VALENTE, M. T. *Engenharia de software moderna: princípios e práticas para times ágeis*. São Paulo: Casa do Código, 2020.

NIELD, Thomas. *Introdução à Linguagem SQL: abordagem prática para iniciantes*. São Paulo: Novatec, 2016.

ZHAO, Alice. *SQL: guia prático: um guia para o uso de SQL*. São Paulo: O'Reilly, 2022.

FORTA, Ben. *SQL em 10 minutos por dia*. 4. ed. São Paulo: Pearson, 2013.

BEAULIEU, Alan. *Aprendendo SQL: dominando os fundamentos de SQL*. São Paulo: O'Reilly, 2009.

ALLEN, Allen G.; OWEN, Chris. *SQL para leigos*. 8. ed. São Paulo: Alta Books, 2020.

MCCLAIN, Cathy. *SQL para análise de dados: técnicas avançadas para transformar dados em insights*. São Paulo: O'Reilly, 2021.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML – Guia do Usuário*. 2. ed. Rio de Janeiro: Elsevier, 2007.

LARMAN, C. *Utilizando UML e Padrões: uma introdução à análise e ao projeto orientado a objetos e ao processo unificado*. 3. ed. Porto Alegre: Bookman, 2007.

PRESSMAN, R. S. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2015.

CORONEL, Carlos; MORRIS, Steven. *Database systems: design, implementation, and management*. 11. ed. Boston: Cengage Learning, 2016.

DATE, C. J. *Introdução a sistemas de banco de dados*. 8. ed. Rio de Janeiro: Campus, 2004.

 ELMASRI, Ramez; NAVATHE, Shamkant B. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

 ROB, Peter; CORONEL, Carlos. *Database systems: design, implementation, and management*. 6. ed. Boston: Course Technology, 2007.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

BEZERRA, Ana Carolina; MATOS, Pedro Henrique. *Gerenciamento de banco de dados com phpMyAdmin: uma abordagem prática*. Fortaleza: Editora Acadêmica, 2022.

FERREIRA, Lucas; LIMA, Rafael. *Banco de dados: teoria e prática com MySQL e phpMyAdmin*. São Paulo: Novatec, 2019.

MACEDO, Jorge; NASCIMENTO, Tiago. *Ambientes de desenvolvimento web: uma introdução ao uso do XAMPP e phpMyAdmin*. Recife: IFPE, 2018.

OLIVEIRA, Camila. *Gerenciamento de dados com phpMyAdmin*. Curitiba: Appris, 2021.

 SILVEIRA, Marlon; CARVALHO, Denise. Ferramentas de apoio ao desenvolvimento web: MySQL Workbench e phpMyAdmin em comparação. In: ENCONTRO NACIONAL DE TECNOLOGIA, 2020. Anais [...]. Porto Alegre: Editora da UFRGS, 2020.

PFLEEGER, Shari Lawrence; ATLEE, Joanne M. *Engenharia de Software: teoria e prática*. 4. ed. São Paulo: Pearson, 2010

|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE A – Identificação da causa raiz do problema | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autores** | **Notas da Revisão** |
| 1.0 | 06/05/2025 | Mateus Pires, Leonardo Rodrigues e Bruno Jucoski | Criação |

## 

Quadro 1 – Identificação da causa raiz do problema.

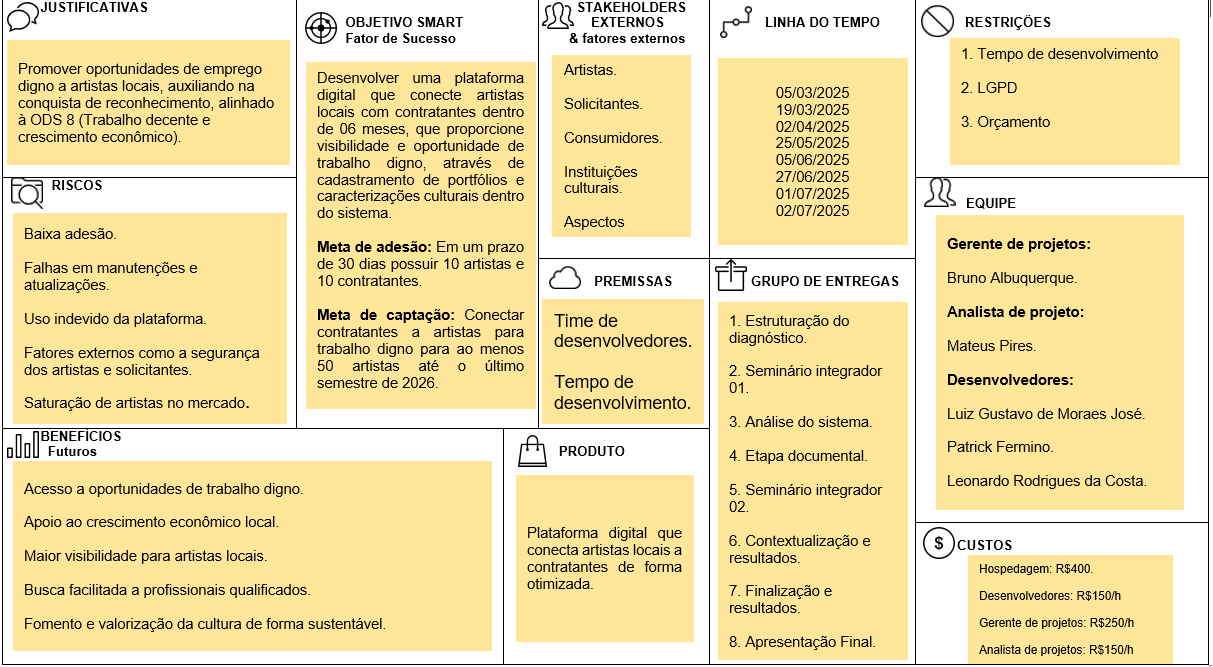
|  |  |
| --- | --- |
| O que? | Desenvolver uma plataforma web para conectar e divulgar artistas ao público. |
| Por que? | Porque os artistas têm dificuldade de conseguir trabalho. |
| Por que? | Porque não existe um canal formal para contratação de artistas. |
| Por que? | Porque geralmente os artistas trabalham informalmente. |
| Por que? | Porque a demanda do mercado se estabelece de maneira informal |
| Por que? | Porque atualmente é mais vantajoso para ambas as partes estabelecerem contratos informais. |
| Causa Raiz: | A carência de uma estruturação e amparo para comunidade artística faz com que o meio se desenvolva de maneira fragmentada e informal, prejudicando o crescimento e garantia de trabalho digno para os profissionais do meio, bem como prejudicar a garantia de uma execução exemplar para contratantes que têm interesse nos serviços que artistas oferecem. |

Fonte: elaborado pelos autores (2025)

## 

|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE B – Estruturação do projeto | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autores** | **Notas da Revisão** |
| 1.0 | 06/05/2025 | Luiz Gustavo e Mateus Pires | Criação |

# Figura 1 – Project Model Canvas – PM Canvas**.**



Fonte: elaborado pelos autores (2025)

|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE C – Requisitos funcionais e não funcionais | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autores** | **Notas da Revisão** |
| 1.0 | 06/05/2025 | Leonardo, Patrick, Bruno, Mateus e Luiz Gustavo | Criação |

Tabela 1 - Requisitos funcionais.

|  |  |  |
| --- | --- | --- |
| Código | Identificação | Objetivo |
| [RF01] | Cadastro de administradores | O sistema deve permitir o cadastro de administradores. |
| [RF02] | Cadastro de artistas | O sistema deve permitir que usuários se cadastrem como artistas fornecendo dados como : nome, CPF, e-mail. data de nascimento, telefone e senha. |
| [RF03] | Cadastro de contratantes | O sistema deve permitir que usuários se cadastrem na plataforma como contratantes fornecendo dados como: nome, documento (CPF ou CPNJ), razão social, data de nascimento, e-mail, telefone e senha. |
| [RF04] | Cadastro de portfólio | O sistema deve permitir que usuários (artistas) complementem seu cadastro com um portfólio, onde poderão se incluir em uma ou mais categorias, informar uma descrição mais completa sobre seus trabalhos. |
| [RF05] | Cadastro de postagens no portfólio | O sistema deve permitir que usuários(artistas) realizem publicações no seu portfólio. As publicações podem conter imagens para que visualizem seu trabalho de maneira prática. |
| [RF06] | Área administrativa | O sistema deve possibilitar que somente usuários administradores acessem a área administrativa. |
| [RF07] | Parametrizações do site. | A área administrativa deve conter uma página de parametrizações que compõem a aparência do *site*, podendo configurar o *banner* da tela *home*, informações no footer e a cor primária do *site*. |
| [RF08] | Indicadores do sistema | A área administrativa deve conter uma página com indicadores para visualizar dados como: Quantidade de *feedbacks* registrados, quantidade de artistas cadastrados, quantidade de contratantes cadastrados, média geral de *feedbacks* e quantidade de acessos. |
| [RF09] | Filtragem dos indicadores | O sistema deve permitir que os usuários administradores filtrem os indicadores que de acordo com período(data de/até) , Categoria e cidade. |
| [RF10] | Exibir artistas cadastrados | O sistema deve permitir que qualquer usuário, mesmo que não tenha *login*, visualize todos os artistas cadastrados na plataforma *web* através de uma página intitulada “Artistas” . |
| [RF11] | Exibir contratantes cadastrados | O sistema deve permitir que qualquer usuário, mesmo que não tenha *login,* visualize todos os usuários que já contrataram um artista na plataforma através de uma página intitulada “Contratantes” . |
| [RF12] | Filtrar pesquisa | Os usuários poderão ter a possibilidade de filtrar a visualização dos artistas cadastrados na plataforma de acordo com a Categoria, |
| [RF13] | Editar informações | O sistema deve permitir que os usuários possam editar suas informações a qualquer momento desde que estejam logados. |
| [RF14] | Notificação de e-mail | Permitir que os artistas recebam e-mails para que os contratantes possam contatá-los |
| [RF15] | Sistema de avaliação | O sistema permite que os contratantes avaliem os artistas contratados |
| [RF16] | Sistema de avaliação | O sistema permite que os artistas avaliem os contratantes |
| [RF17] | Definir status de usuário | O sistema deve permitir a ativação e inativação de usuários no banco de dados |
| [RF18] | Pesquisa sem Login | O sistema permite que os usuários façam pesquisa sem estarem logados |
| [RF19] | Recuperar Senha | O sistema deve permitir que os usuários recuperem sua senha via *e-mail* cadastrado caso seja necessário |
| [RF20] | Solicitar proposta de contrato | O sistema permite que o contratante envie uma solicitação de proposta de contrato para o artista |

Fonte: elaborado pelos autores (2025)

Tabela 2 - Requisitos não funcionais

|  |  |  |
| --- | --- | --- |
| Código | Identificação | Objetivo |
| [RNF001] | Segurança da aplicação | A aplicação deve estar ativa com o protocolo de segurança HTTPS |
| [RNF002] | Usabilidade | A interface da plataforma deve ser intuitiva. Artistas e Contratantes devem ser capazes de se cadastrarem, buscar e ter interações com facilidades |
| [RNF003] | Usabilidade | A plataforma deve ser responsiva, ou seja, deve funcionar bem em dispositivos móveis *(smartphones* e *tablets*) além de *desktops*. |
| [RNF004] | Desempenho | A Plataforma deve carregar completamente |
| [RNF005] | Disponibilidade | A plataforma deve ficar disponível integralmente. |
| [RNF006] | Conexão | Para acessar a plataforma é necessário possuir conexão com a *internet.* |
| [RNF007] | Cadastro de Categorias | Apenas usuários administradores poderão cadastrar, editar ou excluir categorias artísticas. |
| [RNF008] | Segurança | O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados. |

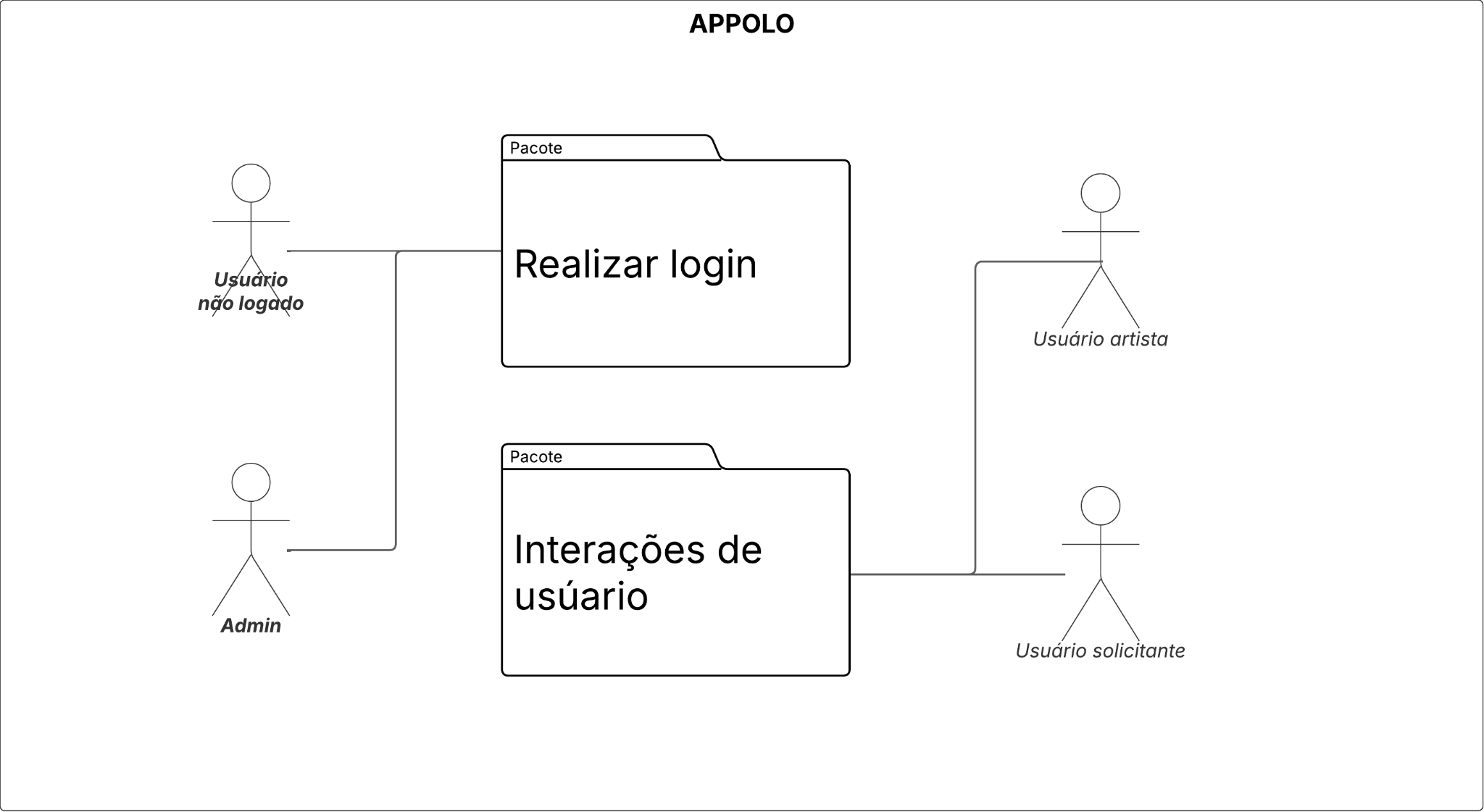
Fonte: elaborado pelos autores (2025)

|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE D – Diagrama de caso de Uso | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autor** | **Notas da Revisão** |
| 1.0 | 06/05/2025 | Mateus V. Pires | Criação |

# 

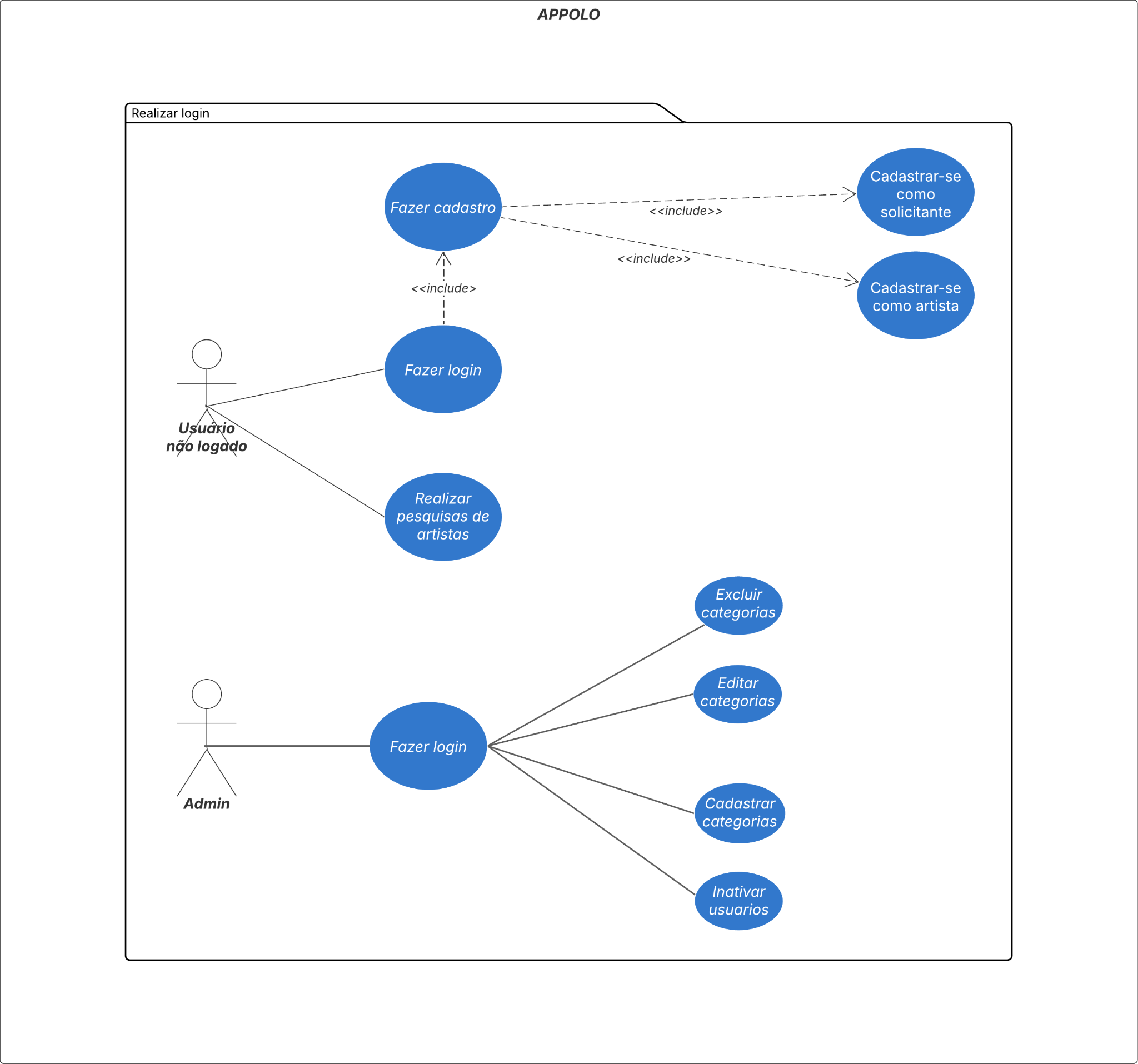
# 

# Figura 2 - Diagrama de caso de uso (Resumo).

****

Fonte: elaborado pelos autores (2025)

# Figura 3 - Diagrama de caso de uso (Realizar *login)*

****

Fonte: elaborado pelos autores (2025)

# 

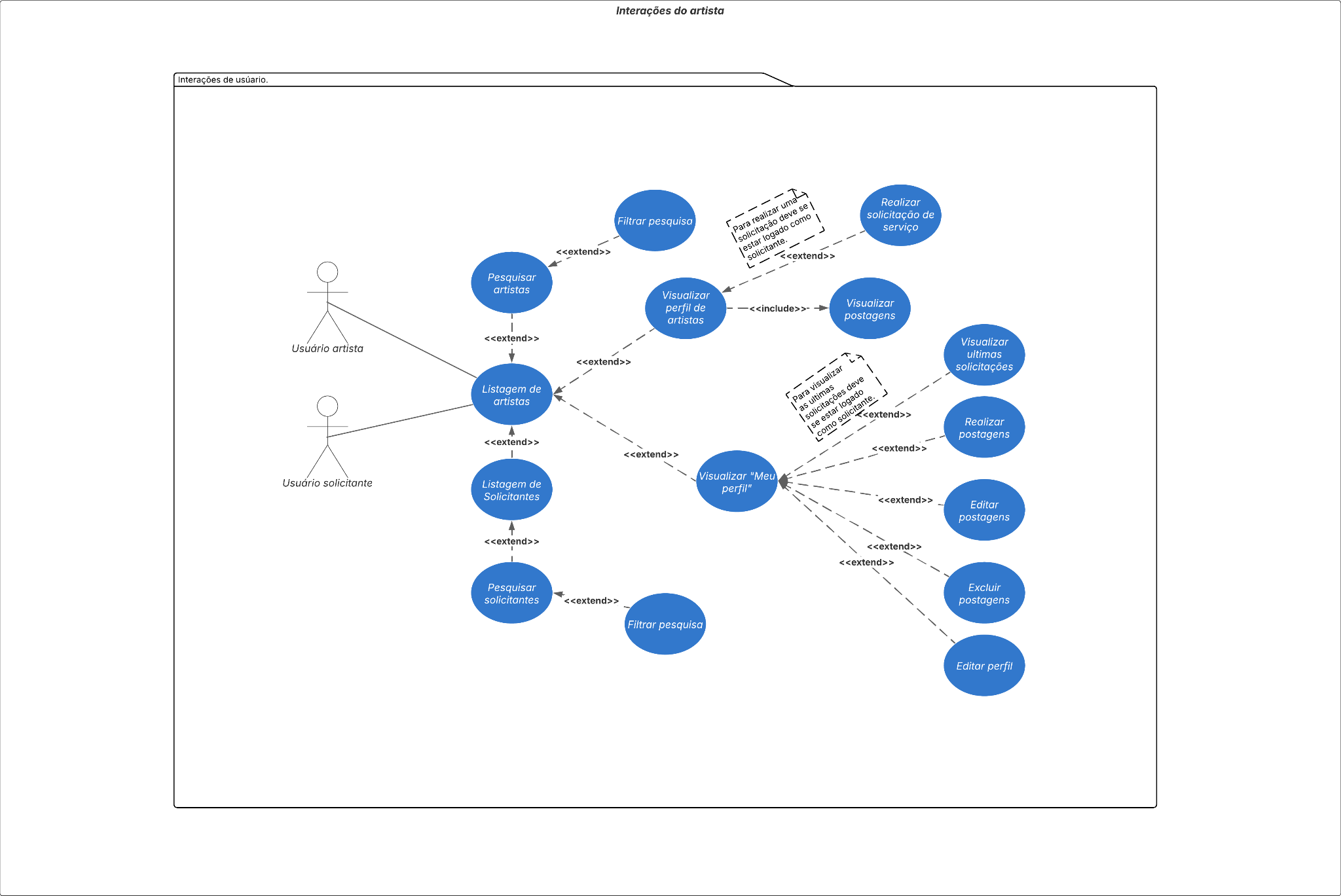
# 

# 

# 

# 

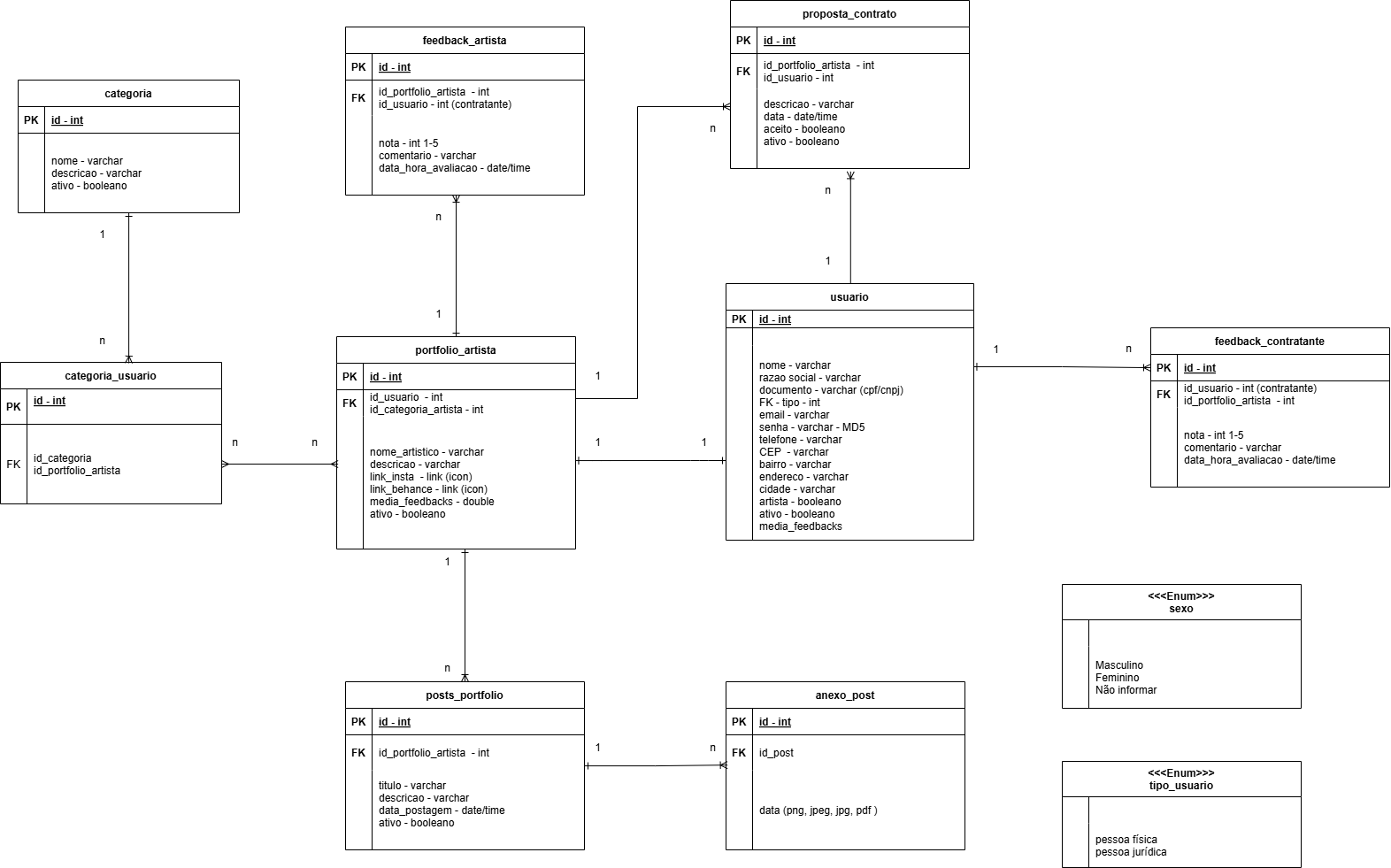
# Figura 4 - Diagrama de caso de uso (Interações do artista)



Fonte: elaborado pelos autores (2025)

|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE E – Modelo E&R Lógico - DER | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autor** | **Notas da Revisão** |
| 1.0 | 06/05/2025 | Bruno Jucoski | Criação |

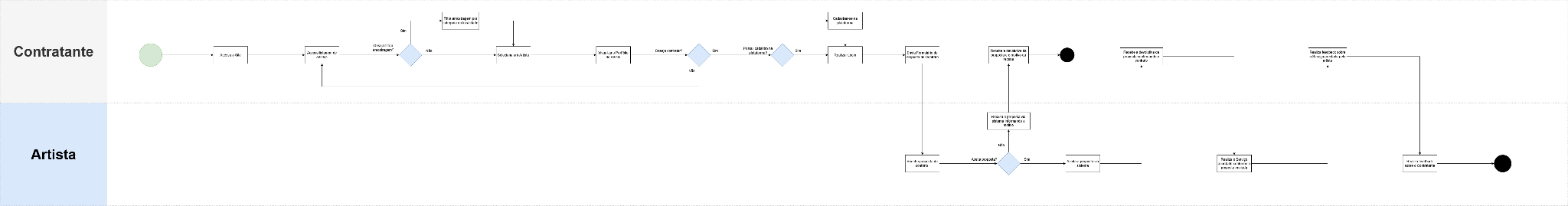
# Figura 5 - Diagrama entidades relacionamentos



Fonte: elaborado pelos autores (2025)

|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE F – Fluxo BPMN | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autor** | **Notas da Revisão** |
| 1.0 | 06/05/2025 | Bruno Jucoski | Criação |

# Figura 6 - Fluxo para contratação de um artista - [LINK](https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=FluxoContrata%C3%A7%C3%A3oArtista.drawio&dark=auto#R%3Cmxfile%3E%3Cdiagram%20name%3D%22P%C3%A1gina-1%22%20id%3D%222g4l2DIUceB_GjQEWBBj%22%3E7V1tc5s4EP41%2FpgO4sXGHxM3ae%2Bm7WTqm971vskg22oAcUJO7Pz6kwDZgGRDUpuXtGkzRWsBYvXs7rMr4Y6sWbj9QGG8%2Fkx8FIxMw9%2BOrPcj0xzbxoT%2FIyS7TGK7NsgkK4r9TFYQzPEzyoVGLt1gHyWljoyQ)



Fonte: elaborado pelos autores (2025)

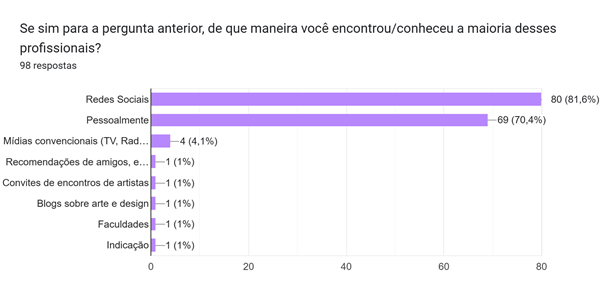
|  |  |  |  |
| --- | --- | --- | --- |
| APÊNDICE G – Questionário | | |  |
| **Nome do Projeto: Appolo** | | |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autor** | **Notas da Revisão** |
| 1.0 | 20/05/2025 | Bruno Jucoski | Criação |

# Gráfico 1 - Pergunta 1



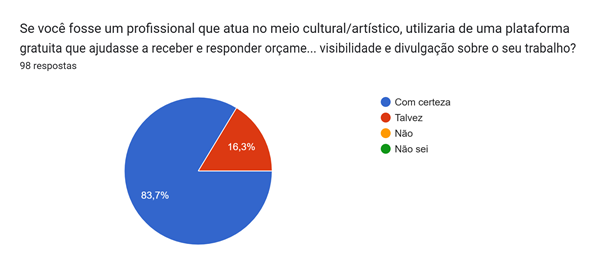
Fonte: elaborado pelos autores (2025)

# Gráfico 2 - Pergunta 2



Fonte: elaborado pelos autores (2025)

# Gráfico 3 - Pergunta 3



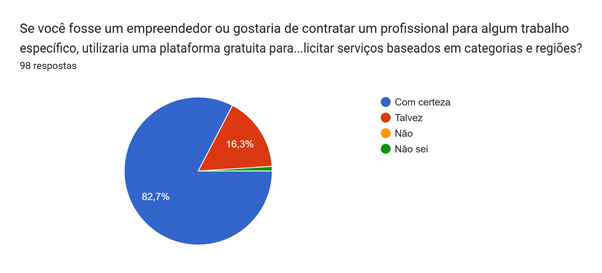
Fonte: elaborado pelos autores (2025)

# Gráfico 4 - Pergunta 4



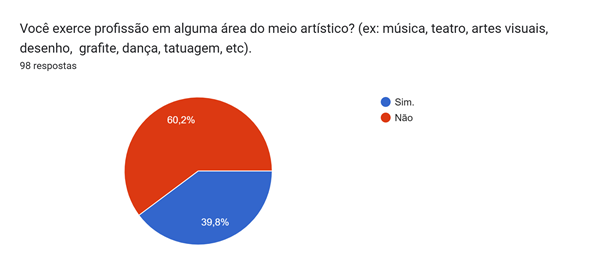
Fonte: elaborado pelos autores (2025)

# Gráfico 5 - Pergunta 5



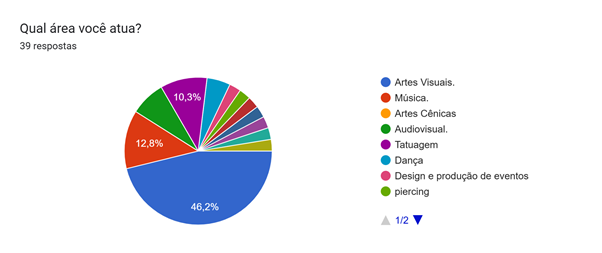
Fonte: elaborado pelos autores (2025)

# Gráfico 6 - Pergunta 6



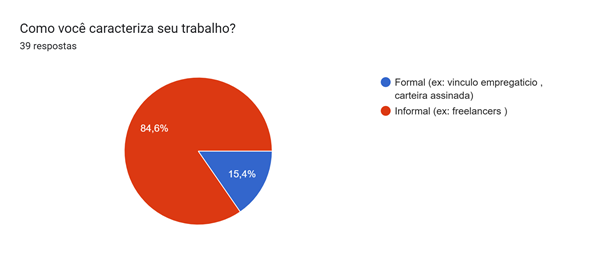
Fonte: elaborado pelos autores (2025)

# Gráfico 7 - Pergunta 7



Fonte: elaborado pelos autores (2025)

# Gráfico 8 - Pergunta 8



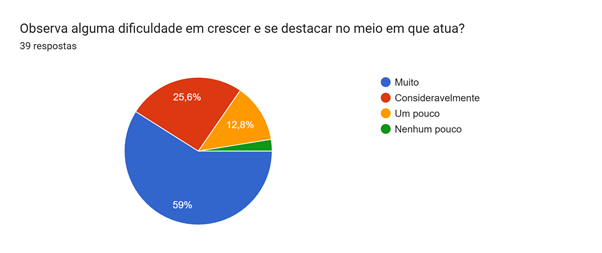
Fonte: elaborado pelos autores (2025)

# Gráfico 9 - Pergunta 9



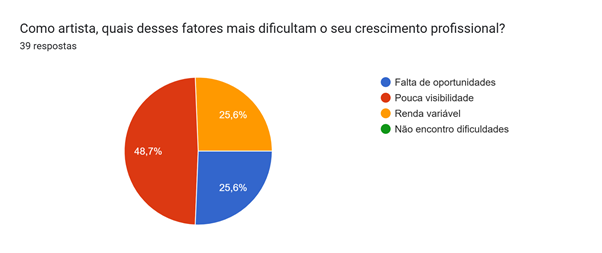
Fonte: elaborado pelos autores (2025)

# Gráfico 10 - Pergunta 10



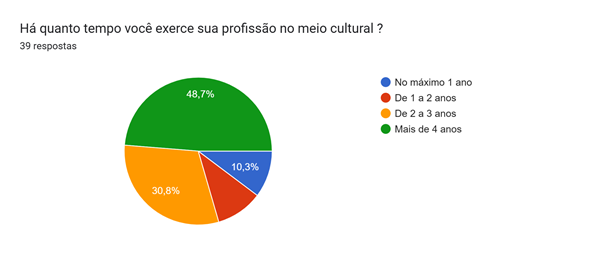
Fonte: elaborado pelos autores (2025)

# Gráfico 11 - Pergunta 11



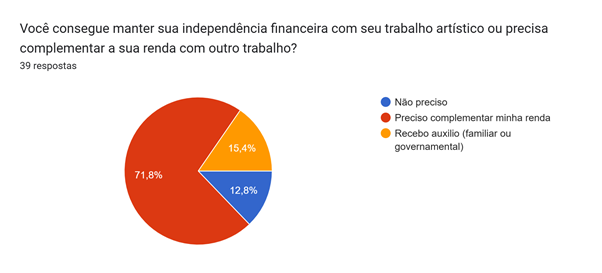
Fonte: elaborado pelos autores (2025)

# Gráfico 12 - Pergunta 12



Fonte: elaborado pelos autores (2025)

# Gráfico 13 - Pergunta 13



Fonte: elaborado pelos autores (2025)