

TÓPICOS EM CIÊNCIAS EXATAS E DA TERRA

VOLUME IX

Frederico Celestino Barbosa

Tópicos em ciências exatas e da Terra

9ª ed.

Piracanjuba-GO
Editora Conhecimento Livre
Piracanjuba-GO

9ª ed.

Dados Internacionais de Catalogação na Publicação (CIP)

Barbosa, Frederico Celestino
B238T Tópicos em ciências exatas e da Terra

/ Frederico Celestino Barbosa. – Piracanjuba-GO

Editora Conhecimento Livre, 2025

65 f.: il

DOI: 10.37423/2025.edcl1144

ISBN: 978-65-5367-680-0

Modo de acesso: World Wide Web

Incluir Bibliografia

1. matemática 2. física 3. química 4. geociências 5. ciências-da-computação I. Barbosa, Frederico Celestino II. Título

CDU: 30

<https://doi.org/10.37423/2025.edcl1144>

O conteúdo dos artigos e sua correção ortográfica são de responsabilidade exclusiva dos seus respectivos autores.

EDITORIA CONHECIMENTO LIVRE

Corpo Editorial

MSc Edson Ribeiro de Britto de Almeida Junior

MSc Humberto Costa

MSc Thays Merçon

MSc Adalberto Zorzo

MSc Taiane Aparecida Ribeiro Nepomoceno

PHD Willian Douglas Guilherme

MSc Andrea Carla Agnes e Silva Pinto

Dr. Walmir Fernandes Pereira

MSc Edisio Alves de Aguiar Junior

MSc Rodrigo Sanchotene Silva

MSc Adriano Pereira da Silva

MSc Frederico Celestino Barbosa

MSc Guilherme Fernando Ribeiro

MSc. Plínio Ferreira Pires

MSc Fabricio Vieira Cavalcante

PHD Marcus Fernando da Silva Praxedes

MSc Simone Buchignani Maigret

Dr. Adilson Tadeu Basquerote

Dra. Thays Zigante Furlan

MSc Camila Concato

PHD Miguel Adriano Inácio

Dra. Anelisa Mota Gregoleti

PHD Jesus Rodrigues Lemos

MSc Gabriela Cristina Borborema Bozzo

MSc Karine Moreira Gomes Sales

Dr. Saulo Cerqueira de Aguiar Soares

MSc Pedro Panhoca da Silva

MSc Helton Rangel Coutinho Junior

MSc Carlos Augusto Zilli

MSc Euvaldo de Sousa Costa Junior

Dra. Suely Lopes de Azevedo

Dr. Francisco Odecio Sales

MSc Ezequiel Martins Ferreira

MSc Eliane Avelina de Azevedo Sampaio

MSc Carlos Eduardo De Oliveira Gontijo

Dr. Rodrigo Couto Santos

Dra. Milena Gaion Malosso

PHD Marcos Pereira Dos Santos

CAPÍTULO 1 5

ANÁLISE ORIENTADA A OBJETOS PARA UM SISTEMA INTEGRADO DE GESTÃO DE CHAMADOS E SUPORTE TÉCNICO BASEADO EM IA

Agatha Kethyllen Ribeiro de Oliveira

Camilly Vitória Menezes Rodrigues

Guilherme dos Santos Damascena

João Lucas Alves Gregório

Micaias Leonardo Costa Viola

Vanessa Helena Bandeli Scarin

Flávio da Silva Waltz

Roberto Cordeiro Waltz

DOI 10.37423/250710215

CAPÍTULO 2 34

DESCOBRINDO A MATEMÁTICA: ATRAVÉS DE OLHARES ESPECIAIS

Fernanda Aparecida Pereira

Ana Maria Rodrigues Pinto

Eliane Aparecida Rocha da Silva

Elisângela Gouvêa de Souza

Maria Iraci Malaquias

Maria Ferreira da Silva

Patricia Aparecida Gonçalves

Patricia Vicente de Oliveira

Silvane Alves de Souza

Tamiris Silvério Braga

DOI 10.37423/250810239

CAPÍTULO 3 40

NÚMEROS PERFEITOS E PRIMOS DE MERSENNE

Carollyne da Cruz Almeida

Thais Silva do Nascimento

Elaynne Xavier Souza Araujo

DOI 10.37423/250810249

Capítulo 1



10.37423/250710215

ANÁLISE ORIENTADA A OBJETOS PARA UM SISTEMA INTEGRADO DE GESTÃO DE CHAMADOS E SUPORTE TÉCNICO BASEADO EM IA

Agatha Kethyllen Ribeiro de Oliveira

Universidade Paulista - UNIP

Camilly Vitória Menezes Rodrigues

Universidade Paulista - UNIP

Guilherme dos Santos Damascena

Universidade Paulista - UNIP

João Lucas Alves Gregório

Universidade Paulista - UNIP

Micaías Leonardo Costa Viola

Universidade Paulista - UNIP

Vanessa Helena Bandeli Scarin

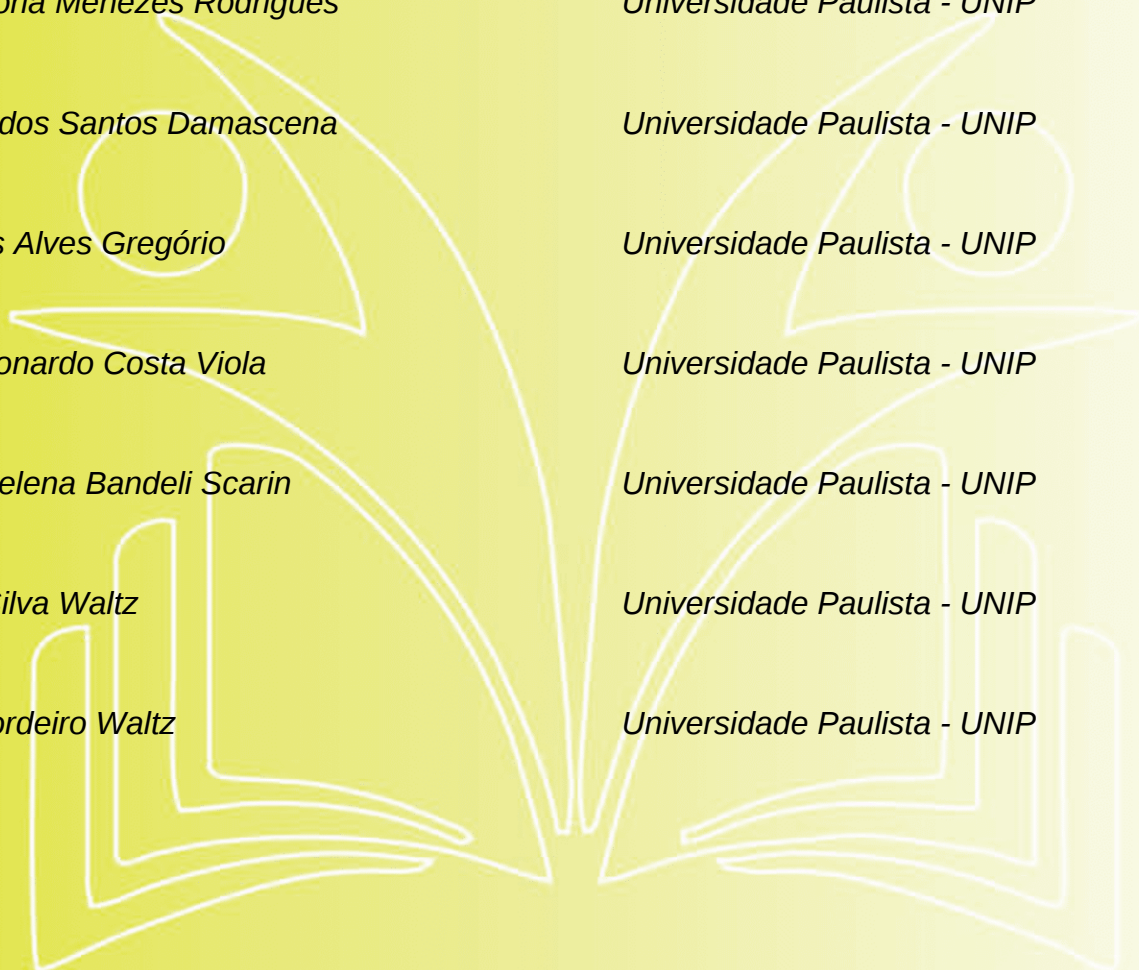
Universidade Paulista - UNIP

Flávio da Silva Waltz

Universidade Paulista - UNIP

Roberto Cordeiro Waltz

Universidade Paulista - UNIP



Resumo: O presente estudo teve como finalidade aplicar a metodologia orientada a objetos para projetar um software de gerenciamento de chamados, com a intenção de otimizar a seção de suporte técnico de uma empresa, com soluções que aumentem a eficiência e a qualidade do suporte, sendo um chamado um registro formal de um pedido de ajuda, suporte ou serviço feito por um usuário para uma equipe responsável, geralmente de suporte técnico. O sistema projetado tem como função centralizar, priorizar e encaminhar as solicitações para os departamentos responsáveis por solucionar os problemas, com o objetivo de viabilizar o atendimento às necessidades dos clientes de forma rápida e eficiente. Nesse escopo, pretende-se que seja implementada e treinada uma Inteligência Artificial (IA) a fim de automatizar os processos de priorização e de solução dos problemas relatados, oferecendo respostas precisas e eficazes. Foi elaborado um protótipo para a apresentação inicial do produto com o fim de abordar os requisitos, validar se o projeto está de acordo com as expectativas do cliente e viabilizar seu desenvolvimento futuro. Profissionais que atuam na área de suporte técnico foram entrevistados para tornar mais eficiente a elicitación de requisitos, e as metodologias utilizadas foram baseadas em análise documental, utilização de artefatos de análise de sistemas orientada a objetos e pesquisas bibliográficas para fundamentar o cenário atual de suporte. Ao fim da realização do planejamento, análise, modelagem do sistema e do protótipo interativo, foram avaliadas features adicionais para desdobramentos futuros do projeto.

Palavras-chave: Gerenciamento de chamados, inteligência artificial, suporte técnico, análise orientada a objetos.

1. INTRODUÇÃO

Conforme pesquisa realizada pela TNS Research, empresas que realizam investimento em tecnologia têm uma porcentagem de aumento de receita e de crescimento 60% maior do que aquelas que optam por não investir (ELS SISTEMAS, 2024). Neste cenário, a utilização de sistemas no ambiente de trabalho entra em ascensão, facilitando assim o gerenciamento de fluxo de informações e otimizando processos.

Para aplicar sistemas informatizados em uma empresa, faz-se necessário a inclusão do setor de gerenciamento de chamados através de uma ferramenta digital. Este irá se concentrar em centralizar, organizar, priorizar e solucionar as solicitações de ajuda realizadas pelos clientes para o departamento de suporte técnico, também conhecidos como chamados (TEIXEIRA, 2024).

Num contexto de mudanças constantes e grande fluxo de informações diárias, surgem inconsistências no atendimento ao cliente humanizado, como tempo elevado de espera, critérios subjetivos para priorização e sobrecarga da equipe especializada em suporte, e como a resolução de problemas de forma rápida e eficiente é a chave de retenção de clientes e posicionamento da marca (SEBRAE, 2024), sugere-se a implementação de uma Inteligência Artificial (IA) na ferramenta digital.

A Inteligência Artificial é um conjunto de tecnologias que permite que os computadores executem diversas funções avançadas. Através do desenvolvimento de algoritmos modelados conforme os processos de decisões do cérebro humano, é possível que a IA aprenda com os dados disponíveis a classificar, prever e a fornecer sugestões cada vez mais precisas, conforme o passar do tempo e a precisão de seu banco de dados (IBM, 2023).

Segundo pesquisa realizada pela Zendesk CX Trends 2023, 72% dos clientes querem receber atendimento imediato. Somado a isso, 74% dos entrevistados acreditam que a IA deve melhorar a eficiência do atendimento ao cliente. Em contrapartida, 47% dos entrevistados pela Capterra 2022 acreditam que a IA pode gerar frustrações ao ser repetitiva ou não conseguir solucionar problemas complexos.

Partindo deste pressuposto, através da análise orientada a objetos, propõe-se um sistema integrado para gestão de chamados e suporte técnico com base em Inteligência Artificial, com o intuito de proporcionar para este setor otimização ao implementar funções de priorização, categorização e sugestões de resolução realizadas pela IA, permitindo, entretanto, que problemas sejam encaminhados para o atendimento humano quando necessário, visando melhorar a experiência do

cliente/usuário em situações complexas, podendo assim diminuir a sobrecarga do setor de suporte e gerar praticidade para o usuário.

1.1 OBJETIVO GERAL

Realizar o levantamento e a análise dos requisitos para a criação de um sistema de suporte técnico inteligente que utilize Inteligência Artificial (IA) para triagem e sugestão inicial de soluções de chamados, visando otimizar o tempo de resolução e evitar sobrecarga dos técnicos de suporte. Os dados pessoais serão armazenados em um banco de dados que contemplará os pressupostos da Lei Geral de Proteção de Dados (LGPD).

1.2 OBJETIVOS ESPECÍFICOS

Para que o objetivo geral proposto seja alcançado, serão considerados os seguintes objetivos específicos:

- Descrever os requisitos funcionais e não funcionais através da elicitação;
- Através da UML (Unified Modeling Language), modelar os requisitos do sistema através de diagramas de caso de uso, diagrama de classe, diagrama de sequência e diagrama de implantação;
- Desenvolver o diagrama de entidade-relacionamento, juntamente com o modelo conceitual descritivo, lógico e dicionário de dados;
- Gerar um script de planejamento do banco de dados em SQL Server;
- Validar a usabilidade do sistema através de um protótipo de alta fidelidade da versão desktop da interface;
- Criar regras de negócio detalhadas para a empresa;
- Elaborar uma planilha de testes unitários;
- Descrever a implementação de uma IA para triagem, encaminhamento e/ou apresentação de solução inicial para os chamados abertos.

2. CARACTERIZAÇÃO DA EMPRESA (SUPORTE TÉCNICO)

Para o presente estudo, uma empresa fictícia foi elaborada, sendo definida como de médio porte localizada no Vale do Paraíba Paulista, atuante no ramo contábil, que apresenta a seguinte estrutura organizacional: Departamento Pessoal, Recursos Humanos, Área Contábil, Área Fiscal, Financeiro, Comercial, Administrativo e TI (Tecnologia da Informação). Aproximadamente 80% de seus funcionários são usuários de microcomputadores interligados em rede, e o perfil do usuário é de baixo

a médio conhecimento em informática, utilizando equipamentos e ferramentas para atividades diversas como: softwares de gerenciamento administrativo (ERP – Enterprise Resource Planning), financeiro (contábeis e bancários), comerciais (CRM – Customer Relationship Management), de desenvolvimento de artes (Adobe Photoshop, Adobe Illustrator) e aplicações como editores de texto, planilhas, gráficos e apresentações, navegadores de Internet e serviços de e-mail.

O setor de TI é responsável por toda a estrutura de informática da empresa e há um histórico de alta demanda por atendimentos, agravada em horários de pico, gerando problemas como demora no atendimento, ineficiência no levantamento de dados e na geração de relatórios, falha na priorização de tarefas, dúvidas quanto à eficiência do setor e insatisfação geral (tanto dos requerentes quanto da equipe de suporte). O atendimento é solicitado via e-mail e continuado pelo aplicativo de mensagens WhatsApp, por ligações telefônicas e presencialmente, se necessário.

Na evidência da necessidade de organizar este processo, a empresa deseja adotar um sistema integrado, no qual os colaboradores possam registrar suas solicitações e utilizar uma ferramenta de Inteligência Artificial (IA) para sugerir soluções automáticas ou encaminhar ao técnico adequado com base no histórico de chamados e complexidade do problema. Partindo desta premissa, se inicia o desenvolvimento do sistema integrado para gestão de chamados e suporte técnico baseado em IA descrito neste estudo e denominado “HELPBOX” (podendo ser personalizado e comercializado para outras organizações).

3. ANÁLISE DE SISTEMAS ORIENTADA A OBJETOS

De acordo com Sommerville (2011), uma diferença importante entre o desenvolvimento de software profissional e amador é a complementação do programa em si com toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente.

Apesar disso, alguns fundamentos se aplicam a todos os tipos de software. Dentre estes fundamentos, está a especificação de requisitos, que também funciona como um gerenciamento de expectativas do cliente. Os requisitos de um sistema são as descrições do que um sistema deve fazer, os serviços oferecidos e as restrições a seu funcionamento (SOMMERVILLE, 2011).

Wazlawick (2014) expõe a necessidade de, além de organizar a estrutura de um sistema em camadas e módulos, organizar o código implementado dentro deles. Os programadores devem aprender as técnicas que podem levar à construção de código reusável e com baixo acoplamento.

Os conceitos básicos da UML (do inglês, Unified Modeling Language) incluem diversos tipos de diagramas que representam diferentes aspectos do sistema em desenvolvimento. Esses diagramas podem ser agrupados em duas categorias principais: diagramas estruturais e diagramas comportamentais, como descrito por Booch et al. (2005).

Diagramas estruturais abordam diagrama de classes, diagrama de objetos, e diagrama de componentes; enquanto diagramas comportamentais seriam o diagrama de casos de uso, diagrama de sequência e diagrama de atividade, por exemplo (BOOCH et al., 2005).

Para Sommerville (2011), casos de uso são uma técnica de descoberta de requisitos, identificando os atores (que podem ser pessoas ou outro sistema) envolvidos em uma interação e nomeando-a. São documentados em seu diagrama com informações adicionais que descrevem a interação com o sistema.

Os casos de uso expandidos podem ser utilizados como comandos (métodos que são ativados por um evento de sistema, ou seja, como uma reação a uma ação do usuário) e consultas (verificação de informação já armazenada) de sistema. A sequência de eventos e respostas em um caso de uso são representados por um diagrama de sequência, com elementos como atores, interfaces do sistema e objetos (WAZLAWICK, 2014).

Já diagramas de classe serão definidos em diversos níveis de detalhamento. Em suma, mostra as classes (definição geral de um tipo de objeto do sistema) e as associações entre elas. Inicialmente, os objetos representam algo do mundo real, entretanto, enquanto uma aplicação é desenvolvida, pode ser necessário definir objetos adicionais de implementação que são usados para fornecer a funcionalidade requerida do sistema.

Por fim, os diagramas de implantação mostram como os componentes de software são distribuídos em diferentes plataformas de hardware (SOMMERVILLE, 2005).

Para Wazlawick (2014), nem todos os diagramas devem ser usados durante o desenvolvimento de um sistema, apenas aqueles que representam informações úteis para o projeto são recomendados. Neste projeto, foram desenvolvidos diagramas de caso de uso, de sequência, de classes e de implementação.

4. ENGENHARIA DE SOFTWARE

A Engenharia de Software é uma área da computação que trata da aplicação de princípios de engenharia no desenvolvimento, operação e manutenção de sistemas de software. Processo de

software é um conjunto de atividades relacionadas que levam à produção de um produto de software (SOMMERVILLE, 2019). Envolve uma série de ações estratégicas e bem estruturadas com o objetivo de planejar, criar ou adquirir, testar e implementar o sistema.

A análise de requisitos em si é um processo que engloba o estudo das necessidades do usuário, para que uma definição correta/completa seja aplicada (COOPERSYSTEM, 2023). Os requisitos funcionais referem-se às principais funcionalidades que o software deve oferecer para cumprir seu propósito, isto é, as operações que o usuário precisa realizar. Já os requisitos não funcionais estão relacionados a aspectos como segurança, manutenção e desempenho do sistema.

Qualidade pode ser definida como um método gerencial que, através de procedimentos disseminados por toda a organização, busca garantir um produto que satisfaça às expectativas do cliente, dentro daquilo que foi acordado inicialmente (DEV MEDIA, 2025), como definido na ISO/IEC 25010, que define um modelo de qualidade de produtos de software.

É compreensível que sistemas de software estejam sujeitos aos mais variados tipos de erros e inconsistências. Para evitar que tais erros atinjam os usuários finais e causem prejuízos, é fundamental introduzir atividades de teste em projetos de desenvolvimento de software (VALENTE, 2022). A elaboração de uma tabela de testes, por exemplo, permite organizar de maneira estruturada os elementos que devem ser verificados, os procedimentos de execução e as evidências necessárias para comprovar o correto funcionamento do sistema. Em casos de falhas (dentre outros), o processo de alteração do software após a entrega é frequentemente chamado de manutenção de software (SOMMERVILLE, 2019). O programa precisa ser aperfeiçoado continuamente para corrigir erros, adaptar-se a novas tecnologias e atender a novas demandas.

Uma das alternativas para gerenciamento de projetos é o método ágil Scrum, marcado pela iteração e incrementação. É definido pelo conjunto de papéis profissionais (Product Owner, Scrum Master e time de desenvolvedores), artefatos como Product Backlog, Sprint Backlog e Burndown Chart e eventos (Sprint Planning, Sprints, Dailys, Sprint Review e Retrospective). Sua utilização minimiza o risco de desenvolver produtos que não atendem as necessidades dos clientes, e também proporciona um fluxo contínuo de trabalho, fomentando o compromisso da equipe (VALENTE, 2020).

5. PROGRAMAÇÃO ORIENTADA A OBJETOS

Atualmente, existem dois tipos de programação considerados usuais, a estruturada e a orientada a objetos. O primeiro paradigma é centrado em funções e no controle de fluxo com etapas lógicas

lineares, enquanto o segundo foca em objetos e classes que interagem entre si através de mensagens, aproximando as estruturas de um programa à estrutura das coisas do mundo real (PROGRAMAE, 2024).

De acordo com dados históricos, os estudos sobre programação orientada a objetos surgiram no final da década de 60 no cenário da “crise do software”. Esse paradigma não possui um único criador, mas sabe-se que a linguagem Simula 67 foi a primeira a introduzir os conceitos de objetos, classes e herança. Esses conceitos foram aprimorados por meio da linguagem Smalltalk, desenvolvida no início da década de 70 pelo Alan Kay, conhecido como um dos pioneiros da POO, no Xerox PARC; mas só ganharam destaque a partir da década de 80 ao serem aplicados nas linguagens Java e C++ (METROPOLE, 2022).

Dentre as vantagens fornecidas por esse paradigma, encontram-se a modelagem do mundo real, que torna o desenvolvimento mais intuitivo e facilita a comunicação interna; a reutilização do código, pois através da herança e do polimorfismo é possível utilizar métodos e classes já existentes em outro contexto; a manutenibilidade simplificada, a possibilidade de desenvolvimento colaborativo, a extensibilidade e a rastreabilidade (BONATINI, 2023).

Em Programação Orientada a Objetos (POO), as classes são estruturas que atuam como moldes ao agrupar itens com as mesmas propriedades e funcionalidades e por meio destas criam-se os objetos, que são as instâncias das classes. Os objetos terão os atributos, que correspondem às suas características; os métodos, que representam as ações e os comportamentos que eles poderão realizar e esses apresentam níveis de visibilidade, que determinam o acesso às informações, podendo ser público, privado ou protegido (NORMANDO, 2024).

Neste cenário surgem os seguintes conceitos cruciais, conhecidos como pilares da programação orientada a objetos: abstração, herança, polimorfismo e encapsulamento.

A abstração visa à simplificação ao modelar comportamentos e características concentrando-se apenas nas informações do objeto que são essenciais e relevantes dentro de um contexto. A herança fornece a funcionalidade da reutilização de métodos e atributos a partir de uma hierarquização de classes. O polimorfismo promove a capacidade de um objeto executar diferentes comportamentos a partir do mesmo método em situações distintas. E o encapsulamento visa controlar o acesso aos métodos e atributos da classe através dos níveis de visibilidade, permitindo a ocultação de dados e da estrutura; por meio desses conceitos, são gerados os benefícios da utilização do paradigma abordado (SILVA, 2023).

6. PROJETO DE INTERFACE COM O USUÁRIO

Segundo a descrição apresentada no site da International Organization for Standardization (ISO) para a normativa ISO/IEC 25010, que aborda a temática de qualidade de produto de software, usabilidade se refere a capacidade deste de ser entendido, aprendido, operado e atraente para o usuário, quando este for usado sob determinadas condições específicas. Essa capacidade refletirá na experiência do usuário (UX), ou seja, nas sensações que existirão durante as interações entre o usuário e o sistema.

Neste cenário, a usabilidade é uma entre as seis principais características de qualidade. Junto a ela encontram-se a portabilidade, confiabilidade, eficiência, manutenibilidade e funcionalidade; isto influenciará diretamente na satisfação do usuário, aumentará a produtividade e a conquista e expansão no mercado. Para avaliar este conceito existem diversas técnicas, como por exemplo avaliações heurísticas e testes de usabilidade (CERQUEIRA, 2024).

A avaliação heurística consiste em uma análise prática e rápida de um produto, interface ou serviço; ela é realizada por um especialista em UX baseando-se nas heurísticas de Nielsen. Essas heurísticas abordam aspectos como a eficiência e a importância dos feedbacks, as "saídas de emergência" para ações indesejadas, a padronização conforme o setor, o uso de modelos conceituais, a clareza das mensagens de erro, a flexibilidade nas formas de realizar ações, entre outros aspectos. E, como citado pelo próprio Jakob Nielsen, as 10 heurísticas em si permaneceram relevantes e inalteradas desde 1994 (TEIXEIRA, 2016).

Em suma, a usabilidade é um fator essencial quando analisado pela perspectiva da experiência do usuário, proporcionando um sistema funcional e confiável e gerando a fidelização dos clientes, trazendo como resultado ascensão da marca no mercado.

7. BANCO DE DADOS

Todo sistema quando gerado necessita de um local para que os dados futuramente armazenados nele sejam guardados de forma segura, podendo assim manter a integridade e a segurança do software. Esse local é chamado de Banco de Dados, uma forma de armazenar, virtualmente, dados de um sistema que, quando ligado a um Sistema de Gerenciamento de Banco de Dados, fornecem qualidade ao sistema no qual estão ligados, fornecendo as ferramentas necessárias para manutenção e controle das informações e melhora a organização das informações (ORACLE, 2025).

Existem dois tipos de banco de dados: os relacionais e os não relacionais. Os não relacionais não possuem uma estrutura definida, ou seja, é possível guardar informações de diversas formas, como

por exemplo por documentos e gráficos. Já o relacional, que é utilizado neste projeto, possui forma, sendo distribuída em linhas e colunas que formam tabelas. Cada tabela é uma entidade e cada linha um registro, sendo criado e manipulado por meio da linguagem SQL e possuindo maior eficiência para o acesso de informações (ORACLE, 2025).

Quando se cria uma entidade para o banco de dados, são especificados atributos dela, como por exemplo *"id_cliente"*. Esses atributos são dispostos em colunas no que é chamado de Dicionário de dados, uma tabela que possui várias informações de cada atributo, sendo importante como referência para os desenvolvedores daquele projeto, visando tornar a comunicação mais fácil entre a equipe e diminuindo erros. Esses dicionários são responsáveis por documentar exatamente as informações sobre cada campo, ou atributo, fornecendo uma visão detalhada da estrutura dos dados existentes e facilitando a geração de um script do banco de dados em linguagem SQL. “Uma das importâncias de um DD (Dicionário de Dados) se dá pelo fato de muitas vezes na atividade de modelagem de um PBD (Projeto de Banco de Dados), os diagramas não suprirem informações significativas sobre os dados que devem ser tratados, deixando a interpretação desses diagramas ineficientes, podendo gerar ambiguidades em sua implementação” (BARBOSA, 2012).

Além dele, existe a normalização, é uma forma de normalizar as tabelas para que não ocorram erros durante a elaboração do banco de dados. Existem 6 normalizações, porém as mais utilizadas e necessárias são as três primeiras formas normais, onde a primeira forma normal busca remover atributos que são multivalorados ou compostos, a segunda forma, eliminar dependências parciais, que é quando um atributo depende apenas de uma das Primary Keys, não da chave composta, e existe a terceira, que visa eliminar dependências transitivas, quando um atributo que não é uma chave depende de outro atributo não-chave. Após o processo de normalização, são gerados diagramas de entidade relacionamento (PUIG, 2023).

“Um diagrama entidade relacionamento (ER) é um tipo de fluxograma que ilustra como “entidades”, pessoas, objetos ou conceitos, se relacionam entre si dentro de um sistema” (SILVA, 2021). Eles permitem a criação de scripts de forma mais assertiva e segura, para que posteriormente sejam melhorados e utilizados no SQL Server. Existem três tipos de diagrama ER: o conceitual descritivo, que mostra as os atributos de cada tabela, descreve o relacionamento e suas cardinalidades, o conceitual em forma de diagrama, que irá dispor das informações do formato descritivo, porém em imagem, e o lógico, que é o diagrama ER em formato visual, porém com as especificações de cada tributo, como por exemplo o tipo, e as chaves existentes em cada.

8. DESENVOLVIMENTO

Após a caracterização do cenário na qual o sistema de gerenciamento de chamados será inserido, partiu-se para a elicitac  o de requisitos funcionais e n  o funcionais. Para tanto, foram realizadas pesquisas bibliogr  ficas, consultas em vers  es gratuitas de softwares do ramo j   estabelecidos no mercado e entrevistas com profissionais da   rea de suporte, buscando definir quais features seriam essenciais, como inserir a atividade da Intelig  ncia Artificial no processo e quais seriam as etapas do processo na experi  ncia dos usu  rios.

Com isso, utilizando o software Astah UML vers  o 10.0.0 (lan  ado em 30/10/24), foram desenvolvidos alguns diagramas comportamentais e estruturais pertinentes ao planejamento do desenvolvimento do sistema, que foi denominado HELPBOX. O arquivo final e os artefatos SCRUM criados foram todos disponibilizados no GitHub. A partir das t  cnicas de elicitac  o, as user story obtidas foram utilizadas para definir o Product Backlog, conforme ilustra a Tabela 1.

Tabela 1 – Product Backlog

REQUISITOS FUNCIONAIS (RF)		REQUISITOS N��O FUNCIONAIS (RNF)	
RF001	Autenticar Usu��rio	RNF0001	Reconhecimento e preven��o de erros
RF002	Abrir Chamado	RNF0002	Ajuda e documenta��o
RF003	Preencher Formul��rio	RNF0003	Manutenibilidade e rastreabilidade
RF004	Avaliar Solu��o	RNF0004	Usabilidade
RF005	Verificar Chamado	RNF0005	Hierarquia de acesso
RF006	Cancelar Chamado	RNF0006	Implementa��o de testes
RF007	Categorizar Chamado		
RF008	Sugerir Solu��o		
RF009	Priorizar Chamado		
RF010	Encaminhar Chamado		
RF011	Diagnosticar Chamado		
RF012	Confirmar Solu��o		
RF013	Excluir Chamado		
RF014	Gerar Relat��rio Mensal		
RF015	Gerar Relat��rio Semestral		
RF016	Gerar Relat��rio Anual		
RF017	Exportar Relat��rio		
RF018	Editar Usu��rio		
RF019	Adicionar Usu��rio		
RF020	Excluir Usu��rio		

Fonte: Os autores, 2025.

Junto do Product Backlog, desenvolveu-se os primeiros diagramas UML, de caso de uso, representando os pacotes de funcionalidades, atores e atividades exercidas, disponíveis no link: Diagramas UML

Um caso de uso descritivo é uma forma de documentação na engenharia de software que apresenta, em texto, o comportamento esperado do sistema durante sua interação com usuários ou outros sistemas. Inicialmente, três 3 (três) casos de uso descritivos considerados fundamentais foram elaborados. Estes estão apresentados nas Tabelas 2, 3 e 4.

Tabela 2 – Caso de uso descritivo “Abrir Chamados”.

Caso de uso:	Abrir chamados
Ator:	Cliente
Pré-condições:	O cliente deve estar autenticado no sistema O cliente deve ter acesso à funcionalidade “Abrir Chamado”
Fluxo normal:	1. <u>Autenticar usuário</u> 2. Cliente seleciona opção “Abrir Chamado” na tela inicial 3. Cliente preenche formulário com o máximo de detalhes possíveis 4. Cliente confirma envio do formulário 5. IA fornece sugestão rápida para o problema informado 6. Cliente avalia solução oferecida pela IA
Extensões:	1a. Se dados incorretos, solicitar nova tentativa de autenticação 1b. Se cliente não recorda seus dados de autenticação, orientar cliente para entrar em contato com o administrador 4a. Se formulário vazio ou conter caracteres insuficientes, solicitar novo preenchimento de acordo com a regra não cumprida 4b. Se cliente não confirmar envio do formulário, retornar para a tela inicial 6a. Se a solução foi útil, chamado é encerrado 6b. Se a solução não foi eficiente, IA fará encaminhamento para o técnico do setor.
Pós-condições:	Um novo chamado é registrado no sistema O cliente recebe uma sugestão automática ou tem o chamado encaminhado para análise técnica

Fonte: Os autores, 2025.

A Tabela 2 descreve a relação entre Cliente e o caso de uso “Abrir chamados”, fundamental para o cumprimento da proposta base do sistema de gerenciamento de chamados.

Tabela 3 – Caso de uso descritivo “Verificar Chamado”.

Caso de uso:	Verificar Chamado (Prioridade)
Ator:	Técnico
Pré-condições:	O técnico deve estar autenticado no sistema. Deve haver chamados registrados com níveis de prioridade. O técnico deve ter acesso à funcionalidade “Verificar Chamado”
Fluxo normal:	1. Autenticar usuário 2. Técnico seleciona opção “Verificar Chamado” na tela inicial 3. Técnico seleciona opção “Filtrar” na tela de verificar chamado 4. Técnico seleciona opção “Prioridade” na função “Filtrar” 5. Lista retorna com filtro de prioridade 6. Técnico fornece resposta do chamado 7. Cliente avalia solução oferecida pelo técnico
Extensões:	1a. Se dados incorretos, solicitar nova tentativa de autenticação 1b. Se técnico não recorda seus dados de autenticação, orientar cliente para entrar em contato com o administrado 7a. Se a solução foi útil, chamado é encerrado
Pós-condições:	O chamado recebe uma resposta O cliente é notificado da resposta e pode avaliá-la O chamado pode ser encerrado ou seguir em aberto, conforme avaliação do cliente

Fonte: Os autores, 2025.

A Tabela 3 apresenta descreve a relação entre Técnico e o caso de uso “Verificar chamado”, representando as principais funcionalidades do sistema associadas à verificação de chamados abertos por clientes através de filtros e sua resolução pelo técnico.

A Tabela 4, por fim, descreve a relação entre dois atores (cliente e técnico) e o caso de uso “Autenticar usuário”, enfatizando aspectos de segurança como a necessidade de contato direto com supervisor técnico ou com administrador para recuperação de credenciais perdidas.

Tabela 4 – Caso de uso descritivo “Autenticação de Usuário”.

Caso de uso:	Autenticar Usuário
Ator:	Cliente e técnico
Pré-condições:	O sistema deve estar disponível O ator deve ter um cadastro válido (e-mail e senha) no sistema
Fluxo normal:	Abrir Sistema Ator preenche e-mail na tela de autenticação Ator preenche senha na tela de autenticação Ator confirma autenticação Abre tela de início do sistema

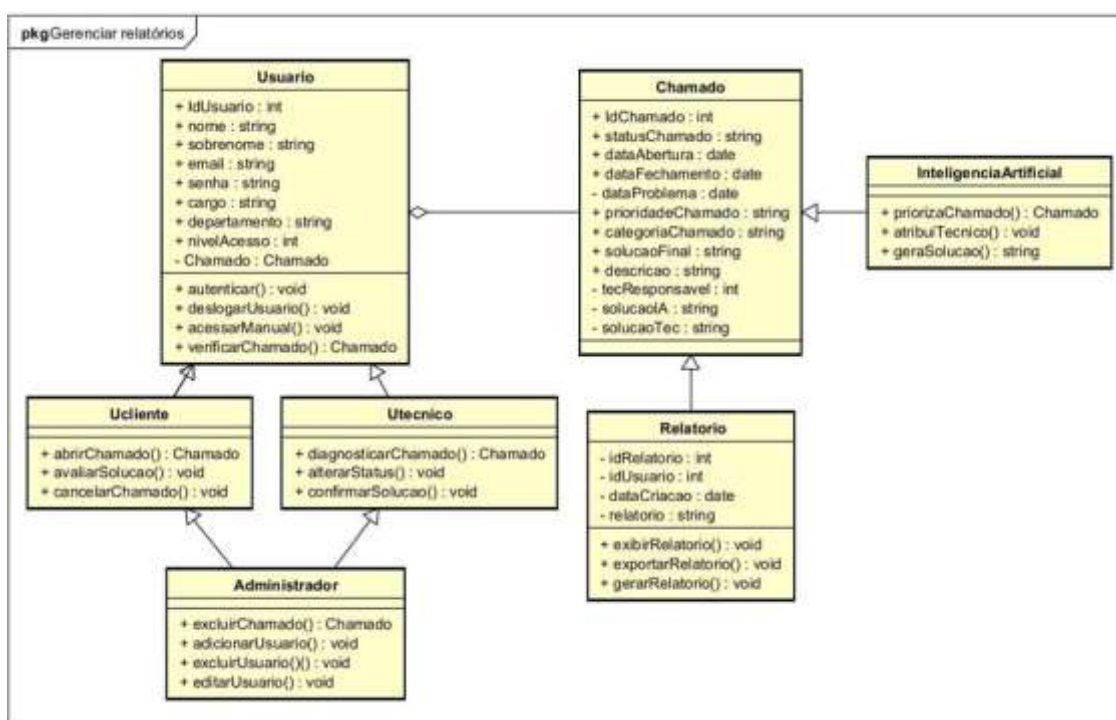
Extensões	1a. Sistema não abre, orientar ator para entrar em contato com o administrador
	2a. Se e-mail incorreto, solicitar nova tentativa de autenticação
	2b. Se ator não recorda seu e-mail de autenticação, orientar ator para entrar em contato com o administrador
	3a. Se senha incorreta, solicitar nova tentativa de autenticação
	3b. Se ator não recorda sua senha de autenticação, orientar ator para entrar em contato com o administrador
Pós-condições:	O ator tem acesso ao sistema autenticado, com funcionalidades liberadas
	Caso não autenticado, permanece na tela de login ou é orientado a recuperar os dados

Fonte: Os autores, 2025.

Cada caso de uso foi submetido à análise para compreensão de quais interfaces e classes estariam envolvidas nas respectivas ações, como elas estarão relacionadas e quais seriam seus métodos e atributos. Com isto, tem-se um guia para uma posterior codificação organizada. Estas informações são consolidadas em diagramas de classes para cada caso de uso, disponibilizados no link: Diagrama de Classes.

Os diagramas de classe oriundos de cada caso de uso também se reúnem em um diagrama de classes geral, que contém todos os elementos e associações, o qual é apresentado na Figura 1.

Figura 1 – Diagrama de classes completo.

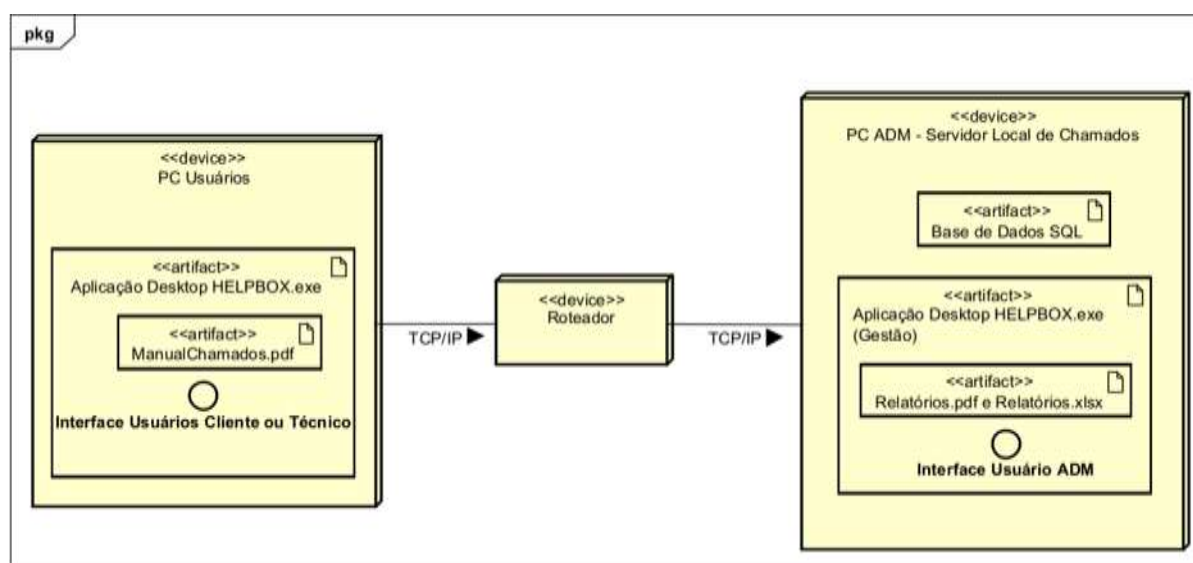


Fonte: Os autores, 2025.

Partindo das classes e interfaces definidas para cada caso de uso, foram construídos diagramas de sequência que demonstram as interações dos atores com estes e suas devidas respostas. No link, estão os mesmos casos de uso utilizados nos diagramas de classe anteriores: Diagrama de sequência

Por fim, descrevendo os componentes necessários para funcionamento do software em seus dispositivos - computadores, servidores, roteadores, protocolos de rede, artefatos – arquivos executáveis, documentos internos ou exportáveis, banco de dados, suas associações e interfaces, produziu-se um diagrama de implantação único, ilustrado na Figura 2.

Figura 2 – Diagrama de implantação.



Fonte: Os autores, 2025.

Em complemento, a elaboração de uma tabela de testes unitários permite organizar de maneira estruturada e hierarquizada os itens que devem ser verificados, os procedimentos de execução e as evidências necessárias para comprovar o correto funcionamento do sistema. Essa prática assegura a verificação e a validação das principais funcionalidades antes da disponibilização do sistema em ambiente produtivo, mitigando riscos de falhas que poderiam resultar em prejuízos, retrabalho ou insatisfação dos usuários. A primeira versão da planilha de testes unitários pode ser acessada em: Testes unitários.

Ademais, a documentação das evidências e das consultas realizadas no banco de dados torna o processo de homologação mais eficiente, seguro e transparente, possibilitando o acompanhamento claro do progresso dos testes por toda a equipe envolvida.

Como primeira parte da tabela, tem-se a numeração do teste, sendo colocado como ID (Identificação), a descrição de teste, que diz qual a ação a ser testada, os passos que deverão ser feitos e o resultado esperado, esse sendo uma expectativa da resposta do sistema à ação executada.

A segunda parte são os comandos ou instruções escritas em linguagem SQL, o envolvimento do banco de dados, as evidências geradas, armazenados como prova do teste, a prioridade do teste a ser feito, o responsável pelo teste e, por fim, o status.

O protótipo do sistema (disponível pelo link: Protótipo Figma) foi desenvolvido para a versão desktop, em resolução 1440 x 1024, no aplicativo Figma e o objetivo foi simular as funcionalidades do sistema a partir do usuário master, que, seguindo as regras de negócio, é o administrador do sistema. A intenção é que o usuário master tenha acesso a todas as funcionalidades como o maior nível hierárquico de acesso ao sistema.

A primeira tela à qual todos os usuários terão acesso será a de login, pois, tendo em vista ser um sistema interno de uma empresa, as funcionalidades estarão disponíveis apenas para usuários autenticados, sendo este requisito demonstrado como essencial no diagrama de caso de uso. Caso o usuário esqueça ou perca a senha, ao selecionar a opção, será informado que essa alteração deve ser solicitada ao administrador, visando proteção de dados.

As funções disponíveis para o administrador são a “Abertura de Chamados”, na qual ele preencherá um formulário dividido em 3 (três) partes, que visa compreender de maneira efetiva o chamado. Após o envio do formulário, o usuário receberá uma solução inicial gerada automaticamente pela Inteligência Artificial e deverá fornecer uma devolutiva se esta foi uma resolução eficiente ou não; em caso de não ter sido, haverá encaminhamento do chamado para um técnico específico.

A funcionalidade “Gerenciar Usuário” permitirá que o administrador adicione, exclua ou altere informações de usuários. A funcionalidade “Verificar Chamados” permitirá acesso aos chamados abertos e ao histórico de chamados. A funcionalidade “Gerar Relatório” permitirá a emissão de relatórios, com base nos filtros aplicados pelo usuário, que poderão ser exportados em formato PDF.

A funcionalidade “Solucionar Chamados” permitirá a verificação de todos os chamados abertos no momento da consulta, a visualização das informações associadas a estes (incluindo solicitante e solução gerada pela IA) e a possibilidade de atender/solucionar o chamado selecionado. Além disso, o usuário terá acesso ao seu perfil ao selecionar a foto no canto superior esquerdo da tela e, ainda,

visualizar o manual do usuário através do botão “Manual do Usuário”, que também poderá ser acessado no menu da tela inicial (também disponível no link: Manual do usuário).

Uma das etapas do processo de abertura de chamados destaca a proposta de solução apresentada pelo módulo de Inteligência Artificial.

A tela de resolução de chamados, que contém informações do solicitante – nome, ID, cargo e setor – e a descrição do problema. Também são destacados o nível de prioridade do chamado, data de abertura e categoria do problema. A proposta de solução gerada pela IA e o status do próprio chamado são informados ao usuário. Por fim, o técnico responsável tem a opção de apresentar uma nova solução para o problema e salvar suas alterações, o que dará andamento ao tratamento do chamado.

O protótipo foi construído visando as heurísticas de Nielsen (1994) e prezando pela usabilidade. Portanto, há prevenção de erros: para todas as ações críticas no sistema, há um pedido de confirmação da ação e há ajuda e documentação através do manual. O usuário Cliente terá acesso às funções de “Abrir Chamado”, “Verificar Chamado” e Manual do Usuário. Enquanto o usuário Técnico terá acesso às funções de “Solucionar Chamados” e Manual do Usuário.

Para início da modelagem do banco de dados, foram definidas as seguintes regras de negócio:

1. O Administrador é um usuário master.
2. O Cliente pode somente gerenciar seus próprios chamados.
3. O Técnico pode somente gerenciar resoluções atribuídas a ele.
4. Somente o administrador pode gerenciar usuários e relatórios.
5. Os atributos do Chamado não podem ser nulos.
6. A priorização dos chamados será definida na abertura (operação da IA).
7. A priorização será definida pela pontuação em três campos do formulário, pela seguinte lógica:
 - a. Ocorre ocasionalmente ou de maneira contínua?
 - i. Ocasionalmente (+1 ponto)
 - ii. Continuadamente (+3 pontos)
 - b. Qual o impacto na demanda?
 - i. Impede a execução do trabalho (+3 pontos)
 - ii. Causa atraso, mas trabalho continua (+2 pontos)

iii. Impacto mínimo e sem prejuízos operacionais (+1 pontos)

c. Ocorre com todos os usuários ou apenas com você?

i. Todos os usuários (+3 pontos)

ii. Atinge apenas um grupo específico (+2 pontos)

iii. Apenas comigo (+1 ponto) Resultado da somatória de pontos:

- Alta prioridade: A partir de 7 pontos
- Média prioridade: 4 a 6 pontos
- Baixa prioridade: 3 pontos

8. Os usuários (Cliente ou Administrador) podem ter mais de um chamado em andamento simultaneamente.

9. Os usuários (Técnico ou Administrador) podem resolver mais de um chamado simultaneamente.

10. Todo Chamado novo possui uma solução automática inicial (resposta da IA).

11. Se a solução automática não atender, o Chamado será atribuído ao Técnico (operação da IA).

12. Toda resolução de chamado pode gerar uma nova solução automática (alimentação da IA).

13. É obrigatória a justificativa para chamados cancelados ou excluídos.

Com base nas regras de negócio e nos requisitos pré-estabelecidos, foram criadas as tabelas para cada entidade disposta no banco de dados físico, sendo estas: Tabela User, Tabela Cliente, Tabela Técnico, Tabela ADM, Tabela Chamado e Tabela Relatório.

As tabelas foram formalizadas e atendem as 3 (três) primeiras formas normais, ou seja, não há atributos multivalorados, sem dependência da Primary Key ou de dependência transitiva. Após a formalização, foi elaborado o dicionário de dados de cada tabela, contendo as informações do nome do atributo, tipo de dados, tamanho do dado em bytes, obrigatoriedade, tipo de chave e descrição.

A formalização e o dicionário de dados foram construídos para todas as tabelas, porém serão dispostas apenas algumas como exemplo, sendo as mais importantes, ilustradas nas Tabelas 5, 6 e 7.

Tabela 5 – Tabela User

Tabela User					
Campo	Tipo de Dado	Obrigatório	Tamanho	Chave	Descrição
id_User	INT (IDENTITY)	Sim	4 bytes	PK	Identificador único do usuário
email_User	VARCHAR	Sim	1 byte	UNIQUE	E-mail do usuário (não pode repetir)
senha_User	VARCHAR	Sim	1 byte	-	Senha do usuário
cargo_User	VARCHAR	Sim	1 byte	-	Cargo ocupado pelo usuário na empresa
departamento_user	VARCHAR	Sim	1 byte	-	Departamento ao qual o usuário pertence
nivelAcesso_User	INT	Sim	4 bytes	-	Nível de acesso do usuário no sistema
nome_User	VARCHAR	Sim	1 byte	-	Primeiro nome do usuário
sobrenome_User	VARCHAR	Sim	1 byte	-	Sobrenome do usuário

Fonte: Os autores, 2025.

Tabela 6- Tabela Relatório.

Tabela Relatório					
Campo	Tipo de Dado	Obrigatório	Tamanho	Chave	Descrição
id_Relatorio	INT (IDENTITY)	Sim	4 bytes	PK	Identificador único do relatório. Gerado automaticamente pelo banco.
id_Cham	INT	Sim	4 bytes	FK	Referência ao chamado relacionado (busca as informações do chamado)
relatorio	VARCHAR	Sim	1 byte	-	Texto do relatório referente ao(s) chamado(s) solicitado(s).
id_UsuarioCriador	INT	Sim	4 bytes	FK	Referencia ao Id do Administrador que gera o relatório.

Fonte: Os autores, 2025.

Tabela 7 – Tabela Chamado.

Tabela Chamado					
Campo	Tipo de Dado	Obrigatório	Tamanho	Chave	Descrição
id_Cham	INT (IDENTITY)	Sim	4 bytes	PK	Identificador do chamado. Gerado automaticamente pelo banco.
status_Cham	VARCHAR	Sim	1 byte	-	Status atual do chamado (aberto, em andamento, fechado)
dataAbertura_Ch am	DATE (DDMMAAAA)	Sim	3 bytes	-	Data de abertura do chamado no formato em dias, meses e anos
dataFechamento_ Cham	DATE (DDMMAAAA)	Não	3 bytes	-	Data de fechamento do chamado no formato em dias, meses e anos
dataProblema	DATE (DDMMAAAA)	Sim	3 bytes	-	Data de início do problema do chamado no formato em dias, meses e anos
prioridade_Cham	CHAR	Sim	1 byte	-	Prioridade do chamado (A - alta, M - média, B - baixa)
categoria_Cham	VARCHAR	Sim	1 byte	-	Categoria do chamado (ex: software, hardware)
solucaoIA_Cham	VARCHAR	Não	1 byte	-	Solução sugerida pela inteligência artificial (se ela houver)
solucaoTec_Cham	VARCHAR	Não	1 byte	-	Solução proposta pelo técnico
descricao_Cham	VARCHAR	Sim	1 byte	-	Descrição do problema informado
solucaoFinal_ Cham	VARCHAR	Não	1 byte	-	Solução final aplicada para resolver o chamado
tecResponsavel_ Cham	VARCHAR	Não	1 byte	FK	Nome do técnico responsável pelo chamado

Fonte: Os autores, 2025.

Além do dicionário de dados de cada tabela, foi feito o dicionário de dados para cada relacionamento: os relacionamentos entre as entidades e o relacionamento entre os relacionamentos, conforme apresentados nas Tabelas 8 e 9.

Tabela 8 – Relacionamento entre entidades.

Dicionário de Dados dos Relacionamentos			
Tabela	Relacionamento	Nome do Relacionamento	Descrição
tbl_Cliente	tbl_Chamado	Abrir	Tabela para abertura de um chamado feito pelo cliente.
tbl_Tecnico	tbl_Chamado	Solucionar	Tabela para solução do técnico de um chamado
tbl_ADM	tbl_Relatorio	Gerar	Tabela para gerar um relatório feito pelo ADM
	tbl_Chamado	Gerenciar	Tabela para gerenciamento de um relatório feito pelo ADM
tbl_Chamado	tbl_Tecnico	Solucionar	Tabela para solução do técnico de um chamado
	tbl_Cliente	Abrir	Tabela para abertura de um chamado feito pelo cliente.
	tbl_ADM	Gerar	Tabela para gerar um relatório feito pelo ADM
		Gerenciar	Tabela para gerenciamento de um relatório feito pelo ADM
tbl_Relatorio	tbl_ADM	Gerar	Tabela para gerenciamento de um relatório feito por um ADM
tbl_Categoria	tbl_Chamado	Possuir	Tabela para adicionar categorias a um chamado.

Fonte: Os autores, 2025.

Tabela 9 – Relacionamento entre relacionamentos.

Dicionário de Dados dos Relacionamentos			
Relacionamento	Tabela 1 - FK	Tabela 2 - PK	Descrição
Abrir	Chamado	Cliente	Relacionamento que descreve qual cliente abre cada chamado
Solucionar	Chamado	Tecnico	Relacionamento que descreve qual tecnico soluciona cada chamado
Gerenciar	Chamado	ADM	Relacionamento que descreve qual ADM gerencia cada chamado
Gerar	Relatorio	ADM	Relacionamento que descreve qual ADM gera cada relatorio

Fonte: Os autores, 2025.

Após o processo de criação de tabelas, foram desenvolvidos os três tipos de diagrama de entidade-relacionamento: o conceitual descritivo, o conceitual e o lógico. O conceitual descritivo foi feito para cada tabela e cada relacionamento, sendo este:

Entidade: User. - Atributos: id_User (Primary Key), nome_User, sobrenome_User, email_User, senha_User, cargo_User, departamento_User, nívelAcesso_User, nome_User

Entidade Cliente - Atributos: id_User (Primary Key e Foreign Key de User). Entidade Técnico - Atributos: id_User (Primary Key e Foreign Key de User). Entidade ADM - Atributos: id_User (Primary Key e Foreign Key de User).

Entidade Chamado - Atributos: id_Cham (Primary Key), dataProblema, tecResponsavel_Cham (Foreign Key), solucaoFinal_Cham, descricao_Cham, solucaoTec_Cham, categoria_Cham, prioridade_Cham, dataFechamento_Cham, dataAbertura_Cham, status_Cham.

Entidade Relatório - Atributos: id_Relatorio (Primary Key), id_Cham (Foreign Key), relatorio, id_Usuariocriador.

Relacionamentos:

Abrir (entre Cliente e Chamado):

Descrição: Um cliente pode abrir nenhum ou vários chamados;

Um chamado é aberto por um único cliente; Cardinalidade: 1:N (Cliente - Chamado).

Solucionar (entre Técnico e Chamado)

Descrição: Um técnico pode solucionar nenhum ou vários chamados;

Um chamado é solucionado por um único técnico; Cardinalidade: 1:N (Técnico - Chamado)

Gerar (entre ADM e Relatório)

Descrição: Um ADM pode gerar nenhum ou mais relatórios;

Um relatório é gerado por um ADM; Cardinalidade: 1:N (ADM - Relatório).

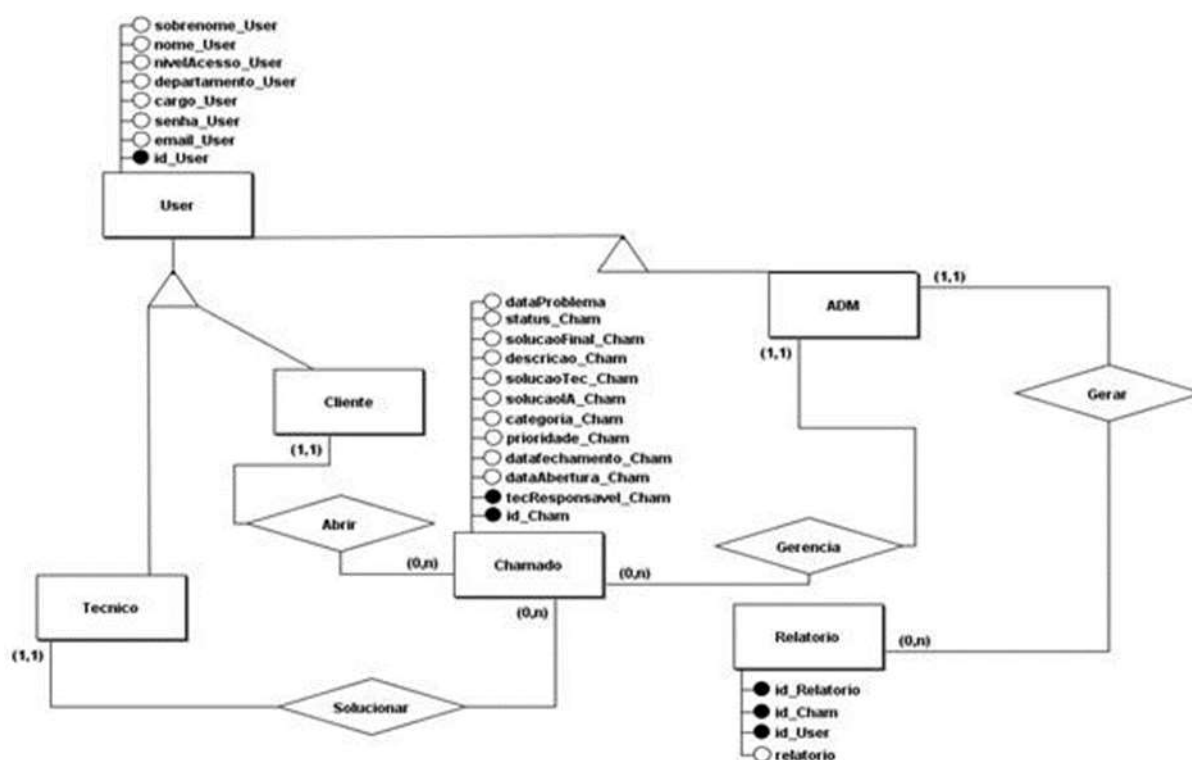
Gerencia (entre ADM e Chamado)

Descrição: Um ADM pode gerenciar nenhum ou mais chamados;

Cada ADM pode gerenciar um chamado; Cardinalidade: 1:1 (ADM - Chamado).

O DER conceitual visual deve seguir as informações à risca do conceitual descritivo, e após a elaboração dele no aplicativo brModelo, obteve-se o diagrama da Figura 3.

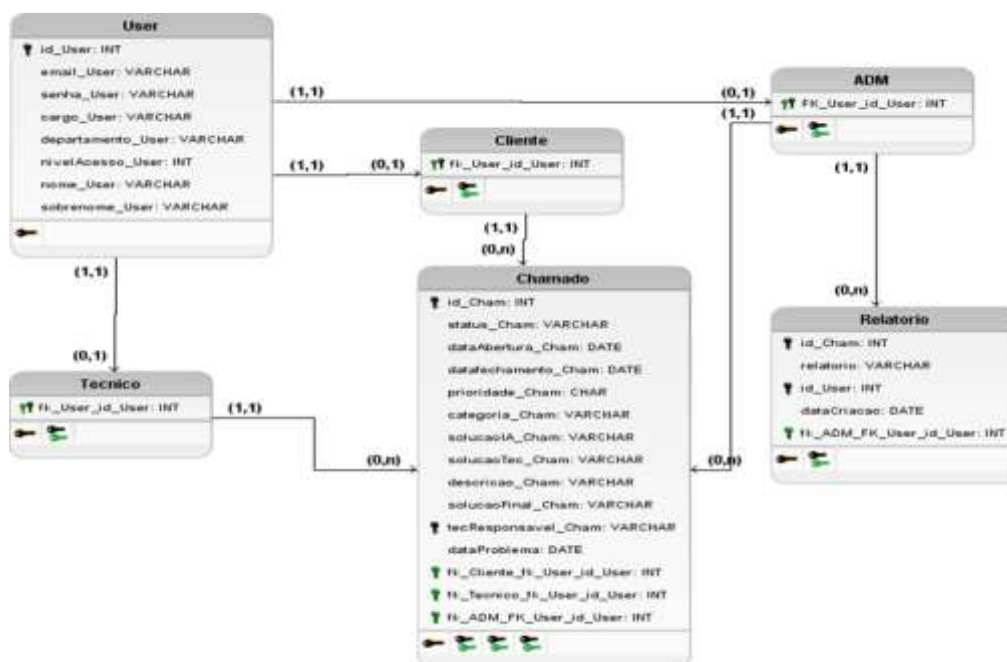
Figura 3 – Diagrama de Entidade-Relacionamento Conceitual Visual.



Fonte: Os autores, 2025.

Com o DER Conceitual em mãos, foi gerado o modelo lógico, facilitando uma posterior criação do banco de dados físico, utilizando o SQL Server, conforme ilustra a Figura 4.

Figura 4 –Diagrama de Entidade-Relacionamento Lógico.



Fonte: Os autores, 2025.

Todas as etapas anteriores são primordiais para a criação de um código em linguagem SQL, visto que realizando esses processos, haverá uma padronização e formalização do banco de dados, evitando que haja erros no processo de criação ou de execução. Quando um defeito é identificado de forma precoce em uma das etapas anteriores, é simples ajustar o que estava equivocado e seguir o processo de uma forma linear.

Por fim, foi desenvolvida a parte física do banco de dados, a saber: o código em linguagem SQL Server. O principal objetivo do script é criar as tabelas necessárias para o funcionamento do sistema, definindo suas colunas, tipos de dados e relacionamentos, garantindo uma estrutura eficiente e organizada para o armazenamento das informações.

Inicialmente, é criada a tabela `Usuario` contendo os campos `id`, `email`, `senha`, `cargo`, `departamento`, `nível de acesso`, `nome` e `sobrenome`. O campo `id` atua como chave primária e como ponte entre a tabela `Usuario` e as demais tabelas especializadas, além disso, duas triggers são definidas:

- `trg_InserirUsuario`: insere automaticamente os dados do usuário na tabela correspondente ao seu tipo (`Cliente`, `Técnico` ou `Administrador`) após um `INSERT` na tabela `Usuario`.
- `trg_Usuario_PadronizarCargo`: padroniza automaticamente os textos dos campos de cargos, garantindo consistência na entrada de dados.

Em seguida, são criadas as tabelas `Cliente`, `Tecnico` e `Administrador`, cada uma contendo apenas o campo `id_user`, que é chave primária da respectiva tabela e chave estrangeira referenciando o `id` da tabela `Usuario`. Essa separação por tipo de usuário permite, futuramente, o armazenamento de dados específicos e exclusivos para cada categoria de usuário.

Posteriormente, é criada a tabela `Chamado`, com os campos `status`, `data_abertura`, `data_fechamento`, `data_problema`, `prioridade`, `categoria`, `descricao`, `solucao_ia`, `solucao_tecnico`, `solucao_final` e `tecnico_responsavel`, sendo este último uma chave estrangeira que referência o campo `id_user` da tabela `Tecnico`. A tabela `Chamado` conta com restrições (constraints) para garantir a integridade dos dados:

- `CK_Status_Chamado`: valida que apenas valores de status permitidos sejam inseridos.
- `CK_Prioridade_Chamado`: valida os valores de prioridade permitidos.

Além disso, foi implementada a trigger `trg_NormalizaStatusChamado`, que padroniza automaticamente os textos inseridos nos campos `status` e `prioridade`. Por exemplo, se o usuário inserir "EM anDAMENTO", o sistema normaliza e armazena como "Em andamento". Por fim, é criada a tabela

Relatorio, composta pelos campos `id_relatorio`, `id_chamado`, `relatorio`, `data_criacao` e `id_usuario_criador`. Desses, dois são chaves estrangeiras:

- `id_chamado` referencia o campo `id_chamado` da tabela Chamado;
- `id_usuario_criador` referencia o campo `id_user` da tabela Usuario.

Para garantir que apenas usuários do tipo Administrador possam criar relatórios, foi definida a trigger `trg_ValidarRelatorioADM`. Essa trigger verifica se o valor de `id_usuario_criador` corresponde a um usuário da tabela Usuario que esteja registrado como Administrador, bloqueando a inserção caso contrário. O código SQL está disponível no link: [Banco de Dados: Scripts](#).

No modelo conceitual visual, o atributo da tabela relatório era `id_User` porém, foi criado outro nome (`id_UsuarioCriador`) referenciando a PK `id_User`, melhorando a manutenibilidade do script. A preocupação com a integridade dos dados e a separação por papéis de usuários reforça o compromisso com a escalabilidade e a clareza do sistema. Com essa estrutura sólida, o sistema está preparado para sustentar as funcionalidades previstas e permitir futuras evoluções com consistência e eficiência.

9. CONSIDERAÇÕES FINAIS

Este estudo surgiu como uma resposta para a otimização de sistemas de gerenciamento de chamados e suporte técnico com base em Inteligência Artificial. Durante a construção deste, ficou evidente a necessidade e utilidade de todos os conceitos da orientação a objetos para que a análise e modelagem do sistema fosse concisa e completa para um desenvolvimento eficiente da codificação do software no futuro.

O levantamento de requisitos, a modelagem UML (caso de uso, classe, sequência e implantação), a construção do modelo conceitual, físico e lógico do banco de dados, em conjunto com o dicionário de dados fez com que fosse possível uma modelagem consistente do sistema, demonstrando habilidade de compreensão perante as necessidades e expectativas do cliente, que auxiliarão na rastreabilidade e manutenibilidade futura. As planilhas de testes de software e os casos de uso descritivos facilitaram a busca pela qualidade do software.

O protótipo gerou a possibilidade de validar as ideias, identificar possíveis problemas de design e de funcionalidade, além da eficiência das mesmas, prezando pela usabilidade. Ademais, o estudo de viabilidade econômica e a delimitação do modelo de contratação gerou uma perspectiva de espaço de crescimento de mercado e construção de uma equipe produtiva em um cenário competitivo.

Posteriormente, sugere-se que algumas alterações e adições sejam implementadas, desenvolvendo features de notificações para atualizações sobre os chamados que o usuário abriu ou para avisar um técnico quando ele for atribuído a um novo chamado, adicionar status do usuário (ativo ou inativo) e a data em que o usuário foi criado, além de analisar a viabilidade de incluir um chat entre clientes e técnicos para centralizar totalmente a comunicação.

É pertinente buscar formas de incluir acessibilidade (aumento de fonte, mudança de contraste, integração com a API vLibras), bem como implementar medidas de segurança para garantir robustez no cumprimento da Lei Geral de Proteção de Dados Pessoais (LGPD) de forma robusta, incluindo também uma nova confirmação no sistema do Administrador para a inclusão de senha em ações críticas, visando segurança.

REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON, O que é o Scrum?, Disponível em: <https://aws.amazon.com/pt/what-is/scrum/>. Acesso em: 08 de maio de 2025.

BARBOSA, Eduardo Pinto; ALVARENGA, Geoflávia Guilarducci de. Uma proposta para ensino de dicionário de dados em projetos de bancos de dados. Anais do Workshop sobre Educação em Computação (WEI), 20., 2012, Curitiba/PR. Porto Alegre: Sociedade Brasileira de Computação, 2012. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/29098/28903>. Acesso em: 10 maio 2025.

BONATINI, B. S. Quais os benefícios da Programação Orientada a Objetos no desenvolvimento de softwares, 2023. Disponível em:

<<https://medium.com/@brunobonatini/quais-os-benef%C3%ADcios-da-programa%C3%A7%C3%A3o-orientada-a-objetos-no-desenvolvimento-de-softwares-c836f7ce275>>. Acesso em: 03 maio. 2025.

BOOCH, G., JACOBSON, I., RUMBAUGH, J. (2005). UML: guia do usuário. 2. ed.

Rio de Janeiro: Campus.

BORSOI, Beatriz T.; COLAZZOS, Kathya S. L.; ASCARI, Rúbia E. O. S.; TOSCAN,

Luiz F.; BOSSOLA, Luiz H.; BOLO, Matheus M.; ARSEGO, Matheus M. Redes neurais aplicadas na estimativa de prazo de projeto de software. IV EPAC – Encontro Paranaense de Computação. Pato Branco: Universidade Tecnológica Federal do Paraná, 2015. Disponível em: <https://cliqueapostilas.com/Content/apostilas/b1e03f65cc0ee3bbb3bcd0d8bd85b514.pdf>. Acesso em: 13 abr. 2025.

CERQUEIRA, Alex. Qualidade em Usabilidade, 2024. Disponível em:

<<https://medium.com/editoraux/qualidade-em-usabilidade-48006cbb3a33>> Acesso em: 10 de abril de 2025.

COOPERSYSTEM, Análise de requisitos: o que é e para que serve?, 2023 Disponível em: <https://www.coopersystem.com.br/analise-de-requisitos-o-que-e-e-para-que-serve/>. Acesso em: 07 de maio de 2025.

Curso de Automação Industrial [Online]. Universidade Federal do Rio Grande do Norte. Disponível em: <<https://materialpublic.imd.ufrn.br/curso/disciplina/1/8/1/3>>. Acesso em: 22 maio 2025.

DEVMEDIA, Qualidade de Software - Engenharia de Software, Disponível em: <https://www.devmedia.com.br/amp/qualidade-de-software-engenharia-de-software-29/18209>. Acesso em: 08 de maio de 2025.

ESL SISTEMAS. Investimento em tecnologia faz empresas crescerem 60% a mais no mercado, 2024. Disponível em: <https://eslsistemas.com.br/investimento-em-tecnologia-faz-empresas-crescerem-60-a-mais-no-mercado/>. Acesso em: 27 de março de 2025.

GAVA, Marcela. Pesquisa sobre atendimento ao cliente mostra SAC melhor agora do que nos últimos dois anos. Capterra, 28 de junho de 2022. Disponível em: <https://www.capterra.com.br/blog/2763/pesquisa-atendimento-cliente> . Acesso em: 28 de março de 2025.

ISO - International Organization for Standardization. Norma ISO/IEC 25010:2011. Disponível em: <<https://www.iso.org/standard/35733.html>>. Acesso em: 18 de julho de 2025.

LECOM, Gestão estratégica de RH: o que é, importância e ferramentas, 2023. Disponível em: <<https://www.lecom.com.br/blog/gestao-estrategica-rh/>>. Acesso em: 04 de maio de 2025.

NIELSEN, J. Enhancing the explanatory power of usability heuristics. Proceedings of the 9th Conference on Human Factors in Computing Systems, New York, New York, EUA, 1994.

NORMANDO, Célio. Orientação a objetos, 2024. Disponível em:

<<https://medium.com/@celionormando/orienta%C3%A7%C3%A3o-a-objetos-1cc5cd11abfa>> Acesso em: 10 de maio de 2025.

ORACLE. What is a database? Disponível em: <https://www.oracle.com/br/database/what-is-database>. Acesso em: 16 abr. 2025.

PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo Scrum para Gerenciar Projetos de Forma Ágil. 2007. Disponível em: https://faculdadeprojecao.nucleoad.net/pos/pluginfile.php/2123/mod_resource/content/36/Entendendo_Scrum_para_Gerenciar_Projetos.pdf. Acesso em: 10 maio 2025.

PUIG, Luis Ezequiel. Normalização em Banco de Dados - Estrutura. Alura, 7 mar. 2023. Disponível em: <https://www.alura.com.br/artigos/normalizacao-banco-de-dados-estrutura>. Acesso em: 4 maio 2025.

SEBRAE. Atendimento ao cliente: o que é e como fidelizar o consumidor, 2024. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/atendimento-ao-cliente-o-que-e-como-fazer-para-fidelizar-o-consumidor%2Ce4b0af9d324bb810VgnVCM1000001b00320aRCRD> . Acesso em: 28 de março de 2025.

SEBRAE. ROI calcula o retorno de seus investimentos. Sebrae, 6 maio 2023. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/roi-calcula-o-retorno-de-seus-investimentos,17870ffe48437810VgnVCM1000001b00320aRCRD>. Acesso em: 10 maio 2025.

SERASA EXPERIAN, RH: Qual a função desse departamento?, 2023. Disponível em: <<https://www.serasaexperian.com.br/carreiras/blog-carreiras/rh-departamentofuncionarios-e-importancia-para-a-empresa/>>. Acesso em: 21 de abril de 2025.

SILVA, Cleuton. Os 4 Pilares da Programação Orientada a Objetos, 2023. Disponível em: <<https://www.dio.me/articles/os-4-pilares-da-programacao-orientada-a-objetos-SSU4Q9>> Acesso em: 02 de maio de 2025.

SILVA, Douglas da. Dados sobre o atendimento ao cliente. ZENDESK, 06 de março de 2025. Disponível em: <https://www.zendesk.com.br/blog/dados-estatisticas-atendimento-cliente/> . Acesso em: 28 de março de 2025.

SILVA, José Carlos da; ALMEIDA, Maria de Fátima Costa; SOUZA, Ana Paula de Lima. Ratio-Entity Diagram: uma ferramenta para modelagem conceitual de dados em Engenharia de Software. Research, Society and Development, [S. l.], v. 10, n. 7, p. e44210717776, 2021. DOI: 10.33448/rsd-v10i7.17776. Disponível em:

<https://rsdjournal.org/index.php/rsd/article/view/17776/15626>. Acesso em: 16 abr. 2025.

Sommerville, I. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

SOMMERVILLE, Ian. Engenharia de Software. 10ª ed. São Paulo: Pearson Universidades, 2019.

SOMMERVILLE, Ian. Engenharia de Software. 10ª ed. São Paulo: Pearson Universidades, 2019.

STRIKER, Cole; KAVLAKOGLU, Eda. O que é inteligência artificial?. IBM, 09 de agosto de 2024. Disponível em: <https://www.ibm.com/br-pt/think/topics/artificial-intelligence> . Acesso em: 28 de março de 2025.

TEIXEIRA, Fabricio. Análise heurística: o que é, como fazer e os benefícios para o projeto, 2016. Disponível em: <<https://brasil.uxdesign.cc/an%C3%A1lise-heur%C3%ADstica-o-que-%C3%A9-como-fazer-e-os-benef%C3%ADcios-para-o-projeto-161f3d94436b>> Acesso em: 12 de abril de 2025.

TEIXEIRA, Rafael Fialho. Sistema de chamados: o que é, como funciona e como implementar. Desk Manager, 25 de outubro de 2024. Disponível em: <https://deskmanager.com.br/blog/sistema-de-chamados/>. Acesso em: 27 de março de 2025.

TWYGO, RH Estratégico, o que é, qual a importância e como implantar, 2022. Disponível em: <<https://twygo.com/blog/rh-estrategico/>>. Acesso em: 04 de maio de 2025.

VALENTE, Marco T. Engenharia de Software Moderna: Princípios e Práticas para o Desenvolvimento de Software e com Produtividade. Editora: Independente, 2020. Disponível em <<https://engsoftmoderna.info/>>. Acesso em: 18 de jul. de 2025.

WAZLAWICK, Raul S. Análise e Design Orientados a Objetos para Sistemas de Informação: Modelagem com UML, OCL e IFML. 3. ed. Rio de Janeiro: GEN LTC, 2014. Disponível em:

<<https://app.minhabiblioteca.com.br/reader/books/9788595153653/>>. Acesso em: 21 mai. 2025.

Capítulo 2



10.37423/250810239

DESCOBRINDO A MATEMÁTICA: ATRAVÉS DE OLHARES ESPECIAIS

Fernanda Aparecida Pereira

IFMT

Ana Maria Rodrigues Pinto

Creche Thayná Gabrielly Oliveira Moraes

Eliane Aparecida Rocha da Silva

Creche Thayná Gabrielly Oliveira Moraes

Elisângela Gouvêa de Souza

Creche Thayná Gabrielly Oliveira Moraes

Maria Iraci Malaquias

Creche Thayná Gabrielly Oliveira Moraes

Maria Ferreira da Silva

Creche Thayná Gabrielly Oliveira Moraes

Patricia Aparecida Gonçalves

Creche Thayná Gabrielly Oliveira Moraes

Patricia Vicente de Oliveira

IFMT

Silvane Alves de Souza

Creche Thayná Gabrielly Oliveira Moraes

Tamiris Silvério Braga

IFMT

CAPÍTULO DESCOBRINDO A MATEMÁTICA: ATRAVÉS DE OLHARES ESPECIAIS

A alfabetização, tradicionalmente associada ao domínio da leitura e da escrita, precisa ser compreendida de maneira mais ampla, integrando também a linguagem matemática como elemento fundamental na formação do sujeito. A aprendizagem dos conceitos matemáticos nos anos iniciais do ensino fundamental representa um dos pilares do desenvolvimento cognitivo, lógico e social dos alunos. No entanto, muitos estudantes enfrentam dificuldades significativas nesse processo, especialmente quando suas necessidades específicas de aprendizagem não são contempladas pelas metodologias tradicionais.

Neste contexto, o projeto "**Descobrimdo a Matemática: Através de Olhares Especiais**" surge como uma proposta inovadora e sensível, que alia inclusão, ludicidade e interdisciplinaridade na promoção de uma alfabetização matemática mais significativa. O projeto foi pensado para atender à diversidade presente nas salas de aula contemporâneas, respeitando diferentes ritmos e estilos de aprendizagem e promovendo o engajamento de todos os alunos, com ou sem deficiência, em um ambiente de aprendizagem colaborativo e acolhedor.

Ao valorizar os múltiplos olhares e formas de compreender o mundo, esta iniciativa reconhece que a matemática não deve ser ensinada como um conjunto rígido de regras abstratas, mas como uma linguagem viva e funcional, que dialoga com o cotidiano e com as experiências culturais dos sujeitos. Dessa forma, o capítulo que segue apresenta os fundamentos, objetivos, métodos e resultados deste projeto, oferecendo uma reflexão crítica sobre as possibilidades de um ensino de matemática que seja, ao mesmo tempo, inclusivo, contextualizado e transformador.

OBJETIVOS DO PROJETO

O projeto foi estruturado com base em quatro grandes objetivos, que norteiam tanto as práticas pedagógicas quanto as escolhas metodológicas adotadas:

1. **Facilitar a Alfabetização Matemática:**

Proporcionar uma base sólida nos conceitos matemáticos fundamentais, utilizando recursos didáticos adaptados às necessidades e capacidades individuais dos alunos.

2. **Promover a Inclusão:**

Desenvolver estratégias pedagógicas que respeitem as diferenças individuais, assegurando o direito de todos à aprendizagem da matemática, especialmente os estudantes com deficiência ou dificuldades de aprendizagem.

3. Estimular o Pensamento Crítico e Lógico:

Incentivar o desenvolvimento de habilidades como o raciocínio lógico, a resolução de problemas e a tomada de decisões fundamentadas, competências essenciais no percurso acadêmico e na vida cotidiana.

4. Integrar Disciplinas:

Estabelecer conexões significativas entre a matemática e outras áreas do conhecimento, particularmente a língua portuguesa, promovendo uma abordagem interdisciplinar e mais coerente com as práticas da vida real.

METODOLOGIAS ATIVAS E INCLUSIVAS

Compreendendo que a aprendizagem acontece de maneira mais eficaz quando o estudante se envolve ativamente no processo, o projeto adotou uma abordagem centrada no aluno. A seguir, apresentamos as principais estratégias pedagógicas utilizadas:

- **Jogos Educativos:**

Os jogos são ferramentas potentes de mediação da aprendizagem. No projeto, foram utilizados jogos que envolvem contagem, comparação, lógica e operações básicas, promovendo a fixação de conteúdos por meio de experiências lúdicas e desafiadoras.

- **Tecnologias Educacionais:**

Ferramentas digitais, como aplicativos, vídeos interativos e softwares matemáticos, foram integradas às aulas para facilitar a visualização de conceitos abstratos e ampliar as possibilidades de aprendizagem, especialmente para alunos com deficiências sensoriais ou cognitivas.

- **Trabalho em Grupo:**

A cooperação foi incentivada como valor pedagógico e social. Em atividades em grupo, os alunos compartilham saberes, constroem juntos soluções e aprendem a respeitar diferentes pontos de vista, fortalecendo a empatia e a inclusão.

- **Projetos Investigativos:**

Através de projetos temáticos, os estudantes foram convidados a investigar situações do cotidiano em que a matemática se faz presente como medir ingredientes, calcular distâncias ou estimar valores, conectando teoria e prática de forma concreta e contextualizada.

RESULTADOS ALCANÇADOS

Os efeitos observados ao longo da implementação do projeto foram bastante positivos. Entre os principais resultados, destacam-se:

MELHORIA NA COMPREENSÃO MATEMÁTICA:

Os alunos demonstraram avanços significativos na assimilação de conceitos matemáticos, especialmente aqueles que antes apresentavam dificuldades de aprendizagem.

AUMENTO DA MOTIVAÇÃO E DA AUTOESTIMA:

Ao participar de atividades lúdicas e inclusivas, os estudantes passaram a se envolver com mais entusiasmo nas aulas de matemática, fortalecendo sua autoconfiança.

AMBIENTE DE APRENDIZAGEM MAIS COLABORATIVO:

A promoção do trabalho em grupo e o respeito às diferenças contribuíram para a construção de uma cultura escolar mais empática e solidária.

INTEGRAÇÃO INTERDISCIPLINAR EFETIVA:

As conexões entre matemática e linguagem escrita tornaram-se mais evidentes, possibilitando uma aprendizagem mais ampla e articulada.

CONSIDERAÇÕES FINAIS

O projeto **"Descobrimos a Matemática: Através de Olhares Especiais"** evidencia o potencial transformador de uma pedagogia inclusiva, ativa e interdisciplinar. Ao reconhecer a diversidade dos estudantes e oferecer estratégias pedagógicas sensíveis a essa realidade, é possível não apenas ensinar matemática, mas formar sujeitos mais confiantes, autônomos e participativos.

A experiência mostra que ensinar matemática com afeto, criatividade e respeito às particularidades humanas é uma ação profundamente política e necessária. Mais do que ensinar números, é ensinar a

pensar, a questionar e a transformar o mundo — objetivos centrais de uma educação democrática e de qualidade.

O modelo apresentado neste capítulo pode ser adaptado a diferentes contextos escolares, desde que respeite as especificidades locais e mantenha como princípios norteadores a inclusão, a ludicidade e o protagonismo dos alunos no processo de aprendizagem.

REFERÊNCIAS

D'Ambrósio, U. Etnomatemática: Um Programa. Autêntica Editora, 2002.

Fazenda, I. C. A. Interdisciplinaridade: História, Teoria e Pesquisa. Papirus, 2011.

Freire, P. Pedagogia da Autonomia: Saberes Necessários à Prática Educativa. Paz e Terra, 1996.

Kilpatrick, J., Swafford, J., & Findell, B. (Orgs.). Adding It Up: Helping Children Learn Mathematics. National Academy Press, 2001.

Mantoan, M. T. E. Inclusão Escolar: O Que é? Por Quê? Como Fazer? Moderna, 2003.

Skovsmose, O. "Towards a Critical Mathematics Education." Educational Studies in Mathematics, 1994.

Smole, K. S., & Diniz, M. I. A Matemática na Educação Infantil: A Teoria e a Prática de Brincar e Aprender. Artmed, 2001.

Valente, J. A. Tecnologia no Ensino: Novos Tempos, Outras Estratégias. Cortez, 1993.

Capítulo 3



10.37423/250810249

NÚMEROS PERFEITOS E PRIMOS DE MERSENNE

Carollyne da Cruz Almeida

Universidade Federal de Mato Grosso

Thais Silva do Nascimento

Universidade Federal de Mato Grosso

Elayne Xavier Souza Araujo

Universidade Federal de Mato Grosso



Resumo: O presente trabalho tem por objetivo expor a relação entre os números perfeitos e os números primos de Mersenne, além de apresentar algumas propriedades interessantes dos números perfeitos quando convertidos para a forma binária. Para demonstrar essa relação, utilizaremos o Teorema de Euclides-Euler, desenvolvida por ambos quando se encontravam inteiramente instigados em desvendar as propriedades matemáticas que esses números apresentavam. Ao longo desse documento, também apresentaremos um pouco da história do surgimento desses números, apresentando os principais matemáticos que contribuíram para esse avanço. Abordamos também, de forma sucinta, a aplicabilidade dos números primos em áreas de grande relevância para a sociedade.

Palavras-chave: Números perfeitos. Primos de Mersenne. Teorema de Euclides- Euler.

1 INTRODUÇÃO

Poucos já ouviram falar sobre a existência dos números perfeitos. O que muitos não sabem é que essa categoria numérica carrega consigo um conjunto esotérico, repleto de ligações místicas e especulações numerológicas. Por muito tempo, os números perfeitos chamaram a atenção dos matemáticos, que, no decorrer da história da humanidade, foram estudando novas propriedades e formulando novas conjecturas, na qual um dos principais resultados garante que existe uma relação entre os números perfeitos e os números primos de Mersenne.

Os números perfeitos fazem parte da aritmética Pitagórica. Desde sua criação, os pitagóricos sempre foram extremamente fascinados pelos números, e acreditavam que eles estavam ligados a origem do universo. Os pitagóricos costumavam classificar os números naturais em diversas categorias, tais como: números perfeitos, números abundantes, números deficientes, números amigáveis. Para eles, os números são entidades reais com natureza própria, se tratava especialmente do princípio de todas as coisas.

Segundo Pitágoras, o número estava presente antes mesmo do homem perceber a sua existência. De acordo com ele, o homem criou a Matemática mas os números e a medida sempre estiveram ali.

Nesse trabalho, abordaremos essa temática, onde inicialmente, exploraremos a trajetória da existência desses números ao longo da história, destacando os principais matemáticos que contribuíram para sua descoberta. Em seguida, discutiremos alguns conceitos fundamentais da matemática, como o Teorema Fundamental da Aritmética e o Crivo de Eratóstenes. Posteriormente, conceituaremos os números perfeitos e os primos de Mersenne, finalizando com a demonstração do Teorema de Euclides-Euler, que estabelece a relação entre esses dois tipos de números. Para finalizar, apresentaremos a magnífica propriedade dos números perfeitos ao serem convertidos em números binários e sua importância para a sociedade.

2 EM BUSCA DA PERFEIÇÃO DOS NÚMEROS

Aqui nessa seção vamos destacar todo o desdobramento histórico dos números perfeitos, enfatizando o seu aspecto matemático e cultural, descrevendo a sua evolução teórica. Como já mencionado anteriormente, os números perfeitos ainda continuam sendo um fascinante objeto de estudo matemático, especialmente na Teoria dos Números, devido as suas propriedades únicas que foram desenvolvidas e aprimoradas por diversas gerações de matemáticos ao longo do tempo. Além disso,

os números perfeitos carregam consigo um simbolismo cabalístico, sendo associados ao conceito de harmonia e perfeição por diferentes grupos culturais.

ARITMÉTICA PITAGÓRICA

A história dos números perfeitos se inicia na Grécia antiga com a fundação da Escola Pitagórica, quando os membros dessa seita filosófica decidiram atribuir à alguns números, que carregavam consigo uma aura mística, o caráter filosófico e mágico. Esses números especiais foram associados à ordem cósmica e ao equilíbrio, despertando em alguns matemáticos da antiguidade um profundo interesse em buscar padrões e estruturas que representassem a propriedade harmônica desses números.

Na obra literária intitulada de Introdução a história da matemática, de Howard Eves, traduzido pelo professor brasileiro Hygino Domingues, é mencionado o surgimento da Aritmética Pitagórica como um marco fundamental no estudo das relações abstratas envolvendo os números, pois os estudos realizados pelos integrantes dessa escola contribuíram significativamente para o desenvolvimento da aritmética, e conseqüentemente, para o avanço considerável da Teoria dos Números que atualmente continua inspirando novas pesquisas. Além disso, o livro ainda destaca o hábito dos gregos antigos realizarem a distinção entre o estudo teórico dos números, denominado de aritmética, e o cálculo prático desses números, esse chamado de logística, na qual hoje em dia, sabemos que a aritmética corresponde a logística dos gregos antigos.

Dessa forma, com o intuito de aprofundar e investigar as relações abstratas e a simbologia mágica que os números carregavam, os pitagóricos começaram a classificar esses números inteiros em números amigáveis, números perfeitos, números abundantes e números deficientes.

É impressionante como essas categorias numéricas apresentam propriedades fascinantes capazes de despertar o interesse eminente em qualquer pessoa, seja ela um teórico matemático ou até mesmo uma pessoa leiga que não compreende nada sobre o assunto. Como ainda salienta Eves (1911-2004), dois números se dizem amigáveis se cada um deles é a soma dos divisores próprios do outro, como é o caso dos números 220 e 284. Pois, os divisores próprios de 220 é 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110, que ao somar esses números obtemos o resultado igual a 284, enquanto que, somando os divisores do número 284, que são: 1, 2, 4, 71 e 142 obtemos o número 220.

Eves ainda ressalva que esses pares numéricos possuem uma relação especial e representam o símbolo da amizade, pois um talismã gravado com esses dois números eram comumente vendidos na Idade Média, e se usado por duas pessoas poderia selar o pacto de amizade ou um amor eterno.

Analogamente, os números perfeitos também apresentam interpretações místicas e esotéricas, demonstrando a sua proximidade com as crenças populares. Algumas civilizações observaram que a Lua orbita a Terra a cada 28 dias e, também acreditavam que Deus havia criado a Terra em 6 dias, reforçando a ideia de que esses números são perfeitos.

Segundo Singh (2006), Pitágoras acreditava que a perfeição estava ligado a quantidade de divisores de um número. Ele era extremamente fascinado pelos números perfeitos e os atribuía várias propriedades elegantes, como, por exemplo, a soma de números perfeitos são sempre a soma de uma série de números inteiros. Aqui consideramos divisores próprios, os divisores positivos do número natural diferente dele mesmo.

Outra propriedade, descoberta por Pitágoras, embora tenha falhado, foi que a perfeição estava ligada ao número 2. Alguns números são conhecidos como potência de 2 e podem ser escritos na forma 2^n , em que $n \in \mathbb{N}$, indica a quantidade de vezes que ele é multiplicado por ele mesmo. No entanto, ao somar os divisores próprios dessa potência o resultado é sempre uma unidade menor, revelando assim, a natureza imperfeita das potências.

Abordando a propriedade de divisibilidade dos números temos que os divisores próprios de 6, que são 1, 2 e 3, cuja soma desses divisores resulta no próprio número, logo ele é perfeito. Caso contrário, quando a soma dos divisores próprios de um número é maior do que ele próprio, é chamado de abundante, como por exemplo, o número 12, pois seus divisores próprios são 1, 2, 3, 4 e 6. Logo o número 12 é um número abundante, pois a soma de seus divisores próprios é igual a 16.

De maneira análoga, ocorria os números deficientes. Um número é dito deficiente se a soma de seus divisores próprios é menor que ele próprio. É o caso do número 10, cujos divisores próprios são 1, 2 e 5, logo a soma dos divisores próprios de 10 é igual a 8.

EUCLIDES

Embora os Pitagóricos já tivessem interesse em padrões numéricos, foi Euclides que formulou a primeira demonstração matemática sobre os números perfeitos, pois a maior parte dos escritos que relatam a existência desses números se encontra no Livro IX da obra Os elementos de Euclides. Este

documento é considerado um tratado por conter sistematicamente quase todo o conhecimento matemático da época.

Conforme argumenta Cruz (2013), apesar de os Elementos de Euclides ser amplamente conhecido como um tratado de Geometria, sua estrutura não se limita apenas a essa área. Dos treze livros que compõem a obra, apenas dez versam sobre Geometria e os outros três deles

- VIII, VII, IX - focam em conceitos relacionados a Aritmética, nesses volumes, é abordado teorias de divisibilidade sobre números primos, números perfeitos, máximo divisor comum e mínimo divisor comum. É exatamente nessa parte que Euclides também desenvolve o Teorema Fundamental da Aritmética e condições necessárias para que um número natural fosse um número perfeito.

NICOMÂCO DE GERASA

Outro matemático que se envolveu profundamente nos estudos dos números perfeitos foi o filósofo grego Nicômaco de Gerasa (c.60 - c.120). Conforme a University of St Andrews (s.d.), Nicômaco não apenas classificava os números em superabundantes, deficientes e perfeitos, mas também os atribuía alguns aspectos morais e biológicos.

Segundo a University of St Andrews(s.d.):

No caso do excesso, produzem-se excesso, superfluidade, exageros e abusos; no caso da falta, produzem-se carências, omissões, privações e insuficiências. E no caso daqueles que se encontram entre o excesso e a falta, isto é, em igualdade, produzem-se virtude, medida justa, propriedade, beleza e coisas desse tipo — das quais a forma mais exemplar é aquele tipo de número que se chama perfeito.(UNIVERSITY OF ST ANDREWS, [s.d.])

Em sua obra intitulada “*Introdução a aritmética*”, conhecendo os quatro primeiros números perfeitos 6, 28, 496 e 8128, Nicômaco de Gerasa formulou cinco conjecturas sobre os números perfeitos sem apresentar qualquer demonstração. Elencamos abaixo as cinco afirmações:

1. O n -ésimo número perfeito tem n dígitos.
2. Todos os números perfeitos são pares.
3. Todos os números perfeitos terminam em 6 e 8 alternadamente.
4. O algoritmo de Euclides para gerar números perfeitos fornecerá todos os números perfeitos, ou seja, todo número perfeito é da forma $2^{k-1}(2^k - 1)$, para algum $k > 1$, onde $2^k - 1$ é primo.
5. Existem infinitos números perfeitos.

Posteriormente, descobriu-se que duas destas afirmações eram falsas. Devido a quantidade de casas dos quatro primeiros números perfeitos, Nicômaco sugeriu que o quinto número perfeito tivesse 5 dígitos. No entanto, em 1536, Hudalrichus Regius provou que essa afirmação estava equivocadamente errada, mostrando que $2^{13} - 1 = 8191$ é primo, descobrindo, dessa forma, o quinto número perfeito $2^{12}(2^{13} - 1) = 33550336$, mostrando que a afirmação (1) é falsa.

Em 1603, Cataldi verificou que $2^{16}(2^{17} - 1) = 8589869056$ resultava no sexto número perfeito, mostrando que $2^{17} - 1 = 131071$ é também primo. Esse resultado provou que a afirmação de Nicômaco de que os números perfeitos terminavam em 6 ou 8 alternadamente era falsa, pois o sexto número perfeito também termina em 6. Cataldi também foi capaz de encontrar o sétimo número perfeito, a saber $2^{18}(2^{19} - 1) = 137438691328$, encontrando também mais um primo da forma $2^{19} - 1 = 524287$.

PE. MARIN MERSENNE

O monge Marin Mersenne (1588-1648) foi um teólogo, filósofo e matemático francês que se destacou pelo fato do seu nome estar associado a uma família de números primos: o famoso primos de Mersenne.

Conforme destaca Andrade (2018), Mersenne dedicou grande parte do seu tempo estudando esses primos, no qual ele sempre trocava correspondência com outros matemáticos, principalmente com o famoso matemático francês Pierre de Fermat, discutindo as propriedades dos números primos. No ano de 1644, ele publicou em sua obra denominada de “Cogitata physica mathematica”, na qual ele afirma que se $M_p = 2^p - 1$ é primo, então $2^{p-1}(2^p - 1)$ é um número perfeito para:

$$p = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$$

Repare que alguns números primos menores, como o 11, 23, 29 e 37 não se encontram listados acima. Isso ocorre porque nem todo M_p é um número primo.

No entanto, segundo aponta a University of St Andrews (s.d.), Mersenne não testou todos esses números, dando abertura para que outros matemáticos os verificassem com mais precisão, como ocorreu com Euler, quando este provou que $2^{31} - 1$ era primo. Após um século, em 1876, Lucas Lehmer verificou que $2^{127} - 1$ também era primo. Posteriormente, Pervouchine e Powers provaram que $2^{61} - 1$, $2^{89} - 1$, $2^{107} - 1$ também eram primos, mostrando que Mersenne havia deixado de considerar esses números. Por fim, em 1922 Kraitichik mostrou que Mersenne havia errado ao

considerar 257, pois verificou que $2^{257} - 1$ não é primo. Dessa forma, em 1947 ficou definido o intervalo de Mersenne para todos os inteiros positivos $p \leq 258$:

$$p = 2, 3, 5, 7, 13, 17, 31, 61, 89, 107 \text{ e } 127$$

LEONHARD EULER

O estudo teórico dos números perfeitos também teve a contribuição significativa do ilustre matemático suíço Leonhard Euler (1707-1785). Como aponta a University of St Andrews (s.d.), em seus manuscritos, Euler provou a inversa do método de Euclides, mostrando que pode-se obter um número par perfeito através da fórmula apresentada por Euclides.

Esse matemático suíço demonstrou uma notável capacidade de investigar profundamente as características desses números, avançando consideravelmente em suas pesquisas a respeito deles. Além de provar que todos os números perfeitos terminam em 6 ou 8, embora não de forma alternada, ele também se dedicou ao estudo da existência de números perfeitos ímpares. Sobre a existência de números perfeitos ímpares, é reconhecido que não há possibilidade de que tais números existam, porém continua um problema em aberto.

Dessa forma, uma consequência do resultado de Euler é que a cada primo de Mersenne $M_p = 2^p - 1$ corresponde um número perfeito par $2^{p-1}(2^p - 1)$ e vice-versa. Assim, descobrir novos primos de Mersenne é o mesmo que descobrir novos números perfeitos pares. Como já era conhecido o oitavo primo de Mersenne $M_{31} = 2^{31} - 1$, Euler foi a primeira pessoa na Europa a encontrar um novo número perfeito em quase dois séculos $2^{30}(2^{31} - 1) = 2305843008139952128$.

3 OS MISTERIOSOS NÚMEROS PRIMOS

O termo aritmética vem do grego arithmos, que significa número, e technes, que se traduz por ciência. Ainda em relação ao sentido literal dessa palavra, de acordo com Melo Teles, o Novo Dicionário Aurélio define aritmética como “a arte dos números”, cujo o principal foco é estudar a divisibilidade e outras propriedades dos números inteiros.

Aqui o principal personagem são os números primos e sua capacidade de produzir todos os números naturais maiores que um, como veremos no Teorema Fundamental da Aritmética. Apesar da definição simples, ainda se tem muitos mistérios sobre os primos ainda não descobertos: fórmulas que geram números primos, testes que decidam se um número é primo, e como os primos estão distribuídos.

Nesta seção apresentaremos o Teorema Fundamental da Aritmética e um pouco sobre a distribuição dos números primos.

TEOREMA FUNDAMENTAL DA ARITMÉTICA

Definição 3.1. Um número natural maior do que 1 e que só é divisível por 1 e por si próprio é chamado de número primo.

Dados dois números primos p e q e um número natural a qualquer, decorrem da definição acima os seguintes fatos:

1. Se $p \mid q$, então $p = q$. De fato, como $p \mid q$ e sendo q primo, temos que $p = 1$ ou $p = q$. Sendo p primo, tem-se que $p > 1$, o que acarreta $p = q$
2. Se $p \nmid a$, então $\text{mdc}(p, a) = 1$. De fato, se $\text{mdc}(p, a) = d$, temos que $d \mid p$ e $d \mid a$. Portanto, $d = p$ ou $d = 1$. Mas $d \neq p$, pois $p \nmid a$ e, conseqüentemente, $d = 1$

Definição 3.2. Chama-se número composto a todo número maior do que 1 e que não é primo. Portanto, se um número n é composto, existirá um divisor n_1 de n tal que $n_1 \neq 1$ e $n_1 \neq n$. Assim existirá um número natural n_2 tal que $n = n_1 \cdot n_2$, com $1 < n_1 < n$ e $1 < n_2 < n$

Por exemplo: 2, 3, 5, 7, 11 e 13 são números primos, enquanto que 4, 6, 8, 9, 10 e 12 são compostos.

No que se refere a estrutura multiplicativa dos números naturais, de forma simultânea, os números primos são simples e suficiente para gerar todos os números naturais, conforme mostraremos a seguir no Teorema Fundamental da Aritmética.

Teorema 3.1. (Teorema Fundamental da Aritmética) Todo número natural maior do que 1 ou é primo ou se escreve de modo único (a menos da ordem dos fatores) como um produto de números primos.

Demonstração. Façamos a demonstração por indução sobre n . Suponhamos que n seja um número natural, e se n é primo, nada temos a demonstrar. Dessa forma, consideremos que n seja composto. Logo existem números naturais n_1 e n_2 tais que $n = n_1 \cdot n_2$, com $1 < n_1 < n$ e $1 < n_2 < n$. Pela hipótese de indução, temos que existem números primos $p_1, \dots, p_r$ e $q_1, \dots, q_s$ tais que $n_1 = p_1 \cdot \dots \cdot p_r$ e $n_2 = q_1 \cdot \dots \cdot q_s$. Portanto, $n = p_1 \cdot \dots \cdot p_r \cdot q_1 \cdot \dots \cdot q_s$.

Para provar a unicidade, suponha $n = q_1 \cdot q_2 \cdot \dots \cdot q_s$ ($s \geq 1$), onde os p_i 's e os q_i 's são números primos. Logo $p_1 \cdot p_2 \cdot \dots \cdot p_r = q_1 \cdot q_2 \cdot \dots \cdot q_s$. Como $p_1 \mid q_1 \cdot q_2 \cdot \dots \cdot q_s$ então p_1 divide pelo menos um dos fatores q_j . Sem perda de generalidade, suponha que $p_1 \mid q_1$ e como ambos são primos entre si $p_1 = q_1$. Cancelando p_1 com q_1 na igualdade inicial obtemos $p_2 \cdot p_3 \cdot \dots \cdot p_r = q_2 \cdot q_3 \cdot \dots \cdot q_s$.

De maneira mais didática a fim de mostrar que um número natural $n > 1$ é escrito como um produto de fatores primos de modo único (unicidade), podemos escrevê-la da seguinte forma:

$$n = p_1 \cdot \dots \cdot p_r = q_1 \cdot \dots \cdot q_s$$

Como p_r e q_s são primos, e $p_1 \mid q_1 q_2 \cdot \dots \cdot q_s$, temos que $p_1 = q_1$, então

$$\frac{n}{p_1} = \frac{\cancel{p_1} p_2 \cdot \dots \cdot p_r}{\cancel{p_1}} = \frac{\cancel{q_1} q_2 \cdot \dots \cdot q_s}{\cancel{q_1}}$$

$$\frac{n}{p_1} = p_2 \cdot \dots \cdot p_r = q_2 \cdot \dots \cdot q_s$$

Agora, observamos que $p_2 \mid q_2 \cdot q_2 \cdot \dots \cdot q_s$ então p_2 divide pelo menos um dos fatores q_j , sem perda de generalidade, podemos supor que $p_2 \mid q_2$ e como ambos são primos entre si $p_2 = q_2$. Então:

$$\frac{n}{p_1 p_2} = \frac{\cancel{p_3} \cdot \dots \cdot p_r}{\cancel{p_2}} = \frac{\cancel{q_3} \cdot \dots \cdot q_s}{\cancel{q_2}}$$

$$\frac{n}{p_1 p_2} = p_3 \cdot \dots \cdot p_r = q_3 \cdot \dots \cdot q_s$$

Continuando com este processo até quando for necessário, obteremos $1 = q_{r+1} \cdot \dots \cdot q_s$. Logo devemos ter $q_{r+1} = \dots = q_s = 1$ e assim chegamos a conclusão que $r = s$ e assim provamos a unicidade.

Exemplo 3.1. Considere o número $n = 120$.

Resolução:

- A fatoração de 120 em primos é:
 $120 = 2^3 \cdot 5 \cdot 3$
- Suponhamos que haja outra forma de escrever o número 120 como um produto de primos:
 $120 = 2 \cdot 3 \cdot 2 \cdot 5$
- Note que é basicamente a mesma coisa, apenas alterou a ordem dos fatores:
 $120 = 2^3 \cdot 5 \cdot 3 = 2 \cdot 3 \cdot 2 \cdot 5$

Portanto, concluímos que, independentemente da ordem dos fatores, os primos devem ser os mesmos, provando a unicidade.

Em outras palavras, o Teorema Fundamental da Aritmética nos diz que todos os números naturais, maiores que 1, ou é, primo ou pode ser decomposto num produto único de números

primos (fatores primos). Costuma-se escolher como padrão a decomposição cujos primos estão ordenados de modo crescente. Vejamos abaixo um exemplo:

Exemplo 3.2. O número 2100 é escrito de maneira única, a menos pela ordem dos fatores, como $2^2 \times 3^1 \times 5^2 \times 7^1$, que são números primos.

Exemplo 3.3. Achando as decomposições de $a = 588$ e $b = 936$, temos que $a = 588 = 2^2 \cdot 3 \cdot 7^2$ e $b = 936 = 2^3 \cdot 3^2 \cdot 13$

CRIVO DE ERATÓSTENES

Após o estudo simples dos números primos, faremos uma breve abordagem sobre a distribuição desses números. Inicialmente provamos a sua infinitude e posteriormente apresentamos um método para obter os números primos até uma determinada ordem usando um famoso algoritmo próprio para isso: o *Crivo de Eratóstenes*

Teorema 3.2. *Existem infinitos números primos.*

Demonstração. Suponha, por absurdo, que exista apenas um número finito de números primos $p_1, \dots, p_r$. Considere o número natural

$$n = p_1 p_2 \cdot \dots \cdot p_r + 1 \quad (1)$$

Pelo Teorema 3.1, o número n possui um fator primo p que, portanto, deve ser um dos $p_1, \dots, p_r$ e, conseqüentemente, divide o produto $p_1 p_2 \cdot \dots \cdot p_r$. Mas isto implica que p divide 1, o que é absurdo.

Agora que sabemos que existem infinitos números primos, surge-nos a seguinte indagação: como obter uma lista desses números até uma determinada ordem? Bom, o Crivo de Eratóstenes é um dos métodos mais antigos, geralmente utilizados para se determinar números primos. Esse algoritmo consiste basicamente na construção de uma tabela, onde são listados todos os números primos de ordem menos elevada, uma vez que o conjunto desses números é finito.

Dado um número $N > 0$ escrevemos em uma tabela todos os números inteiros menores ou iguais a N para determinar todos os primos menores que esse número. Por exemplo, para $N = 150$ temos a seguinte tabela:

Figura 1 – Crivo de Eratóstenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150

Fonte: Elaborado pela autora.

Maneira de solução: Escrevemos todos os números naturais de 1 a 150. Riscam-se, de modo sistemático, todos os números compostos da tabela, seguindo o roteiro abaixo:

1. Risque todos os múltiplos de 2 acima de 2, já que nenhum deles é primo;
2. Em seguida, risque todos os múltiplos de 3 maiores que 3, com exceção do próprio 3, já que ele é primo;
3. Depois risque todos os múltiplos de 5, com exceção dele mesmo, já que o 5 é um número primo;
4. O quarto número não riscado que aparece é o 7, que é primo. Risque todos os múltiplos de 7 maiores do que 7 pois esses não são primos.

É necessário prosseguir com este procedimento até chegar a 150? A resposta é não e se baseia no seguinte resultado devido ao próprio Eratóstenes.

Lema 3.1. *Se um número natural $n > 1$ não é divisível por nenhum número primo p tal que $p^2 \leq n$, então ele é primo.*

Demonstração. Suponhamos, por absurdo, que n não seja divisível por nenhum número primo p tal que $p^2 \leq n$ e que não seja primo. Seja q o menor número primo que divide n então, $n = qn_1$, com $q \leq n_1$. Segue daí que $q^2 \leq qn_1 = n$. Logo, n é divisível por um número primo q tal que $q^2 \leq n$, o que é uma contradição.

Dessa forma, temos na nossa tabela os números de 1 a 150, devemos riscar todos os números até alcançar o primo 11, pois o próximo primo é 13, cujo quadrado supera 150.

4 A RELAÇÃO ENTRE PERFEIÇÃO E PRIMALIDADE

Como vimos anteriormente em nosso breve passeio pela história, em busca de repostas sobre os números perfeitos foram encontrados alguns primos especiais chamados posteriormente de primos de Mersenne, em homenagem ao matemático que os formulou. Nesta seção, apresentaremos o conceito dos números perfeitos e dos primos de Mersenne, finalizando com a demonstração do Teorema de Euclides-Euler, que estabelece a relação entre esses dois tipos de números.

NÚMEROS PERFEITOS E PRIMOS DE MERSENNE

Os números perfeitos são definidos da seguinte forma:

Definição 4.1. *Um número natural $n > 1$ se diz um número perfeito quando for igual a soma de seus divisores próprios.*

Assim, para sabermos se um número natural $n > 1$ é perfeito, temos que encontrar todos os divisores de n ; somar todos os seus divisores de n , exceto n ; verificar se a soma obtida é n ou não.

Se essa soma for menor que o próprio número é chamado de *deficiente*, caso contrário, se a soma for maior que o número é dito *abundante*.

De acordo com University of St Andrews(s.d.), os quatro primeiros números perfeitos foram descobertos pelos povos da antiguidade e não há registros evidentes de sua descoberta. Atualmente, são contabilizados 52 números perfeitos, sendo que o último da lista apresenta 41024320 dígitos. Optamos por não apresentá-lo aqui, pois a quantidade de dígitos ultrapassaria os limites da página ou exigiria várias folhas para ser completamente exibido. Construímos a seguir, uma tabela que contém os onze primeiros números perfeito em função do número primo p que aparece na fórmula que os geram, que veremos mais tarde.

Tabela 1 - Números perfeitos e seus respectivos primos de Mersenne associados a diferentes valores de p

p	Números perfeitos
2	6
3	28
5	496
7	8128
13	33550336
17	8589869056
19	137438691328
31	2305843008139952128
61	2658455991569831744654692615953842176
89	191561942608236107294793378084303638130997321548169216
107	1316403645869648337239735460458722910223472318386943117783728128

Fonte: Elaborado pela autora.

Definição 4.2. Os números primos da forma $2^p - 1$, onde p é um número primo, são chamados *Primos de Mersenne*. Denotamos por $M_p = 2^p - 1$

Podemos definir no intervalo $2 \leq p \leq 5000$ os números de Mersenne que são chamados de primos de Mersenne, que correspondem aos seguinte valores de p : 2, 3, 5, 7, 13, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, 3217, 4253 e 4423.

Segundo o blog da OBMEP (s.d.), em 1996 foi criado um grupo que reúne voluntários de todo o mundo com o intuito de encontrar números de Mersenne: o **Great Internet Mersenne Prime Search - GIMPS**. Trata-se, portanto, de uma comunidade que instala em seus computadores altamente potentes alguns softwares que processam continuamente números enviados por um servidor central. Essa comunidade oferece um prêmio em dinheiro para aquele que encontrar números perfeitos e primos de Mersenne. Você também pode participar da busca por esses números, basta acessar : www.mersenne.org.

Os números perfeitos e números primos de Mersenne que são definidos no intervalo

de $2 \leq p \leq 136279841$ podem ser verificados na página oficial da GIMPS: <https://www.mersenne.org/primes/>. Construímos a seguir, uma tabela que contém os onze primeiros primos de Mersenne, em que $2 \leq p \leq 107$.

Tabela 2 - Números perfeitos e seus respectivos primos de Mersenne associados a diferentes valores de p

p	Primos de Mersenne
2	3
3	7
5	31
7	127
13	8191
17	131071
19	524287
31	2147483647
61	2305843009213693951
89	618970019642690137449562111
107	162259276829213363391578010288127

Fonte: Elaborado pela autora.

Geralmente, os números primos são utilizados na criptografia para proteger informações na forma de código, ou seja, são basicamente usados no sistema de segurança de dados, em um banco ou militares e até mesmo para realizarmos uma compra segura na internet. Segundo Marreiros et al. (2024, p. 1):

Existe uma afirmação (Teorema Fundamental da Aritmética) que diz que todo número é primo ou produto de primos, o que significa que um número com 30 milhões de dígitos pode ser escrito de forma única como um produto de números primos. Dessa forma, uma pessoa pode esconder informações em números e só quem sabe quais são os primos que multiplicados dão aquele número é que saberá qual é o código. A vantagem disso é que leva muito tempo para se calcular os fatores primos de um número grande e, quando alguém finalmente descobre o código, esse número já mudou e a descoberta não serve para nada. (MARREIROS, 2024)

Dessa forma, com o avanço da tecnologia e a constante evolução dos computadores, que se tornam cada vez mais rápidos e eficientes, a busca por esses números se torna mais acessível, impulsionando, assim, o desenvolvimento dessa área da matemática, além de contribuir significativamente para a

segurança de dados confidenciais, possibilitando que as pessoas naveguem com mais tranquilidade e proteção pela internet.

RELAÇÃO ENTRE NÚMEROS PERFEITOS E PRIMOS DE MERSENNE

Nessa seção forneceremos a fórmula para calcular a soma dos divisores de um número natural, e posteriormente faremos a demonstração do Teorema de Euclides-Euler, e consequentemente, a sua recíproca.

Para realizar a demonstração usaremos o livro *Fundamentos da Aritmética* de Hygino Domingues e o livro intitulado de Elementos da Aritmética de Abramo Hefez. De acordo com Hygino Domingues em sua obra denominada *Fundamentos da Aritmética*:

Seja n um número natural maior do que 1, denotemos por $S(n)$ a soma de todos os seus divisores.

Se $n \geq 2$, o próximo resultado nos fornecerá uma fórmula para $S(n)$ em função da decomposição de n em fatores primos.

Proposição 4.1. *Seja $n = p_1^{\alpha_1} \dots p_r^{\alpha_r}$, onde $p_1, \dots, p_r$ são números primos e $\alpha_1, \dots, \alpha_r \in \mathbb{N}^*$. Então,*

$$S(n) = \frac{p_1^{\alpha_1+1} - 1}{p_1 - 1} \cdot \frac{p_2^{\alpha_2+1} - 1}{p_2 - 1} \dots \frac{p_r^{\alpha_r+1} - 1}{p_r - 1}$$

Demonstração. Se p é primo, como os únicos divisores de p são 1 e p , então $S(p) = 1 + p = \frac{p^2 - 1}{p - 1}$. Se p é primo e $\alpha \in \mathbb{N}$, então os divisores de p^α são $\alpha + 1$ números:

$$1, p, p^2, \dots, p^\alpha$$

Portanto, no caso em que temos apenas um fator primo, vale que:

$$S(p^\alpha) = 1 + p + \dots + p^\alpha = \frac{(p^\alpha + p^{\alpha-1} + \dots + p + 1)(p - 1)}{p - 1} = \frac{p^{\alpha+1} - 1}{p - 1}$$

Agora consideremos o caso com mais de um fator primo distinto. Como já vimos a proposição é válida para $r = 1$. Seja $r > 1$ e admitamos que seja verdadeira para $r - 1$. Faremos a prova por indução sobre r . Supondo

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_r^{\alpha_r} \quad (2)$$

Com $p_1, p_2, \dots, p_r$ primos distintos. Conforme o enunciado, façamos

$$p_2^{\alpha_2} \cdot p_3^{\alpha_3} \cdot \dots \cdot p_r^{\alpha_r} = m \quad (3)$$

Com $p_2, p_3, \dots, p_{r-1}$ primos e sejam $1 = d_1, d_2, \dots, d_k = m$ os divisores de m . Como um divisor de n é necessariamente do tipo $p_1^{\beta_1} \cdot p_2^{\beta_2} \cdot \dots \cdot p_r^{\beta_r}$ ($0 \leq \beta_i \leq \alpha_i; i = 1, 2, \dots, r$) Então esses

divisores são da forma:

$$p_1^{\beta_i} \cdot d_i (i = 1, 2, \dots, k) \quad (4)$$

Logo,

$$\sum_{\beta_1=0}^{\alpha_1} \sum_{i=1}^k p_1^{\beta_1} d_i = \left(\sum_{\beta_1=0}^{\alpha_1} p_1^{\beta_1} \right) \cdot \left(\sum_{i=1}^k d_i \right) = S(p_1^{\alpha_1}) S(m) = \frac{p_1^{\alpha_1+1} - 1}{p_1 - 1} \cdot S(m)$$

Mas, pela hipótese de indução:

$$S(m) = \frac{p_2^{\alpha_2+1} - 1}{p_2 - 1} \cdot \dots \cdot \frac{p_r^{\alpha_r+1} - 1}{p_r - 1} \quad (5)$$

Donde:

$$S(n) = \frac{p_1^{\alpha_1+1} - 1}{p_1 - 1} \cdot \frac{p_2^{\alpha_2+1} - 1}{p_2 - 1} \cdot \dots \cdot \frac{p_r^{\alpha_r+1} - 1}{p_r - 1} \quad (6)$$

Portanto, por indução, a fórmula da soma dos divisores é verdadeira. □

Exemplo 4.1. Como $20 = 2^2 \cdot 5$, então

$$s(20) = \frac{2^3 - 1}{2 - 1} \cdot \frac{5^2 - 1}{5 - 1} = 7 \cdot 5 = 42$$

é a soma dos divisores de 20. Vamos tirar a prova:

Sabemos que os divisores de 20 são 1, 2, 4, 5, 10 e 20, então efetuando a soma temos:

$$1 + 2 + 4 + 5 + 10 + 20 = 42$$

Após a obtenção do resultado acima, Abramo Hefez expõe o seguinte corolário:

Corolário 4.1. A função $S(n)$ é multiplicativa, isto é, se $\text{mdc}(u, v) = 1$, então $S(uv) = S(u) \cdot S(v)$.

Demonstração. Se u e v são dois números inteiros positivos tais que o $\text{mdc}(u, v) = 1$, e se $u = 1$ ou $v = 1$, então:

$$S(uv) = S(u) \cdot S(v)$$

Agora suponhamos que $u > 1$ e $v > 1$ e sejam:

$$u = p_1^{k_1} p_2^{k_2} \cdot \dots \cdot p_i^{k_i}, v = q_1^{h_1} q_2^{h_2} \cdot \dots \cdot q_j^{h_j}.$$

Como $p_x \neq q_y$ para $x = 1, 2, \dots, i$ e $y = 1, 2, \dots, j$, visto que o $\text{mdc}(u, v) = 1$, segue que a *decomposição canônica* do produto uv é dada pela igualdade:

$$uv = p_1^{k_1} p_2^{k_2} \cdot \dots \cdot p_i^{k_i} q_1^{h_1} q_2^{h_2} \cdot \dots \cdot q_j^{h_j}$$

E, portanto:

$$S(uv) = \frac{p_1^{k_1+1} - 1}{p_1 - 1} \cdot \frac{p_2^{k_2+1} - 1}{p_2 - 1} \cdot \dots \cdot \frac{p_i^{k_i+1} - 1}{p_i - 1} \cdot \frac{q_1^{h_1+1} - 1}{q_1 - 1} \cdot \frac{q_2^{h_2+1} - 1}{q_2 - 1} \cdot \dots \cdot \frac{q_j^{h_j+1} - 1}{q_j - 1}$$

Assim, também obtemos $S(uv) = S(u)S(v)$. □

Exemplo 4.2. Vejamos alguns exemplos abaixo

$$\bullet S(2) = \frac{2^2 - 1}{2 - 1} = 3$$

$$\bullet S(3) = \frac{3^2 - 1}{3 - 1} = 4$$

$$\bullet S(5) = \frac{5^2 - 1}{5 - 1} = 6$$

$$S(6) = S(2 \cdot 3) = S(2)S(3) = \frac{2^2 - 1}{2 - 1} \cdot \frac{3^2 - 1}{3 - 1} = 12$$

$$\bullet S(10) = S(2 \cdot 5) = S(2)S(5) = \frac{2^2 - 1}{2 - 1} \cdot \frac{5^2 - 1}{5 - 1} = 18$$

$$\bullet S(18) = S(2 \cdot 3^2) = S(2)S(3^2) = \frac{2^2 - 1}{2 - 1} \cdot \frac{3^3 - 1}{3 - 1} = 39$$

Dado o corolário anterior, examinemos essa propriedade: Sabemos que se $\text{mdc}(n, m) = 1$, ou seja se n e m forem primos entre si, então:

$$S(n \cdot m) = S(n) \cdot S(m)$$

Dessa forma, temos que:

$$\text{mdc}(2, 5) = 1$$

$$S(2) \cdot S(5) = 3 \cdot 6 = 18$$

$$S(2 \cdot 5) = S(10) = 18$$

Portanto o corolário acima vale para $\text{mdc}(n, m) = 1$. Mas cuidado, note que $S(18) = 39 \neq 48 = S(3) \cdot S(6)$; e portanto, a conclusão do corolário acima não vale se $(n, m) \neq 1$.

O lema a seguir será importante para conseguirmos demonstrar o Teorema Euclides-Euler, e nos fornece uma caracterização de números primos usando a soma dos seus divisores.

Lema 4.1. *Seja $n \in \mathbb{N}^*$. Tem-se que $S(n) = n + 1$ se, e somente se, n é um número primo. Demonstração.*
Se $S(n) = n + 1$, segue que $n > 1$ e que os únicos divisores de n são 1 e n ; logo n é primo.

$$S(n) = \frac{n^2 - 1}{n - 1} = n + 1 \quad \square$$

Reciprocamente, se n é primo, da proposição 4.1, segue-se que

Usando a notação $S(n)$ para a soma de todos os divisores de n , temos que n é um número perfeito se, e somente se, $S(n) = 2n$. Analogamente, n é um número deficiente se $S(n) < 2n$ e é um número abundante se $S(n) > 2n$. A seguir, apresentamos a demonstração do Teorema Euclides-Euler que faz a relação entre números perfeitos e primos de Mersenne.

Teorema 4.1. (Euclides-Euler). *Um número n é um número perfeito par se, e somente se, $n = 2^{p-1}(2^p - 1)$, onde $2^p - 1$ é um primo de Mersenne.*

Demonstração. Suponha que p é um número primo e $n = 2^{p-1}(2^p - 1)$, em que $2^p - 1$ é um primo de Mersenne. Vamos mostrar que n é par e perfeito, pois $S(n) = 2n$

De fato, como temos que 2^{p-1} é ímpar e $2^p - 1$ é par, devemos ter que o $\text{mdc}(2^{p-1}, 2^p - 1) =$

1. Usando que $S(n)$ é multiplicativa, temos: $S(n) = S(2^{p-1}(2^p - 1)) = S(2^{p-1})S(2^p - 1)$. Pela Proposição 4.1 temos:

$$S(2^{p-1}) = \frac{2^{p-1+1} - 1}{2 - 1} = 2^p - 1$$

como $2^p - 1$ é primo, segue que o Lema 4.1 é:

$$S(2^p - 1) = 2^p - 1 + 1 = 2^p$$

Substituindo em $S(n)$ temos:

$$S(n) = S(2^{p-1})S(2^p - 1) \quad (7)$$

$$S(n) = (2^p - 1)2^p \quad (8)$$

$$S(n) = [(2^p - 1)2^{p-1}] \cdot 2 = 2n \quad (9)$$

Portanto, n é perfeito par.

Agora vamos demonstrar a recíproca. Suponha que $n \in \mathbb{N}^*$ é par e perfeito. Seja 2^{p-1} a maior potência de 2 que divide n . Logo, $p \geq 2$, ou seja, $p > 1$, sendo $n = 2^{p-1} \cdot b$, para algum b ímpar.

Temos então que $\text{mdc}(2^p - 1, b) = 1$ e pela Proposição 4.1 e o Corolário 4.1 segue que:

$$S(n) = S(2^{p-1} \cdot b) = S(2^{p-1})S(b) = (2^p - 1) \cdot S(b)$$

Sabemos que $S(n) = 2n$, então fazendo a substituição temos:

$$S(n) = (2^p - 1) \cdot S(b) \quad (10)$$

$$2n = (2^p - 1) \cdot S(b) \quad (11)$$

$$2 \cdot (2^{p-1} \cdot b) = (2^p - 1) \cdot S(b) \quad (12)$$

$$2^p \cdot b = (2^p - 1) \cdot S(b) \quad (13)$$

$$S(b) = \frac{2^p \cdot b}{2^p - 1} \quad (14)$$

Dessa forma, chegamos a conclusão que $2^p - 1 \mid 2^p \cdot b$.

Como $\text{mdc}(2^p - 1, 2^p) = 1$ temos que $2^p - 1 \mid b$. Então existe $c \in \mathbb{N}^*$, com $c < b$, tal que

$$b = (2^p - 1) \cdot c \quad (15)$$

Substituindo (15) em (13) obtemos:

$$(2^p - 1)S(b) = 2^p \cdot b \quad (16)$$

$$(2^p - 1)S(b) = 2^p \cdot (2^p - 1) \cdot c \quad (17)$$

$$S(b) = 2^p \cdot c \quad (18)$$

De outra forma, somando c de ambos os lados em (15), temos:

$$b + c = (2^p - 1) \cdot c + c \quad (19)$$

$$b + c = (2^p - 1 + 1)c \quad (20)$$

$$b + c = 2^p \cdot c \quad (21)$$

$$b + c = S(b) \quad (22)$$

Note que $c = 1$. Caso contrário, b tem pelo menos três divisores distintos 1 , c e b . Neste caso, $S(b) \geq 1 + c + b > b + c = S(b)$, o que é uma contradição.

Portanto, de (15) e (22) segue-se que $S(b) = b + 1$ e $b = 2^p - 1$. Logo, pelo Lema 4.1, $b = 2^p - 1$ é primo. Temos, assim, que $n = 2^{p-1}(2^p - 1)$ com $2^p - 1$ sendo um primo de Mersenne. □

NÚMEROS PERFEITOS E NÚMEROS BINÁRIOS

Atualmente, o nosso sistema numérico é constituído pela base decimal, que são formados por todos os números de ordem posicional compreendidos entre 0 a 9. Já o sistema de numeração binário, como o próprio nome sugere, apresenta base 2, na qual os números 0 e 1 são os dígitos deste sistema. O sistema binário apresenta uma grande importância para a Ciências Exatas, especialmente na área da Ciências da Computação e Eletrônica, pois ele facilita a operação de circuitos digitais, garantindo o funcionamento de computadores e dispositivos eletrônicos.

A REAMAT, uma Revista Eletrônica de Matemática originária de um projeto de extensão do Departamento de Matemática e Estatística do Rio Grande do Sul, menciona a conceituação de sistema binário da seguinte forma:

Exemplo 4.3. *O sistema de numeração em base dois é chamado de binário e os algarismos binários são conhecidos como bits (do inglês binary digits). Um bit pode assumir dois valores distintos: 0 ou 1. Por exemplo:*

$$x = (1001, 101)_2 \quad (23)$$

$$= 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \quad (24)$$

$$= 8 + 0 + 0 + 1 + 0,5 + 0 + 0,125 = 9,625 \quad (25)$$

O procedimento realizado acima representa a conversão de um número representado na base binária para o sistema decimal, mostrando que $(1001, 101)_2$ é igual a 9,625 no sistema decimal.

Sabemos o quanto é complicado determinar números perfeitos manualmente, pois a maioria possuem formas muito extensa. Com o surgimento de novos computadores altamente potentes e bem desenvolvidos tecnologicamente foi possível gerar mais números perfeitos e encontrar novas propriedades sobre esses números.

Segundo Marreiros et al.(2024) uma dessas descobertas foi que todo número perfeito escrito em linguagem binária é formado por uma sequência de n uns por $n - 1$ zeros, onde n é o mesmo n da fórmula: $2^{n-1} \cdot (2^n - 1)$.

Isto é fácil de provar, basta observar que:

$$2^n - 1 = \underbrace{2^{n-1} + \dots + 2^2 + 2 + 1}_{n \text{ parcelas}}^x$$

$$2^{n-1}(2^n - 1) = \underbrace{2^{2n-2} + \dots + 2^{n+1} + 2^n + 2^{n-1}}_{n \text{ parcelas}}^x$$

Expomos alguns exemplos a seguir para melhorar a nossa compreensão:

1. $2(2^2 - 1) = 6 = (110)_2$
2. $2^2(2^3 - 1) = 28 = (11100)_2$
3. $2^4(2^5 - 1) = 496 = (111110000)_2$
4. $2^6(2^7 - 1) = 8.128 = (1111111000000)_2$
5. $2^{12}(2^{13} - 1) = 33.550.336 = (111111111111000000000000)_2$

Após a exposição desses exemplos observa-se que, no exemplo 1, temos 2 bits 1 representado por $n = 2$ na fórmula dada anteriormente, seguido por 1 bit 0, resultado dado por $n - 1 = 1$ que também se encontra na fórmula.

Outra característica que Marreiros et al (2024) também destaca é que durante a conversão dos números perfeitos para os números binários é possível encontrar algumas somas que possui como resultados números primos e outros números quadrados perfeitos, como pode ser observado nos exemplos a seguir:

Tabela 3 - Conversão de números perfeitos para binário com contagem de bits

Decimal	Binários	bits 1	bits 0	soma
6	110	2	1	3
28	11100	3	2	5
496	111110000	5	4	9
8128	111111100000	7	6	13
33550336	111111111111100000000000	13	12	25
137438691328	1111111111111111000000000000000000	19	18	37

Fonte: Elaborado pela autora.

Ao observar a tabela, note que a soma dos bits 0 e 1, que compõem o número perfeito 6, resulta no número primo 3. Já a soma dos bits dos números perfeitos 496 e 33550336 resulta em um quadrado perfeito. Vale ressaltar que nem toda soma terá esta característica, por exemplo, a somas dos bits do número perfeito $216(217 - 1) = 8589869056$ é igual a 33, que não é primo nem quadrado perfeito. Apesar de não termos afirmações sempre válidas, esta observações de padrões atizam a curiosidade.

5 CONCLUSÃO

A partir do que foi exposto neste trabalho, podemos concluir que os números perfeitos possuem grande importância para a sociedade. Além disso, despertam fascínio em muitas pessoas por apresentarem características cabalísticas e propriedades matemáticas interessantes, como a disposição dos bits 1 e 0 quando convertidos para a numeração binária. Ainda foi possível descrever a contribuição de diversos matemáticos ao longo da história na busca por números perfeitos, seja pelas fascinantes propriedades que esses números apresentam, seja por sua aura mística.

Outro aspecto relevante observado foi a sua relação com os números primos de Mersenne, que, por serem extremamente grandes, são amplamente utilizados nas áreas de criptografia e computação —

especialmente na segurança de dados em transações online e em acessos confidenciais a determinados sites. Também é importante destacar a abordagem algébrica empregada para demonstrar o Teorema de Euclides-Euler, o qual evidencia que todo número perfeito par pode ser gerado a partir de primos de Mersenne. Ao longo do documento, analisamos também a aplicabilidade do Teorema Fundamental da Aritmética na codificação de dados confidenciais, na qual esse teorema estabelece que todo número natural é primo ou pode ser decomposto unicamente como produto de números primos. Dessa forma, até que alguém consiga decodificar um código composto por milhões de dígitos, o número já terá sido alterado, tornando a descoberta inútil.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, D. Números primos e números de Mersenne. *Jornal Eletrônico de Ensino e Pesquisa de Matemática (Jeepema)*, Maringá: v. 2, n. 1, p. 81–89, 2018. Disponível em: <http://www.dma.uem.br/kit/jeepema>. Acesso em: 19 abr. 2025.

CRUZ, S.O. Números Perfeitos, 2013, Tese de Mestrado Profissional em Matemática em Rede Nacional (profmat), Centro de Ciências Exatas e da Natureza, UFPB, 2013, p. 9-10, p. 39-42, p.49 - 52.

EVES, H. Introdução a história da Matemática/Howard Eves; tradução: Hygino H.Domingues, Campinas, SP: Editora da UNICAMP, p.98 - 100, 2004.

HEFEZ, A. Elementos da Aritmética, Rio de Janeiro, 2005, p.82 - 103. HYGINO, H. D. Fundamentos da Aritmética, São Paulo: atual, p.62-65, 1991.

JUSTO, Dagoberto Adriano Rizzotto et al. Cálculo Numérico: um livro colaborativo – versão Python. [S.l.]: UFRGS, 2020. Disponível em: <https://www.ufrgs.br/reamat>. Acesso em: 19 de Abril 2025.

MARREIROS, E. C. N. et al. Números perfeitos: uma abordagem computacional. *Revista FT*, v. 28, n. 135, jun. 2024. Disponível em:

<https://revistaft.com.br/numeros-perfeitos-uma-abordagem-computacional/>. Acesso em: 19 abr. 2025

OBMEP. Números especiais: números perfeitos. Clube de Matemática da OBMEP. Disponível em: <http://clubes.obmep.org.br/blog/numeros-especiais-numeros-perfeitos/>. Acesso em: 19 abr. 2025.

SINGH, Simon.O último teorema de Fermat. Tradução de Jorge Luiz Calife. 12. ed. Rio de Janeiro: Record, 2006, p.33-34.

TELES, Rosinalda Aurora de Melo. A aritmética e álgebra na matemática escolar. In: ENCONTRO NACIONAL DE EDUCAÇÃO MATEMÁTICA, 8., 2004, Recife. Anais [...]. Recife: SBEM, 2004.

UNIVERSITY OF ST ANDREWS. Perfect numbers. MacTutor History of Mathematics archive, [s.d.]. Disponível em: https://mathshistory.st-andrews.ac.uk/HistTopics/Perfect_numbers/ Acesso em: 12 Fev. 2025.

$$\int_{-\infty}^{\infty} \psi_n^*(x) \psi_n(x) dx = 1 \quad \frac{1}{2m} [p^2 + (m\omega x)^2] \psi = E\psi$$

$$\psi(r, \theta, \phi) = R(r)Y(\theta, \phi) \quad \frac{1}{\Theta} \left[\sin \theta \right]$$

$$KE = \frac{1}{2}mv^2 + \frac{1}{2} \cdot (LEM = \frac{1}{2}m) \cdot v^2$$

$$\psi = \sum_{n=1}^{\infty} c_n \psi_n(x) = \sqrt{\frac{2}{a}} \sum_{n=1}^{\infty} c_n \sin\left(\frac{n\pi}{a}x\right)$$

Nuclear radius = $A^{1/3} \cdot 1.2$

$\Phi = \mu_0 A I N^2 / \ell$

$\tau_{1/2} = \ln(2)\tau, N =$

$G \frac{Mm}{r}, P F = m a \hbar (small \hbar), F = G \frac{Mm}{r^2} = mg$

$B\ell = \mu_0 I$ for single wire $B = \frac{\mu_0 I}{2\pi r}$

$c_n = \int \psi_n(x)^* \psi(x) dx$

$$P_a + \frac{1}{2}\rho_a v_a^2 + \rho_a g h_a = P_b + \frac{1}{2}\rho_b v_b^2 + \rho_b g h_b$$

$U_{\text{capacitor}} = Q^2 / (2C)$



Δ

Π

$$\frac{d^2 \Phi}{d\phi^2} = -\mu$$

$E^2/2 + B^2/(2\mu_0) = \text{energy/volume}$

TÓPICOS EM CIÊNCIAS

EXATAS E DA TERRA

VOLUME IX

$L_z = m_\ell \hbar, m_\ell = -\ell, \dots, \ell$

$\Psi_n(\mathbf{r}, t) = \psi_n(\mathbf{r}) e^{-iE_n t / \hbar}$

$F = qvB \sin \theta, F = ILB \sin \theta$

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2} = -k^2$$

Ω

$h = 2.9 \times 10^{-3} \text{ m} \cdot \text{K}$

$c_{\text{vac.}} = 1/\sqrt{\epsilon_0 \mu_0}, c = \lambda f$

Uncertainty

$E = hf, p = \hbar k$

$u = 931 \text{ MeV}/c^2 = 1.67 \times 10^{-27} \text{ kg}$

$V\psi = E\psi$

$k = 8.99 \times 10^9 \text{ Nm}^2/\text{C}^2, \epsilon_0 = 8.85 \times 10^{-12} \text{ C}^2/(\text{Nm})$

$\psi_n(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{n\pi}{a}x\right)$

