UNIVERSIDADE
GREGÓRIO SEMEDO

Faculdade de ngenharia e Novas Tecnologias

Curso de Engenharia Informática

Trabalho de Conclusão de Curso

Desenvolvimento de Software de Gestão de Projetos Acadêmicos

Tomás K. Ferraz Kalemba – 221246

Orientador: *Msc. Helmer Capassola*

Luanda, 2025

Faculdade de ngenharia e Novas Tecnologias

Curso de Engenharia Informática

Trabalho de Conclusão de Curso

Desenvolvimento de Software de Gestão de Projetos Acadêmicos

Tomás K. Ferraz Kalemba – 221246

Orientador: *Msc. Helmer Capassola*

Luanda, 2025

TOMÁS K. FERRAZ KALEMBA

DESENVOLVIMENTO DE SOFTWARE DE GESTÃO DE PROJETOS ACADÊMICOS

Trabalho de Conclusão do Curso apresentado à Universidade Gregório Semedo, como parte dos requisitos para obtenção do grau académico de Licenciado em Engenharia Informática.

Júri

Presidente

_____

(grau académico, nome, cargo).
Universidade Gregório Semedo

Arguente

_____

(grau académico, nome, cargo).
Universidade Gregório Semedo

Orientador

_____

(grau académico, nome, cargo).
Universidade Gregório Semedo

Luanda, _____ de _____de Ano de Defesa

Dedicatória

Dedico este trabalho aos meus pais Tomas Kalemba e Deolinda Kiasauka, meus primeiros mestres, que com amor, dedicação e sacrifício foram os pilares da minha formação. A vocês, que sempre acreditaram em mim, mesmo nos momentos em que eu mesmo duvidei da minha capacidade.

Agradeço pelo apoio incondicional, pelas palavras de incentivo, pelos conselhos sábios e pelo exemplo de integridade, esforço e coragem que sempre me inspirou.

Este trabalho é uma pequena conquista diante de tudo o que vocês já fizeram por mim. Sem vocês, nada disso seria possível.

Com todo o meu amor e gratidão.

Agradecimentos

A realização deste trabalho representa não apenas uma conquista pessoal, mas também o reflexo do apoio, carinho e incentivo de muitas pessoas que caminharam ao meu lado ao longo desta jornada. Agradeço primeiramente a Deus, por me dar forças nos momentos mais difíceis e por iluminar meu caminho até aqui. Aos meus pais, pela base sólida que me deram com amor, valores e dedicação incondicional. Vocês são meus maior exemplo, a minha princesa minha filha kailany tu és a minha principal motivação. Aos meus irmãos, pelo companheirismo, incentivo e apoio constantes. Em cada etapa deste percurso, vocês foram fonte de inspiração e força. Aos meus professores, que com paciência, sabedoria e comprometimento contribuíram imensamente para minha formação acadêmica e pessoal. Cada ensinamento deixado por vocês fez diferença no meu crescimento.

A minha doce e amada parceira que esteve ao meu lado durante essa trajetória: obrigado por acreditar em mim, por compartilhar os desafios e celebrar as vitórias. A tua presença tornou tudo mais leve e significativo. A todos que, de alguma forma, fizeram parte deste capítulo da minha vida e em destaque ao Emanuel Gonga, meus sinceros agradecimentos.

Epígrafe

"Os computadores são inúteis. Eles só podem dar respostas."

Pablo Picasso

Resumo

A gestão de projetos no ambiente acadêmico desempenha um papel fundamental na organização e no sucesso de atividades como pesquisas, trabalhos em grupo e projetos de extensão. No entanto, a ausência de ferramentas específicas para esse contexto ainda representa um desafio significativo para estudantes e instituições. Este trabalho tem como objetivo desenvolver um software de gestão de projetos voltado para o meio universitário, com foco na organização de tarefas, definição de prazos, acompanhamento de metas e melhoria da comunicação entre os membros das equipes. A solução proposta busca oferecer uma interface intuitiva e funcionalidades que atendam às necessidades do contexto acadêmico, promovendo maior produtividade, controle e eficiência. A metodologia utilizada inclui levantamento de requisitos, modelagem do sistema e desenvolvimento de um protótipo funcional. Os resultados esperados incluem a facilitação da gestão de projetos acadêmicos e a contribuição para o aprimoramento da rotina universitária por meio da tecnologia.

Palavras-Chaves: Gestão de projetos, ambiente acadêmico, organização, produtividade, colaboração.

Abstract

Project management in the academic environment plays a fundamental role in organizing and ensuring the success of activities such as research, group assignments, and extension projects. However, the lack of specific tools for this context still represents a significant challenge for students and educational institutions. This work aims to develop project management software tailored to the university setting, focusing on task organization, deadline tracking, and goal monitoring, and improving communication among team members. The proposed solution seeks to offer an intuitive interface and functionalities that meet the specific needs of the academic context, promoting greater productivity, control, and efficiency. The methodology includes requirements gathering, system modeling, and the development of a functional prototype. The expected outcomes include facilitating academic project management and contributing to the improvement of the university routine through technology.

Keywords: Project management, academic environment, organization, productivity, collaboration.

Índice

Lista de Acronimos

FENT – Faculdade de Engenharia e Novas Tecnologias;

Lista de Abreviaturas

UGS – Universidade Gregório Semedo;

1. Introdução

A gestão de projetos acadêmicos é um componente essencial para o sucesso de estudantes e instituições de ensino. Envolve o planejamento, acompanhamento de tarefas, definição de prazos, metas e, sobretudo, a comunicação eficaz entre os membros dos projectos. Em um cenário acadêmico cada vez mais exigente e dinâmico, a organização eficiente dos projetos torna-se um diferencial competitivo e uma necessidade real. Atividades como pesquisas científicas, projetos de extensão e outras iniciativas acadêmicas requerem uma estrutura de gestão clara e bem definida. No entanto, muitas instituições ainda enfrentam desafios significativos na coordenação dessas atividades, o que pode comprometer a qualidade dos resultados e gerar retrabalho, atrasos e desmotivação entre os envolvidos.

Neste contexto, o desenvolvimento de um software de gestão de projetos acadêmicos surge como uma solução inovadora e necessária. Ao centralizar informações, automatizar processos e facilitar a colaboração entre estudantes, professores e orientadores, essa ferramenta tem o potencial de transformar a forma como os projetos são planejados e executados dentro das instituições acadêmicas.

Mais do que uma solução tecnológica, trata-se de uma proposta que visa agregar valor à jornada acadêmica, promovendo maior controle, produtividade e engajamento. Com isso, este trabalho propõe o desenvolvimento de um sistema intuitivo e funcional, capaz de atender às demandas específicas do ambiente acadêmicas e contribuir efetivamente para a melhoria dos processos acadêmicos.

1.1. Problemática

A gestão eficaz de projetos acadêmicos é um desafio presente em muitas instituições de ensino superior. Apesar da crescente demanda por organização e controle das atividades estudantis, a maioria dos professores e estudantes enfrentam dificuldades relacionadas ao acompanhamento de tarefas, cumprimento de prazos e comunicação entre os membros. Esses problemas podem resultar em atrasos na entrega, baixa qualidade dos trabalhos e aumento do estresse entre os envolvidos.

Além disso, as ferramentas disponíveis atualmente no mercado, em sua maioria, são genéricas e não atendem às necessidades específicas do ambiente acadêmico, como a integração entre professores, alunos e orientadores, nem facilitam o controle das múltiplas atividades simultâneas exigidas pelos cursos. Essa lacuna representa um obstáculo para o desenvolvimento acadêmico eficiente, limitando o potencial dos estudantes e prejudicando a produtividade institucional.

Diante desse cenário, surge a necessidade de uma solução tecnológica adaptada às particularidades do contexto universitário, capaz de centralizar informações, organizar

prazos e melhorar a comunicação entre os participantes dos projetos. Portanto, o objeto de estudo deste trabalho é a análise e o desenvolvimento de um software de gestão de projetos acadêmicos que atenda a essas demandas, contribuindo para a melhoria dos processos e resultados educacionais.

1.2. Objectivos do Trabalho

1.2.1. Objectivo Geral

Desenvolver um software de gestão de projetos acadêmicos com interface intuitiva, que atenda às necessidades dos docentes e estudantes, na organização e no acompanhamento de projectos e no cumprimento de prazos de entrega.

1.2.2. Objectivo Específico

- Analisar o contexto acadêmico para compreender as principais dificuldades na organização e gestão de projetos.
- Definir os requisitos do sistema com base nas necessidades de estudantes e professores.
- Projetar uma interface simples, prática e intuitiva, que facilite a usabilidade para diferentes perfis de usuários.
- Desenvolver as funcionalidades essenciais para o controle de tarefas, prazos, metas e comunicação entre os membros do projecto.
- Implementar o protótipo do sistema utilizando tecnologias acessíveis e adequadas ao ambiente educacional.

1.3. Metodologias de Pesquisa

A metodologia deste trabalho é de natureza aplicada, com abordagem qualitativa e foco no desenvolvimento experimental de software. O projeto será conduzido em cinco etapas principais: inicialmente, será realizado um levantamento de requisitos junto a estudantes e professores, por meio de entrevistas e questionário informal, para identificar as necessidades do sistema. Em seguida, será feita a modelagem do sistema com uso de diagramas UML, representando sua estrutura e fluxos. O desenvolvimento utilizará tecnologias web modernas e bancos de dados relacionais, priorizando uma interface responsiva e funcional. Após a implementação, serão realizados testes com usuários reais para validação e coleta de feedback. Por fim, todo o processo será documentado e os resultados analisados quanto à eficácia do sistema proposto.

1.4. Organização do Trabalho

O presente relatório está organizado em quatro capítulos, incluido o presente capítulo, o segundo capítulo consta os principais conceitos relacionados com o trabalho, propiamente conceitos relativos a desenvolvimento de software, base de dados e linguagens de programação e etc. O terceiro capítulo fala sobre as ferramentas e tecnologias utilizadas para o desenvolvimento do sistema, e o quarto capítulo, ultimo por sinal, contém as conclusões gerais do trabalho desenvolvido.

2. Conceitos Teóricos

Neste capítulo, serão abordados os principais conceitos teóricos que fundamentam o desenvolvimento deste trabalho. A compreensão aprofundada das bases teóricas é essencial para contextualizar a importância da gestão de projetos no ambiente acadêmico e para justificar as escolhas metodológicas e tecnológicas adotadas.

Serão explorados temas relacionados à gestão de projetos, suas práticas e ferramentas, além das particularidades do contexto educacional que impactam a organização e a execução dos trabalhos acadêmicos. Também serão discutidos conceitos sobre desenvolvimento de software e tecnologia educacional, que sustentam a proposta de criação de um sistema voltado para facilitar a administração de actividades acadêmicas.

Essa fundamentação teórica oferece o suporte necessário para o desenvolvimento do projeto, estabelecendo conexões entre o conhecimento acadêmico e a aplicação prática, e orientando as decisões ao longo de todo o processo.

2.1. Gestão de Projectos

A gestão de projetos consiste na aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto com o intuito de atender aos seus requisitos. [1]

No contexto acadêmico, a gestão de projetos pode ser aplicada a diversas atividades como trabalhos de conclusão de curso, pesquisas científicas, projetos interdisciplinares e extensões universitárias.

Segundo o Project Management Institute (PMI), um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. A gestão eficiente desses projetos envolve cinco grupos de processos: iniciação, planejamento, execução, monitoramento e encerramento. [1]

No meio acadêmico, é comum a ocorrência de falhas na organização e acompanhamento de projetos estudantis, seja por falta de planejamento ou ausência de ferramentas adequadas. Isso reforça a necessidade de soluções tecnológicas que possibilitem um controle mais eficaz de prazos, metas, atribuições de tarefas e avaliação de desempenho.

2.2. Gestão de Projectos Acadêmicos

A gestão de projetos acadêmicos possui particularidades que a diferenciam dos projetos corporativos. Envolve, muitas vezes, equipes multidisciplinares, orientação de professores, prazos institucionais e objetivos pedagógicos.

De acordo com Souza (2020), a utilização de metodologias de gerenciamento, como Scrum ou Kanban, adaptadas ao contexto educacional, pode aumentar significativamente o engajamento dos estudantes e a eficiência dos projetos. Além disso, o uso de sistemas informatizados contribui para a transparência e organização das atividades, favorecendo a comunicação entre os envolvidos. [2]

2.3. Desenvolvimento de Software

O desenvolvimento de software é o processo de concepção, especificação, programação, documentação, testes e manutenção de aplicativos e sistemas. Para que um sistema como o proposto seja eficaz, é necessário seguir boas práticas de engenharia de software, garantindo qualidade, usabilidade e escalabilidade. [3]

Conforme Pressman (2016), o ciclo de vida de desenvolvimento de software (SDLC) pode seguir modelos como o cascata, incremental, espiral ou ágil. No caso deste trabalho, optou-se pelo uso de abordagens ágeis, que oferecem maior flexibilidade e adaptabilidade ao longo do projeto. [3]

2.4. Tecnologia Educacional

A tecnologia educacional refere-se ao uso de ferramentas tecnológicas com o propósito de facilitar o ensino e a aprendizagem. Sistemas de gestão de projetos aplicados ao meio educacional se inserem nesse contexto, pois permitem uma administração mais efetiva das atividades pedagógicas, promovendo maior autonomia e responsabilidade por parte dos alunos. [4]

Moran (2018) afirma que a tecnologia, quando bem utilizada, favorece a personalização do aprendizado e o desenvolvimento de competências colaborativas. Um sistema de gerenciamento acadêmico pode ser um recurso essencial para acompanhar o progresso dos estudantes e fomentar a inovação pedagógica. [4]

2.5. Sistema de Informação

Um sistema de informação (SI) é um conjunto de componentes inter-relacionados que coleta, processa, armazena e distribui informações para apoiar a tomada de decisão. No ambiente acadêmico, os SI podem facilitar a gestão administrativa, pedagógica e de projetos.

O sistema proposto neste trabalho atua como um SI, ao integrar funcionalidades que permitem a docentes, estudantes e administradores realizar o acompanhamento e gestão de projetos em tempo real, com acesso a dados relevantes e relatórios automatizados. [5]

2.6. Tecnologia da Informação na Educação

A Tecnologia da Informação na Educação envolve o uso de sistemas digitais no ambiente universitário. A digitalização de processos educacionais traz vantagens como maior controle, acessibilidade e eficiência na gestão.

Kenski (2012) destaca que, além de modernizar a infraestrutura das instituições, a TI facilita a personalização da aprendizagem e permite a integração entre diferentes plataformas de ensino. Exemplos incluem o Moodle, amplamente usado para gestão de cursos online, e o SIGA, sistema de gestão acadêmica utilizado por universidades brasileiras. [5]

2.7. Desenvolvimento Web

O desenvolvimento web está diretamente ligado à construção de sistemas acessíveis e funcionais na internet. Ele se baseia na arquitetura cliente-servidor, onde o frontend (lado do cliente) interage com o usuário e o backend (lado do servidor) processa os dados e lógica do sistema.

Tecnologias como PHP, MySQL, JavaScript e frameworks como Bootstrap são amplamente utilizadas para o desenvolvimento de sistemas acadêmicos. Segundo Nixon (2018), a combinação dessas tecnologias permite criar aplicações robustas, dinâmicas e responsivas. [6]

2.8. Banco de Dados Relacional

Um banco de dados relacional organiza informações em tabelas interligadas, utilizando a linguagem SQL para gerenciamento. A integridade e a consistência dos dados são mantidas por meio de chaves primárias e estrangeiras. [6]

O MySQL, por exemplo, é uma das ferramentas mais populares e se adapta bem a sistemas acadêmicos que necessitam de controle estruturado sobre projetos, alunos, orientadores e atividades correlatas. [6]

2.9. Usabilidade e Experiência do Usuário

A usabilidade é um fator crítico no sucesso de um sistema, especialmente em ambientes acadêmicos, onde os usuários têm diferentes níveis de familiaridade com tecnologia. Princípios de design centrado no usuário foram considerados para tornar a navegação mais intuitiva e acessível (Nielsen, 1994). [7]

2.10. Segurança da Informação

A proteção de dados sensíveis, como notas, documentos e informações pessoais, exige que o sistema siga boas práticas de segurança da informação, como criptografia, controle de acesso e backups regulares. A segurança de uma determinada informação pode ser afectada por factores comportamentais e uso de quem se utiliza dela, pelo ambiente computacional que a cerca ou por pessoas mal-intencionadas que têm o objectivo de furtar, destruir ou modificar tal informação. [8]

Garantir a protecção da informação e mitigar os riscos, não são somente desafios técnicos, de responsabilidade de analistas e engenheiros, mas desafios que envolvem a gestão como um todo. Em um sistema de software, este requisito não funcional caracteriza a segurança de que acessos não autorizados ao sistema e dados associados não serão permitidos. Portanto, é assegurada a integridade do sistema quanto a ataques intencionais ou acidentes. Dessa forma, a segurança é vista como a probabilidade de que a ameaça de algum tipo será repelida. [8]

Adicionalmente, à medida que os sistemas de software tornam-se distribuídos e conectados a redes externas, os requisitos de segurança vão se tornando cada vez mais importantes. [8]

Os princípios básicos da segurança da informação que atualmente norteiam na análise, panejamento e na implementação da segurança dos dados, são:

Confidencialidade: consiste em garantir que apenas as pessoas que estão autorizadas a ter acesso à informação são capazes de fazê-lo; [10]

Integridade: garantia de que as informações serão protegidas contra alterações não autorizadas e mantidas a exactidão das mesmas, tal qual como foi armazenada e disponibilizada. [10]

Disponibilidade: consiste em assegurar que os sistemas de informação e a informação estarão disponíveis e operacionais quando necessários, apoiando, assim, os processos de negócios. [10]

Autenticidade: garantia de que a informação é verdadeira, de fonte segura e não sofreu alterações em seu percurso; [10]

Não repúdio: garantia de que o emissor de algum dado ou informação não possa, posteriormente, negar que tenha enviado e/ou alterado alguma informação. [10]

Os benefícios evidentes são reduzir os riscos com vazamentos, fraudes, erros, uso indevido, sabotagens, roubo de informações e diversos outros problemas que possam comprometer estes princípios básicos. [8]

A segurança visa também aumentar a produtividade dos usuários através de um alinhamento estratégico dos controles e recursos de segurança aos requerimentos do

negócio, provendo um ambiente mais organizado, onde exista um maior controle sobre os recursos de informática e finalmente, viabilizando aplicações críticas da empresa. [8]

Uma política de segurança bem elaborada (com base no negócio), adequada gestão de risco, a implantação de uma arquitectura de sistemas segura, complementadas com o apoio dos diversos níveis da organização, são fundamentais para garantir uma protecção eficiente para organização. [8]

2.11. Plataformas Web Responsivas

Com o aumento do uso de dispositivos móveis, é essencial que sistemas web sejam responsivos e se adaptem a diferentes tamanhos de tela. Frameworks como Bootstrap e Tailwind CSS são comumente utilizados para esse fim. [9]

2.12. Linguagem de Programação

Uma linguagem de programação é um conjunto de símbolos, sintáticas, regras semânticas e palavras-chave que permite criar códigos com instruções para controlar as acções de uma máquina. Ou seja, ela é uma forma de comunicação entre ser humano e computador. [11]

2.12.1. Tipos de Linguagem de Programação

Hoje, existem diversos tipos de linguagens de programação, as quais são escritas pelos programadores, algumas dessas linguagens são compreendidas pelo computador e outras ajudam na forma de tradutores, com destaque para: JavaScript, Java, Ruby, Python, C#, C, Swift, PHP.

2.12.2. Classificação de Linguagens de Programação

- As linguagens podem ser classificadas em de três tipos:
- Linguagem de máquina;
- Linguagens assembly;
- Linguagens de alto nível;

2.12.3. Linguagem de Máquina

É uma linguagem "crua", ou seja, não muda seu estado natural. Essa linguagem é formada de string de números, definindo a realização das operações em um computador, sendo realizado uma tarefa de cada vez. [11]

2.12.4. Linguagem Assembly

Essa linguagem consiste de abreviações de expressões em inglês que são operações elementares, onde se originou a base da linguagem Assembly. Os assemblers como conhecidos são programas tradutores que convertem os primeiros programas de linguagem assembly em linguagem de máquina a velocidade do computador. Embora o código seja mais claro para seres humanos, ele é incompreensível para computadores até ser traduzido em linguagem de máquina. [11]

2.12.5. Linguagem de Alto Nível

São instruções únicas que podem ser escritas para realizar tarefas substanciais. Os programas tradutores são conhecidos também pelo nome de compiladores - convertem os programas de linguagem em alto nível em linguagem de máquina. Esse tipo de linguagem permite aos programas escrever instruções que se pareçam com o inglês e contêm notações matemáticas comumente utilizadas. [11]

Segue-se abaixo o exemplo de linguagens alto nível:JavaScript, Java, Ruby, Python, C#, C, Swift, PHP.

Em relação às vantagens das linguagens de alto nível, podemos dizer que a principal é a facilidade de aprendizagem, mas esse não é o único ponto. Outro grande benefício é a produtividade que a pessoa programadora ganha ao usar esse tipo de linguagem. Iso porque a abstracção proporcionada pela linguagem de alto nível permite que você escreva mais códigos em menos tempo, o que é extremamente positivo quando se trabalha em sistemas complexos. Além disso, caso outra pessoa precise verificar o seu código, ela conseguirá entender com mais facilidade o que foi feito. Dessa forma, o processo de manutenção também 6 simplificado. [11]

Por outro lado, as linguagens de alto nível apresentam algumas desvantagens. Existem casos, por exemplo, em que o desempenho de um programa pode ser prejudicado, pois eles exigem maior tempo de processamento. Normalmente, também ocupam mais memória quando comparadas a uma linguagem de baixo nível. Ao contrário das linguagens de alto nível, as de baixo nível são voltadas para o entendimento da máquina. Por isso, elas têm uma sintaxe mais complexa e não contam com comandos tão intuitivos. Isso porque a linguagem da máquina é constituída apenas por sequências de 0 e 1, o chamado código binário. Então, as linguagens de baixo nível têm instruções mais diretas para o processador por isso, são mais próximas da linguagem da máquina. Sendo assim, para utilizá-las é preciso conhecer não só a linguagem, mas também o hardware do dispositivo com o qual irá trabalhar. No mais, a principal linguagem de baixo nível que temos é o Assembly. [11]

2.12.6. PHP

O PHP (Hypertext Preprocessor) é uma linguagem de programação interpretada, de código aberto e amplamente utilizada para o desenvolvimento web. Ele foi criado em 1994 por Rasmus Lerdorf e, desde então, evoluiu para se tornar uma das tecnologias de servidor mais populares do mundo.

Segundo Nixon (2018), o PHP é especialmente projetado para a criação de páginas dinâmicas e interativas, sendo executado no servidor e gerando HTML como saída para os navegadores dos usuários. [12]

2.12.7. Funcionamento

- Interpretação no Servidor: Quando um cliente solicita uma página php, o servidor web (como Apache ou Nginx) passa o arquivo ao interpretador PHP. [13]

- Execução: O interpretador PHP processa o código, realizando conexões com banco de dados, lógicas de programação, validações, entre outras tarefas. [13]

- Geração de HTML: O resultado desse processamento é transformado em HTML puro, que é então enviado ao navegador do usuário. [13]

- Integração com Banco de Dados: Um dos principais usos do PHP é o acesso a bancos de dados como MySQL, permitindo criação, leitura, atualização e exclusão de dados (CRUD). [13]

- Esse funcionamento facilita a construção de aplicações dinâmicas, como sistemas de login, portais educacionais, plataformas de e-commerce e sistemas de gestão acadêmica como o SGPA (Sistema de Gestão de Projetos Acadêmicos).

2.12.8. Características do PHP

Linguagem do lado do servidor: PHP é uma linguagem que é executada no servidor (server-side), o que significa que o código é processado no servidor web antes de ser enviado ao navegador do usuário. [12]

Código aberto (Open Source): Uma das principais características do PHP é que ele é gratuito para uso e distribuição, com seu código-fonte disponível para desenvolvedores em todo o mundo. [12]

Fácil de aprender: A sintaxe do PHP é relativamente simples e similar à de linguagens como C e Java, tornando-o acessível para iniciantes e rápido de dominar. [12]

Ampla compatibilidade: O PHP é compatível com diversos servidores (Apache, Nginx, IIS) e sistemas operacionais (Linux, Windows, macOS), além de suportar uma grande variedade de bancos de dados, especialmente o MySQL. [12]

Integração com HTML: O PHP pode ser facilmente embutido em documentos HTML, o que facilita o desenvolvimento de aplicações web híbridas. [12]

Suporte a Programação Orientada a Objetos (POO): Desde a versão 5, o PHP passou a oferecer suporte robusto a POO, permitindo a criação de sistemas modulares, reutilizáveis e mais organizados (Nixon, 2018).

Comunidade ativa e vasta documentação: Com milhões de desenvolvedores ao redor do mundo, o PHP conta com uma comunidade ativa e inúmeros recursos, fóruns, bibliotecas e frameworks (como Laravel, CodeIgniter e Symfony). [12]

2.12.9. Vantegens do PHP

Desempenho optimizado: Apesar de ser interpretado, o PHP tem desempenho eficiente, especialmente com as versões mais recentes (PHP 7 e 8), que trouxeram melhorias significativas de performance.

Escalabilidade: O PHP é escalável e pode ser utilizado desde sites simples até aplicações web complexas e de grande tráfego, como o Facebook, Wikipedia e WordPress. [12]

Amplo suporte de frameworks: Frameworks PHP como Laravel e Symfony ajudam a acelerar o desenvolvimento, manter a organização do código e seguir boas práticas como MVC (Model-View-Controller). [13]

Integração com bancos de dados: O PHP possui suporte nativo para diversos sistemas de gerenciamento de banco de dados, como MySQL, PostgreSQL, SQLite, Oracle e outros. [13]

Segurança: Embora a segurança dependa da forma como o código é escrito, o PHP oferece recursos como filtros de entrada, hashing de senhas (com password_hash()), sessões seguras e proteção contra SQL Injection com PDO (PHP Data Objects). [13]

Alta empregabilidade: Por ser uma linguagem tão difundida, o domínio do PHP garante boas oportunidades no mercado de trabalho, especialmente em desenvolvimento web.

2.13. Banco de Dados

É uma colecção de dados inter-relacionados, representando informações sobre um domínio específico", ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados. [14]

Dados é a informação apresentada em qualquer forma que seja acordada entre partes que criam e usam os dados. [14]

2.13.1. Modelo Conceitual

Uma das técnicas mais utilizadas dentre os profissionais da área é a abordagem entidade-relacionamento (ER), onde o modelo é representado graficamente através do diagrama entidade-relacionamento (DER). [15]

2.13.2. Modelo Lógico

Descreve o BD no nível do SGBD, ou seja, depende do tipo particular de SGBD que será usado. Não podemos confundir com o Software que será usado. O tipo de SGBD que o modelo lógico trata é se o mesmo é relacional, orientado a objectos, hierárquico, etc. [15]

Quando se inicia o desenvolvimento de um novo sistema, ou mesmo de uma nova funcionalidade para um sistema existente, um dos primeiros passos a ser executado é o estudo e levantamento dos requisitos necessários para a construção do produto final. [18]

Durante essa análise, identificam-se as principais partes e objectos envolvidos, suas possíveis acções e responsabilidades, suas características e como elas interagem entre si.

A partir das informações obtidas, pode-se desenvolver um modelo conceitual que será utilizado para orientar o desenvolvimento propriamente dito, fornecendo informações sobre os aspectos relacionados ao domínio do projecto em questão. [18]

2.13.3. Modelo Hierárquico

O modelo hierárquico organiza dados em uma estrutura do tipo árvore, onde cada registo tem um único "pai" ou raiz. Registos "irmãos" são classificados em uma ordem especifica. [15]

Esa ordem é usada como a ordem fisica para armazenar o banco de dados: Este modelo é bom para descrever muitas relações do mundo real. [15]

2.13.4. Modelo Entidade Relacionamento

Este modelo capta as relações entre entidades do mundo rea, to toma parcia com o Este mode rede, mas não está directamente ligado à cstritura fisica do banco de tarta Em vez disso, ele é frequentemente usado para projectar um banco do deta conceitualmente. [15]

Aqui, pessoas, lugares e coisas sobre quais pontos de dados são armazenados e referidos como entidades, cada uma das quais possui certos atributos que, em conjunto, compõen seu domínio. A cardinalidade, ou relações entre entidades, também são mapeadas. [15]

2.13.5. Modelo de banco de dados orientado para objectos

Este modelo define o banco de dados como uma colecção de objectos, ou elementos de soware reutilizáveis, com recursos e métodos associados. Há vários tipos de bancos de dados orientados para objectos: [16]

Um banco de dados multimídia incorpora mídia, como imagens, que não podem ser armazenadas em um banco de dados relacional. [16]

Um banco de dados de hipertexto permite que qualquer objecto seja vinculado a qualquer outro objecto. É útil para organizar lotes de dados diferentes, mas não é ideal para a análise numérica. [16]

O modelo de banco de dados orientado a objectos é o modelo de banco de dados pós-relacional mais conhecido, uma vez que ele incorpora tabelas, mas não se limita a elas. Tais modelos também são conhecidos como modelos de bancos de dados híbridos. [16]

2.13.6. Diagrama Entidade Relacionamento

Enquanto o MER é um modelo conceitual, o Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinónimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstracto demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras. [16]

O diagrama facilita ainda a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado. [16]

2.13.7. Aplicação de Banco de Dados

As aplicações de bancos de dados podem ser classificadas em três categorias: Orientadas à transações; de suporte à decisão; e para a Internet. A descrição dos dados e o tipo de dados diferem para cada categoria. Na categoria orientada à transacção, as transações são curtas como, por exemplo: débito e crédito. No ambiente Internet as transacções são mais longas em função da manipulação de diferentes tipos de dados, incluindo objectos multimídia. O perfil dos profissionais muda em função das características singulares de cada categoria. A disponibilidade das aplicações varia com a situação de negócio da empresa, entretanto as aplicações para a Internet devem operar 24 horas por dia nos 7 dias da semana. O ambiente de desenvolvimento de aplicações muda para cada categoria, sendo recomendado o uso de SQL com opção nativa para orientação por objecto na Internet. [16]

A característica das aplicações orientadas à transações é a rigidez da especificação. No lado usuário existe uma tela que deve ser preenchida para a obtenção de resultados bem definidos. São aplicativos para suporte a operação da empresa. As transações são pequenas e o tempo de resposta deve ser o menor possível para aumentar a produtividade do pessoal operacional. As estruturas de SQL são normalmente pré-definidas levando-se em consideração os resultados esperados e a melhor performance de acesso físico. Nesta situação o usuário não necessita possuir experiência com processamento de dados. Na área de desenvolvimento são envolvidos os programadores e o DBA (Database Administrator). [16]

Para suporte à decisão o modelo de dados deve representar o negócio da empresa sem levar em conta os aspectos de desempenho do banco de dados. Os dados seleccionados para carregar esses bancos de dados devem ser para atender aos data warehouses e datamarts. Os usuários finais devem possuir conhecimentos de como relacionar os dados de um dicionário de dados e o DBA deve estar familiarizado com os conceitos de data warehouse. Nesses casos as queries são montadas de acordo com os requerimentos de negócios imediatos. O uso de queries pré-definidas requererá um esforço muito grande para a obtenção dos resultados. [16]

2.13.8. Segurança de Banco de Dados

A principal característica de um sistema é controlar os processos de uma empresa. Dessa forma, cada solução que encontramos hoje no mercado de tecnologia possui

características com objectivo de proporcionar aos clientes uma qualidade considerável em requisitos de segurança, performance, escalabilidade e, acima de tudo, coerência no uso da informação. [16]

Esta coerência se trata de garantir que uma informação será verdadeira, será confiável e integra. [16]

Quando um sistema controla os dados de uma organização, estes dados devem ser cuidadosamente analisados, afinal eles estarão de alguma forma interligados entre si no que diz respeito ao processo do negócio como um todo. [16]

Por exemplo, um cadastro de fornecedores estará de alguma forma se comunicando com o cadastro de produtos, afinal os produtos pertencem a um fornecedor. Assim, a integridade de uma entidade pode ter impacto directamente em outra entidade.

Normalmente cada funcionário tem um cargo e uma função; operar um sistema faz parte do dia a dia deste funcionário. Quando existe mais de um funcionário com a mesma função, eles utilizam o mesmo sistema e realizam os mesmos processos.

Isso pode parecer simples, porém se um sistema não tiver regras de integridade de dados, a forma de inserir os dados neste sistema ocorrerá de forma desordenada, causando um grande problema de registos sem uma regra definida ou mesmo inseridos de forma incorrecta. [16]

2.13.9. Sistema de Gerenciamento de Banco de Dados Mysql

O MySQL é um sistema de gerenciamento de banco de dados relacional (SGBD) gratuito, de código aberto, e atualmente é mantido pela Oracle, que utiliza a linguagem SQL.[14]

Atualmente, o MySQL é um dos bancos de dados mais utilizados do mundo, podendo ser encontrado em projetos dos mais variados tamanhos. Conta com uma grande comunidade de usuários ativos, o que facilita a busca por soluções de problemas e dicas de uso. O MySQL é um banco de dados leve, de fácil instalação e utilização. Caso você conheça outro SGBD não terá dificuldades em iniciar seus estudos com o MySQL, pois utiliza a linguagem SQL como interface de comunicação. [14]

Através das ferramentas de linha de comando é possível utilizar qualquer recurso do MySQL, porém, se preferir, é possível utilizar o MySQL Workbench, ferramenta gráfica feita para o MySQL. [14]

- É um SGBD gratuito e bastante popular.
- Utiliza a linguagem SQL.

Possui uma ferramenta com uma interface gráfica que facilita a interação com o banco de dados. [14]

Por último, temos de conceituar um sistema de banco de dados como o conjunto de quatro componentes básicos: dados, hardware, software e usuários. Date conceituou que

"sistema de bancos de dados pode ser considerado como uma sala de arquivos

electrónicos". [14]

2.14. Paginas Web

Uma página web é, nada mais do que uma página solitária de um site. Cada página web tem um URL único, que pode aceder, mas, ao contrário do que acontece com os websites, não a pode navegar. [17]

Uma aplicação Web pode conter uma colecção de páginas, porém o conteúdo destas páginas é montado dinamicamente, ou seja, é carregado através de solicitações (requisições) à um banco de dados, que conterá armazenado os textos e indicação dos caminhos das imagens ou Mídias que a página precisa exibir. Porém um HTML não tem sosso direito à um banco de dados, e esta comunicação deve ser feita por uma linguagem de programação de sentidor Web, Esta aplicação escrita com uma linguagem de servidor que tem o poder de acessar o banco de dados e montar a página HTML conforme o solicitado pelo navegador. Estas solicitações podem ser feitas de várias manciras, inclusive utilizando JavaScript. Portanto uma aplicação Web é mais complexa porque precisa de uma linguagem de servidor para poder intermediar as solicitações do navegador, um banco de dados, e muitas vezes (porém não obrigatoriamente) exibit páginas HTML com estes conteúdos. [17]

2.14.1. Tipos de Páginas Web

Existem basicamente dois tipos de páginas web, estática e dinâmica.

- Página Web estática: Apesar de existirem cada vez menos, as páginas estáticas caracterizam-se por ter sempre o mesmo conteúdo e tem normalmente a linguagem HTML [17]

- Página Web dinâmica: Por outro lado, uma página web dinâmica permite que os seus conteúdos estejam em constante mudança, pois provêm de diferentes fontes.

2.14.2. HTML

A Linguagem de Marcação de Hipertexto (HTML) é uma linguagem de computador que compõe a maior parte das páginas da internet e dos aplicativos online. Um hipertexto é um texto usado para fazer referência a outros textos, enquanto uma linguagem de

marcação é composta por uma série de marcações que dizem para os servidores da web qual é o estilo e a estrutura de um documento. [17]

O HTML é baseado no conceito de Hipertexto. Hipertexto são conjuntos de elementos - ou nós - ligados por conexões. Estes elementos podem ser palavras, imagens, vídeos, áudio, documentos etc. Estes elementos conectados formam uma grande rede de informação. Eles não estão conectados linearmente como se fossem textos de um livro, onde um assunto é ligado ao outro seguidamente. A conexão feita em um hipertexto é algo imprevisto que permite a comunicação de dados, organizando conhecimentos e guardando informações relacionadas. [17]

O HTML não é considerado uma linguagem de programação, já que ele não pode criar funcionalidades dinâmicas. Ao invés disso, com o HTML, os usuários podem criar e estruturar seções, parágrafos e links usando elementos, tags e atributos. [17]

2.14.3. HTML 5

O HTMLS é a nova versão do HTML4. Enquanto o WHATWG define as regras de marcação que usaremos no HTMLS e no XHTML, eles também definem APIs que formarão a base da arquitectura web. Essas APIs são conhecidas como DOM Level 0. Um dos principais objectivos do HTML5 é facilitar a manipulação do elemento possibilitando o desenvolvedor a modificar as características dos objectos de forma não intrusiva e de mancira que seja transparente para o usuario final. Ao contrario das versoes anteriores, o HTMLS fornece ferramentas para a CSS e o Javascript fazerem seu trabalho da melhor maneira possível. O HTMLS permite por meio de suas APIs a manipulação das características destes elementos, de forma que o website ou a aplicação continue leve e funcional. O HTMLS também cria novas tags e modifica a função de outras. As versões antigas do HTML não continham um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, sidebar, menus e etc. Não havia um padrão de nomenclatura de IDs, Classes ou tags. Não havia um método de capturar de maneira automática as informações localizadas nos rodapés dos websites. Há outros elementos e atributos que sua função e significado foram modificados e que agora podem ser reutilizados de forma mais eficaz. Por exemplo, elementos como B ou I que foram descontinuados em versões anteriores do HTML agora assumem funções diferentes e entregam mais significado para os usuários. O HTML5 modifica a forma de como escrevemos código e organizamos a informação na página. Seria mais semântica com menos código. Seria mais interactividade sem a necessidade de instalação de plugins e perda de performance. É a criação de código interoperável, pronto para futuros dispositivos e que facilita a reutilização da informação de diversas formas. O WHATWG tem mantido o foco para manter a retrocompatibilidade. Nenhum site deverá ter de ser refeito totalmente para se adequar aos novos conceitos e regras. O HTMLS está sendo criado para que seja compatível com os browsers recentes, possibilitando a utilização das novas características imediatamente. [17]

2.14.4. CSS

CSS é a sigla para o termo em inglês Cascading Style Sheets que, traduzido para o português, significa Folha de Estilo em Cascatas. O CSS é fácil de aprender e entender e é facilmente utilizado com as linguagens de marcação HTML ou XHTML. 21] CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

CSS foi desenvolvido pelo W3C (World Wide Web Consortium) em 1996, por uma razão bem simples. O HTML não foi projectado para ter tags que ajudariam a formatar a página. Você deveria apenas escrever a marcação para o site. Tags como < font› foram introduzidas na versão 3.2 do HTML e causaram muitos problemas para os desenvolvedores. Como os sites tinham diferentes fontes, cores e estilos, era um processo longo, doloroso e caro para reescrever o código. Assim, o CSS foi criado pelo W3C para resolver este problema. [17]

2.14.5. Bootstrap

Bootstrap é um framework front-end de código aberto originalmente desenvolvido por Mark Otto e Jacob Thornton no Twitter. Lançado em 2011, o Bootstrap tem como objetivo facilitar o desenvolvimento de interfaces web responsivas e modernas, padronizando componentes e layouts com base em HTML, CSS e JavaScript. [29]

2.14.5.1. Caracteristicas Príncipais

Responsividade: Utiliza um sistema de grid baseado em 12 colunas que se adapta automaticamente a diferentes tamanhos de tela, proporcionando compatibilidade com dispositivos móveis (smartphones, tablets, desktops etc.). [29]

Componentes pré-definidos: Inclui diversos elementos de interface prontos para uso, como botões, formulários, tabelas, cards, barras de navegação, alertas, modais, carrosséis e muito mais. [29]

Personalização: Permite personalizar variáveis do Sass para ajustar o estilo de acordo com a identidade visual do projeto. [29]

Compatibilidade entre navegadores: Garante que a interface tenha aparência consistente em diferentes navegadores modernos (Chrome, Firefox, Safari, Edge, etc.). [29]

Extensibilidade: Por ser open source e modular, Bootstrap pode ser estendido com plugins JavaScript ou combinado com outras bibliotecas como jQuery ou frameworks como Angular e React. [29]

2.14.5.2. Funcionamento

O funcionamento do Bootstrap é baseado em três pilares:

- HTML – estrutura da página e marcação dos elementos com classes específicas do framework. [29]
- CSS (ou Sass) – aplicação de estilos pré-definidos a partir das classes utilizadas no HTML. [29]
- JavaScript – fornece interatividade aos componentes (por exemplo, modais, carrosséis, tooltips), geralmente por meio de plugins nativos ou jQuery. [29]

O Bootstrap utiliza um sistema de grid responsivo, com classes como container, row, col, e variantes como col-sm-, col-md-, col-lg-, para organizar os elementos da página com flexibilidade. A integração é simples: o desenvolvedor inclui os arquivos CSS e JS do Bootstrap (ou usa um CDN), e pode rapidamente montar interfaces sem escrever estilos do zero. [29]

2.14.5.3. Vantagens do Bootstrap

Produtividade: Reduz significativamente o tempo de desenvolvimento de interfaces.

Padronização visual: Mantém consistência visual entre diferentes páginas e aplicações.

Facilidade de aprendizado: Desenvolvedores com conhecimentos básicos de HTML e CSS conseguem utilizá-lo rapidamente.

Comunidade ativa: Possui ampla documentação e suporte da comunidade, com fóruns, tutoriais e exemplos prontos.

Integração fácil com outras tecnologias: Pode ser utilizado com backends em PHP, Python, Java, etc. [27]

Uso no contexto do desenvolvimento do Sistema de Gestão de Projetos Acadêmicos (SGPA), o uso do Bootstrap proporciona uma interface limpa, intuitiva e responsiva, fundamental para a usabilidade do sistema por estudantes, docentes e administradores. O sistema pode ser acessado de forma eficiente tanto em computadores quanto em dispositivos móveis, garantindo acessibilidade e praticidade.

2.15. Desenvolvimento de Software

Existem diversos processos de desenvolvimento de software, no entanto há algumas atividades básicas comuns à grande parte dos processos existentes. [19]

2.15.1. Levantamento de Requisitos

Esta actividade tem como objectivo, compreender o problema, dando aos desenvolvedores e usuários, a mesma visão do que deve ser construído para resolução do problema. Desenvolvedores e clientes, em conjunto, buscam levantar e priorizar as necessidades dos futuros usuários do software (necessidades essas denominadas como requisitos). [19]

O Levantamento de Requisitos é a etapa mais importante, no que diz respeito ao retorno de investimentos no projecto. Vários projectos são abandonados pelo baixo levantamento de requisitos, ou seja, membros da equipe não disponibilizaram tempo suficiente para essa fase do projecto, em compreender as necessidades dos clientes em relação ao sistema a ser desenvolvido. [19]

E como um sistema de informações geralmente é utilizado para automatizar processos de negócio em uma organização, esses processos da organização devem ser bem compreendidos para que o restante das actividades do processo de desenvolvimento flua de acordo com as reais necessidades do cliente. [19]

2.15.2. Requisitos Funcionais

Um requisito de sistema de software que especifica uma função que o sistema ou componente deve ser capaz de realizar. Estes são requisitos de software que definem o comportamento do sistema, ou seja, o processo ou transformação que componentes de software ou hardware efectuam sobre as entradas para gerar as saídas. Esses requisitos capturam as funcionalidades sob o ponto de vista do usuário. [19]

2.15.3. Requisitos Não Funcionais

Em engenharia de sistemas de software, um requisito não funcional de software é aquele que descreve não o que o sistema fará, mas como ele fará. Assim, por exemplo, têm-se requisitos de desempenho, requisitos da interface externa do sistema, restrições de projecto e atributos da qualidade. A avaliação dos requisitos não funcionais é feita, em parte, por meio de testes, enquanto que outra parte é avaliada de maneira subjectiva. Pereba que tanto os requistos funcionais quanto os nao funcionais possuem

importanciano desenvolvimento de urn sistema de sofivare Entetanto, o requistios não funcionais, também denominados de atributos de qualidade, tem um papel relevante durate o desenvolvimento de um sistema, actuando como critérios na seleção elou composição de uma arquitectura de software, dentre as virias allemativas de projecto. Cabe salientar que à medida que os sistemas se tomam maiores e mais complexos, o suporte a requisitos não funcionais depende cada vez mais de decisões tomadas no projecto da arquitectura de software. Trata-se de uma visão compartilhada pelos profissionais da área e, especificamente, pela comunidade de arquitectura de software. [19]

Requisitos não funcionais são aqueles que não estão diretamente relacionados à funcionalidade de um sistema. O termo requisitos não funcionais é também chamado de atributos de qualidade. Os requisitos não funcionais têm um papel de suma importância durante o desenvolvimento de um sistema, podendo ser usados como critérios de selecção na escolha de alternativas de projecto, estilo arquitectural e forma de implementação. [19]

Desconsiderar ou não considerar adequadamente tais requisitos é dispendioso, pois torna difícil a correcção uma vez que o sistema tenha sido implementado. Suponha, por exemplo, que uma decisão tenha sido feita de modularizar a arquitectura de um sistema de modo a facilitar sua manutenção e adição de novas funcionalidades. Entretanto, modularizar um sistema adicionando uma camada a mais pode comprometer um outro requisito, o de desempenho. Portanto, faz-se necessário definir logo cedo quais requisitos não funcionais serão priorizados na definição de uma arquitectura. [19]

Os requisitos não funcionais abordam aspectos de qualidade importantes em sistemas de software. Se tais requisitos não são levados em consideração, então o sistema de software poderá ser inconsistente e de baixa qualidade, conforme discutido acima. Para tanto, quanto mais cedo forem definidos os critérios arquitecturais, mais cedo o projectista pode renifica o estilo, ou combinação de estilos, mais apropriado ao sistema considerado. [19]

2.15.4. Análise de Requisitos

Esta etapa, também chamada de especificação de requisitos, é onde os desenvolvedores fazem um estudo detalhado dos dados levantados na actividade anterior. De onde são construídos modelos a fim de representar o sistema de software a ser desenvolvido. [20]

O interesse nessa actividade é criar uma estratégia de solução, sem se preocupar como essa estratégia será realizada, ou seja, utilizar as necessidades dos clientes, depois de compreendido o problema, para resolução do problema solicitado. Assim é necessário definir o que o sistema deve fazer, antes de definir como o sistema irá fazer. [20]

O que acontece com frequência, é quando as equipes de desenvolvimento partem para a solução do problema do software, sem antes ter definido completamente o problema em questão. Nesta fase deve-se então realizar a validação e verificação dos modelos construídos, antes de partir para solução do problema:

- Validação: tem por objetivo, assegurar que o sistema de software está atendendo às reais necessidades do cliente;
- Verificação: verifica se os modelos construídos na análise estão em conformidade com os requisitos do cliente. [20]

2.15.5. Projecto

Nesta fase é que deve ser considerado, como o sistema funcionará internamente, para que os requisitos do cliente possam ser atendidos. Alguns aspectos devem ser considerados nessa fase de projecto do sistema, como: arquitectura do sistema, linguagem de programação utilizada, Sistema Gerenciador de Banco de Dados (SGBD) utilizado, padrão de interface gráfica, entre outros. [20]

No projecto é gerada uma descrição computacional, mencionando o que o software deve fazer, e deve ser coerente com a descrição realizada na fase de análise de requisitos. O projecto possui duas actividades básicas: projecto da arquitectura (ou projecto de alto nível), e projecto detalhado (ou projecto de baixo nível). [20]

Em um processo de desenvolvimento orientado a objectos, o projecto da arquitectura normalmente é realizado por um arquitecto de software. O projecto da arquitectura visa distribuir as classes de objectos relacionados do sistema em subsistemas e seus componentes, distribuindo também esses componentes pelos recursos de hardware disponíveis. [20]

Já no projecto detalhado, são modeladas as relações de cada módulo com o objectivo de realizar as funcionalidades do módulo. Além de desenvolver o projecto de interface com o usuário e o projecto de banco de dados. [20]

2.15.6. Implementação

Nessa etapa, o sistema é codificado a partir da descrição computacional da fase de projecto em uma outra linguagem, onde se torna possível a compilação e geração do código-executável para o desenvolvimento software. Em um processo de desenvolvimento orientado a objectos, a implementação se da, definindo as classes de objectos do sistema em questão, fazendo uso de linguagens de programação como, por exemplo: Delphi (Object Pascal), C++, Java, etc. Pode-se também utilizar na implementação ferramentas de software e bibliotecas de classes preexistentes para agilizar a actividade, como também o uso de ferramentas CASE, dinamizzam o processo de desenvolvimento, nas várias atividades, onde inclui-se geração de código-fonte, documentação, etc. [20]

2.15.7. Testes

Diversas actividades de testes são executadas a fim de se validar o produto de software, testando cada funcionalidade de cada módulo, buscando, levando em consideração a especificação feita na fase de projecto. Onde o principal resultado é o relatório de testes, que contém as informações relevantes sobre etros encontrados no sistema, e seu comportamento em vários aspectos. Ao final dessa actividade, os diversos módulos do sistema são integrados, resultando no produto de software. [20]

2.15.8. Implantação

Por fim a implantação compreende a instalação do software no ambiente do usuário. O que inclui os manuais do sistema, importação dos dados para o novo sistema e treinamento dos usuários para o uso correcto e adequado do sistema. Em alguns casos quando da existência de um software anterior, também é realizada a migração de dados anteriores desse software. [20]

2.16. Diagramas UML

Os desenvolvedores criam diagramas UMi para entender projectos, arquitetura de código e propostas de implementação de sistemas de software complexos. Um diagrama UML é uma forma de visualizar sistemas e sofiares usando a Linguagem de Modelagem Unificada (Unified Modeling Language). Os diagramas UML também são usados para modelar workflows e processos de negócios. Desenvolver códigos para softwares pode ser um processo complicado, pois possui muitos elementos conectados. São milhares de linhas de linguagem de programação que podem ser difíceis de entender olhando rapidamente. Os diagramas UML ajudam a manter o controlo das relações e hierarquias entre linhas importantes de código. Embora estes diagramas sejam semelhantes a

árvores de decisão ou fluxogramas, têm atributos únicos. A simplificação de atributos complexos é extremamente útil para engenheiros e stakeholders de outras áreas. Essa padronização permite que eles permaneçam por dentro dos projectos que se trabalha e evitam se perder nas infinitas complexidades inerentes à programação de software. Os diagramas UML também dividem os componentes e subcomponentes que são essenciais para a construção de um softwar. O diagrama UML simplifica estas informações ao uma criar referência visual que é fácil de compreender rapidamente. Também possui elementos e estruturas patronizadas, que seguem o mesmo estilo e formato, o que possibilita uma comprensão rapida de quem já conhece a linguagem UML. [32]

Há duas subcategorias de diagramas UML, conhecidas como:

- Diagramas Estructurais: Descrevem os componentes que fazem parte de um sistema e a relação existente entre estas, representando aspectos estáticos;
- Diagramas Comportamentais: Representam o que acontece dentro de um sistema. Mostram como os componentes interagem uns com os outros.

2.17. Ferramentas e Tecnologias

2.17.1. Visual Studio Code

O VS Code é um editor de código-fonte leve e poderoso, com suporte para diversas linguagens de programação. Durante o desenvolvimento do sistema, foi utilizado como editor principal para os arquivos PHP, JavaScript, HTML e CSS. Suas extensões e suporte ao controle de versão com Git também foram vantajosos para a organização do projeto. [36]

2.17.2. Sublime Text

O Sublime Text foi utilizado como editor complementar ao VS Code, principalmente em tarefas rápidas de edição. Sua leveza, interface limpa e suporte a múltiplas linguagens o tornam uma ferramenta bastante eficiente para desenvolvedores. [27]

2.17.3. Bootstrap

O Bootstrap é um framework front-end baseado em HTML, CSS e JavaScript que facilita o desenvolvimento de interfaces web responsivas e modernas. Utilizou-se o Bootstrap para garantir um layout agradável, adaptável a diferentes tamanhos de tela e dispositivos, sem a necessidade de criar tudo do zero em CSS puro. [29]

2.17.4. XAMPP

O XAMP é um pacote com os principais servidores de código aberto do mercado, indluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Pel.

Com ele, é possivel rodar sistemas como WordPress e Drupal localmente, o que facilita e aliza o desenvolvimento. Como o conteido estará armazenado numa rede local, o soesso aos arquivos é realizado instantaneamente. O pacote de servidores é baixado cercade 600 mil vezespor mês, de acordo com dados do SourceForge. Actualmente, o XAMPP está disponível para quatro sistemas operacionais: Windows, Linux, Mac OS X e Solaris. [24]

2.17.5. Mysql Workbench

O MySQL Workbench é uma fantástica ferramenta que permite o desenho e gestão de base de dados. Com esta app pode criar diagramas BER, gerar scripts SQL, gestão de utilizadores, fazer consultas às bases de dados, backups, gestão de privilégios, criar funções e muito mais. [25]

2.17.6. Astah Community

A Astah é uma ferramenta CASE (Computer-Aided Software Engineering) vastamente utilizada para a modelagem de soluções de software fazendo uso da UML. Ela dispõe de uma versão free "community" e de uma versão paga "professional". Astah é desenvolvido na plataforma JAVA e permite que seja modelado soluções de software fazendo uso de uma linguagem que seja mais próxima do pensamento humano. [26]

2.17.7. AJAX

AJAX (Asynchronous JavaScript and XML) é uma técnica de desenvolvimento web que permite a atualização de partes de uma página da web sem precisar recarregá-la completamente. Ele funciona através da combinação de JavaScript e objetos como XMLHttpRequest (ou fetch nas versões modernas), permitindo enviar e receber dados do servidor de forma assíncrona. Isso melhora significativamente a experiência do usuário, deixando as aplicações mais rápidas e interativas. [31]

2.18. Engenharia de Software

Ao passar do tempo, ninguém imaginava que o software se tornaria um elemento muito importante para o mundo e teria a capacidade de manipular a informação. Com muitos elementos computacionais tiveram mudanças até hoje e continuam tendo. Com este crescimento computacional, levam a criação de sistemas perfeitos e problemas para quem desenvolvem softwares complexos.

As preocupações dos engenheiros de software para desenvolverem os softwares sem defeitos e entregarem estes produtos no tempo marcado, assim leva a aplicação da disciplina de engenharia de software. Com o crescimento desse segmento muitas empresas possuem mais especialistas em TI em que cada um tem sua responsabilidade no desenvolvimento de software e é diferente de antigamente que era um único profissional de software que trabalhava sozinho numa sala. [18]

2.18.1. Gerencia de Projectos de Software

Gestão de projectos é um conjunto de práticas que serve de guia a um grupo para trabalhar de maneira produtiva. Ela compreende métodos e ferramentas que organizam as tarefas, identificam sua sequência de execução e dependências existentes, apoia a alocação de recursos e tempo, além de permitir o rastreamento da execução das actividades e medição do progresso relativo ao que foi definido no plano de projecto. [18]

Uma quantidade significativa das actividades actuais é orientada a equipes e envolve múltiplas organizações, sendo estas características determinantes das actividades futuras de projectos. Lidar com equipes e com ambiente corporativo diverso, visand desenvolvimento de (novos) sistemas ou produtos de software, requer uma habilidade que Combina arte e ciência e a isso se denomina gestão de projectos (de software). [18]

Três pilares formam a base da gestão de projectos: ter foco no cliente, fazer a equipe trabalhar bem (leia-se de forma produtiva e colaborativa) e administrar os recursos (de tempo, pessoal. financeiro) do projecto. A gestão de projectos de software compreende actividades que visam assegurar que o (sistema ou produto de) software seja entregue ao cliente no prazo pré-definido e esteja de acordo com os requisitos definidos pelo cliente

Essa necessidade da gestão de projectos se deve ao fato de o desenvolvimento de software estar sempre sujeito às restrições de qualidade, tempo e orçamento. [18]

Além disso, dada a natureza flexível do software, uma boa prática é adoptar um processo de desenvolvimento orientado para arquitectura para que as funcionalidades e restrições existentes no projecto e implementacão do sistema possam ser adequadamente tratadas. Fiste artigo caracteriza a gestão de projectos de software. destaca sua importância e mostra como ela pode impactar directamente no sucesso do projecto e produtividade das empresas do sector. [18]

Qualquer que seja o projecto com o qual você esteja envolvido, o plano de projecto é um documento essencial o qual o gerente de projecto 'vive e respira' ao longo de todo o projecto. É obrigatório o seu desenvolvimento e manutenção. O plano de projecto define os marcos de projecto (isto é, os 'milestones') e as principais actividades necessárias à

execução do projecto. Este documento define a data de cada marco de projecto, as actividades associadas, os artefactos gerados e respectivos responsáveis. [18]

Perceba que é, até certo ponto, natural àqueles que desenvolvem novos produtos e sistemas de software iniciarem as atividades de desenvolvimento antes mesmo que eles entendam o que tem de ser feito, ou seja, antes mesmo de saberem qual é o problema a ser tratado. Esse tipo de atitude, comumente, resulta no insucesso de projectos. Estudos apontam que cerca de 40% de insucesso de projectos são devidos ao mau panejamento ou, até mesmo, a inexistência de plano de projecto, enquanto que quase 20% se deve a uma gestão inadequada do projecto. Interessante destacar que aproximadamente 50% dos problemas e insucesso de projecto se devem a uma 'pobre comunicação entre os principais envolvidos no projecto (ou seja, os stakeholders). Num projecto, as questões essenciais que um gerente deve se fazer são:

- Que problema precisamos solucionar?
- Quais recursos necessito para resolver?
- Quanto tempo disponho para o projeto e implementação da solução?

A lição que fica é: sem o entendimento completo do problema a ser tratado e um bem elaborado plano de projecto em mãos, você (gerente) e sua equipe não saberão onde querem e precisam chegar. A consequência é você (juntamente com sua equipe) deparar-se com a inserção de defeitos logo cedo no desenvolvimento, os quais virão, apenas bem mais tarde, a serem descobertos, resultando em atraso no projecto e/ou comprometendo a qualidade do produto final. [13]

A gestão de projectos é uma actividade ortogonal às demais actividades de projecto e actua como guia para a boa execução do projecto. Todas as pessoas envolvidas com um projecto têm a necessidade de acesso às suas informações. Quando lidamos com projecto de médio a grande porte, e de natureza complexa, uma actividade chave é a coordenação. Um gerente de projecto precisa coordenar:

- Múltiplas pessoas de formação diversa;
- Múltiplas tarefas onde ocorre relação de dependência;
- Uso de múltiplos recursos (como equipamentos, ferramentas, laboratórios);
- Decisão e aprovação em múltiplos pontos num projeto;
- Alocação adequada de recursos humanos e financeiros a tarefas.

Portanto, é preciso dar visibilidade e compartilhar informações de projecto, pois as decisões precisam ser tomadas com base de informações bem entendidas e explicitadas.

A qualidade resultante de um produto ou sistema é determinado a partir do início de seu desenvolvimento. Uma criteriosa análise, feita logo cedo no projecto, visa encontrar erros, identificar inconsistências e averiguar quão correcto e completo é o entendimento do problema e adequada é a solução trabalhada. Isto torna a gestão de projecto uma actividade essencial à execução de projectos e sucesso de produtos. A gestão de

projectos de software pode ser vista sob duas perspectivas: técnica e pessoal onde a ênfase se dá sobre atividades de panejamento e execução. [19]

3. Resultados e Implementação

Este capítulo apresenta o processo de implementação do sistema proposto e os principais resultados obtidos com o desenvolvimento do projeto. Serão descritas as etapas práticas da construção da aplicação, as tecnologias utilizadas, bem como as funcionalidades implementadas com base nos requisitos definidos anteriormente.

Além disso, será realizada uma análise dos resultados alcançados, com foco na usabilidade, funcionalidade e aderência às necessidades identificadas no contexto acadêmico. A proposta é demonstrar como o sistema desenvolvido pode contribuir de forma efetiva para a organização e gestão de projetos acadêmicos, facilitando o dia a dia de estudantes e professores.

A apresentação detalhada do protótipo e sua estrutura funcional visa comprovar a viabilidade da solução e seu potencial para ser aplicada em ambientes reais de ensino superior.

3.1. Metodologias

Para o desenvolvimento do Software de Gestão de Projetos Acadêmicos (SGPA), adotou-se a metodologia Iterativa e Incremental, por ser a mais adequada à realidade de um projeto de TCC desenvolvido individualmente. Esta abordagem combina as vantagens do desenvolvimento incremental (entrega em partes) com o processo iterativo (revisões e melhorias contínuas), permitindo que o sistema evolua de forma progressiva até alcançar sua versão final. [21]

3.1.1. Justificatica da Escolha

A escolha pela metodologia Iterativa e Incremental se fundamenta na sua flexibilidade e adaptabilidade, especialmente em contextos onde apenas um desenvolvedor está envolvido. Esse modelo facilita o acompanhamento do progresso do sistema, permite testes contínuos e possibilita a identificação precoce de falhas ou melhorias, contribuindo diretamente para a construção de um produto mais estável e funcional. [22]

Além disso, a possibilidade de se trabalhar em módulos menores e entregáveis reduz a complexidade do desenvolvimento e possibilita uma melhor gestão do tempo, aspecto fundamental em projetos de conclusão de curso. [23]

3.2. Implementação

3.2.1. Visão Geral do Sistema

O Sistema de Gestão de Projetos Acadêmicos (SGPA) tem como objetivo auxiliar o gerenciamento de projetos no contexto acadêmico, permitindo que docentes, estudantes e administradores possam cadastrar, acompanhar e avaliar projetos de forma eficiente e centralizada.

3.2.2. Ferramentas e Tecnologias Utilizadas

Para o desenvolvimento do presente projecto, foram utilizadas as seguintes ferramentas e tecnologias:

FERRAMENTAS	TECNOLOGIAS
Visual Studio Code	Bootstrap
Sublime Text	AJAX
XAMPP	
MYSQL Workbench	
Astah Community	

3.2.3. Requisitos do Sistema

3.2.3.1. Requisitos Funcionais

Como já definido no seu conceito antes, para facilitar a compreensão esta definida que os requisitos funcionais são requisitos directamente ligado a funcionalidades do sistema, descrevem as funções que o sistema deve executar. Eis os requisitos funcionais que o sistema possui:

RF001-O sistema deve permitir o professor fazer cadastro.

RF002-O sistema deve permitir o estudante fazer cadastro.

RF003-O sistema deve permitir o professor fazer login com email e senha.

RF004-O sistema deve permitir o estudante fazer login com email e senha.

RF005-O sistema deve permitir o estudante ver projectos atribuidos a ele.

RF006-O sistema deve permitir o estudante submeter projectos e enviar arquivo em pdf.
RF007-O sistema deve permitir o estudante fazer download de arquivos em pdf.

RF008-O sistema deve permitir o estudante ver nota de avaliação e feedback.

RF009-O sistema deve permitir o docente registar grupos.

RF0010-O sistema deve permitir o docente criar projectos.

RF0011-O sistema deve permitir o docente ver projectos submetidos pelos estudantes. RF0012-O sistema deve permitir o docente ver lista de projectos.

RF0013-O sistema deve permitir o docente atribuir projecto ao grupo.

RF0014-O sistema deve permitir o docente editar e excluir projectos.

RF0015-O sistema deve permitir o docente avaliar e dar feedback da submissao feito pelos estudantes.

RF0016-O sistema deve permitir o admin realiza todas as tarefas do estudante e docente. RF0017-O sistema deve permitir o administrador registar estudante.

RF0018-O sistema deve permitir o administrador registar professor.

RF0019-O sistema deve permitir o administrador Activar conta do professor.

RF020-O sistema deve permitir o administrador eliminar conta de usuarios(Estudante e Professor)

RF021-Os projectos devem ter prazos de entrega.

RF022- O sistema deve ter Dashboards e gráficos.

RF023- O sistema deve permitir Upload e download de arquivos.

3.2.3.2. Requisistos Não Funcionais

Os requisitos não funcionais descrevem os critérios específicos que podem ser usados para avaliar o funcionamento de um sistema, exemplo, desempenho, segurança, disponibilidade. Abaixo temos os requisitos não funcionais:

RNF001- O sistema deve criptografar as senhas dos usuários. (Segurança)

RNF002- O sistema deve controlar o acesso por tipo de usuário, admin, professor, estudante. (Segurança).

RNF003- A busca por projetos deve retornar resultados em até 3 segundos. (Desempenho).

RNF004- O sistema deve ter interface simples, amigável e fácil de usar(Usabilidade). RNF005- Deve haver mensagens claras no sistema em caso de erro ou sucesso (Usabilidade).

RNF006- O sistema deve funcionar nos principais navegadores Chrome, Firefox, Edge (Compatibilidade).

RNF007- O sistema deve poder ser instalado em outros servidores com pouca configuração (Portabilidade).

RNF008- O sistema deve ser adaptável a diferentes tamanhos de tela(Compatibilidade).
RNF009- O sistema deve permitir adicionar novos usuários ou funcionalidades no futuro (Escalabilidade).

RNF010- O sistema deve estar acessível 24 horas por dia, 7 dias por semana (Disponibilidade).

3.2.4. Arquitectura de Software

Arquitetura de Software Utilizada: Modelo Cliente-Servidor com Arquitetura em Camadas (MVC).

O sistema SGPA foi desenvolvido seguindo a arquitetura Cliente-Servidor, com a separação de responsabilidades baseada no padrão MVC (Model-View-Controller). Esse modelo é amplamente adotado no desenvolvimento de aplicações web devido à sua modularidade, organização e facilidade de manutenção.

- Cliente (Front-End)

A camada cliente é composta pelas interfaces web acessadas por estudantes, docentes e administradores através de navegadores. Foi construída com:

HTML5 e CSS3, para estruturação e estilização das páginas;

JavaScript, para interatividade e manipulação dinâmica dos elementos;

Bootstrap, para garantir responsividade e design moderno.

- Servidor (Back-End)

A camada do servidor foi implementada em PHP, responsável por processar as requisições feitas pelo cliente, aplicar regras de negócio e interagir com a base de dados.

Utiliza o padrão MVC:

Model: Responsável pela manipulação dos dados (CRUD, regras de negócio e integração com MySQL);

View: Responsável pela apresentação das páginas aos usuários;

Controller: Faz a ponte entre o Model e a View, coordenando a lógica do sistema.

- Banco de Dados

A camada de dados foi implementada utilizando MySQL, onde são armazenadas informações de usuários, projetos, grupos, submissões, avaliações e interações entre os perfis do sistema.

3.2.5. Diagrama de Caso de Uso

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário (actor), ele descreve as principais funcionalidades do sistema e a interacção. O diagrama de caso de uso não oferece muitos detalhes, não espere, por exemplo, que ele mostre a ordem em que os passos são executados. Em vez disso, um diagrama de caso de uso adequado dá uma visão geral do relacionamento entre casos de uso, actores e sistemas.[32]Para o nosso projecto separamos o diagrama de caso de uso em três atores conforma as figuras abaixo:

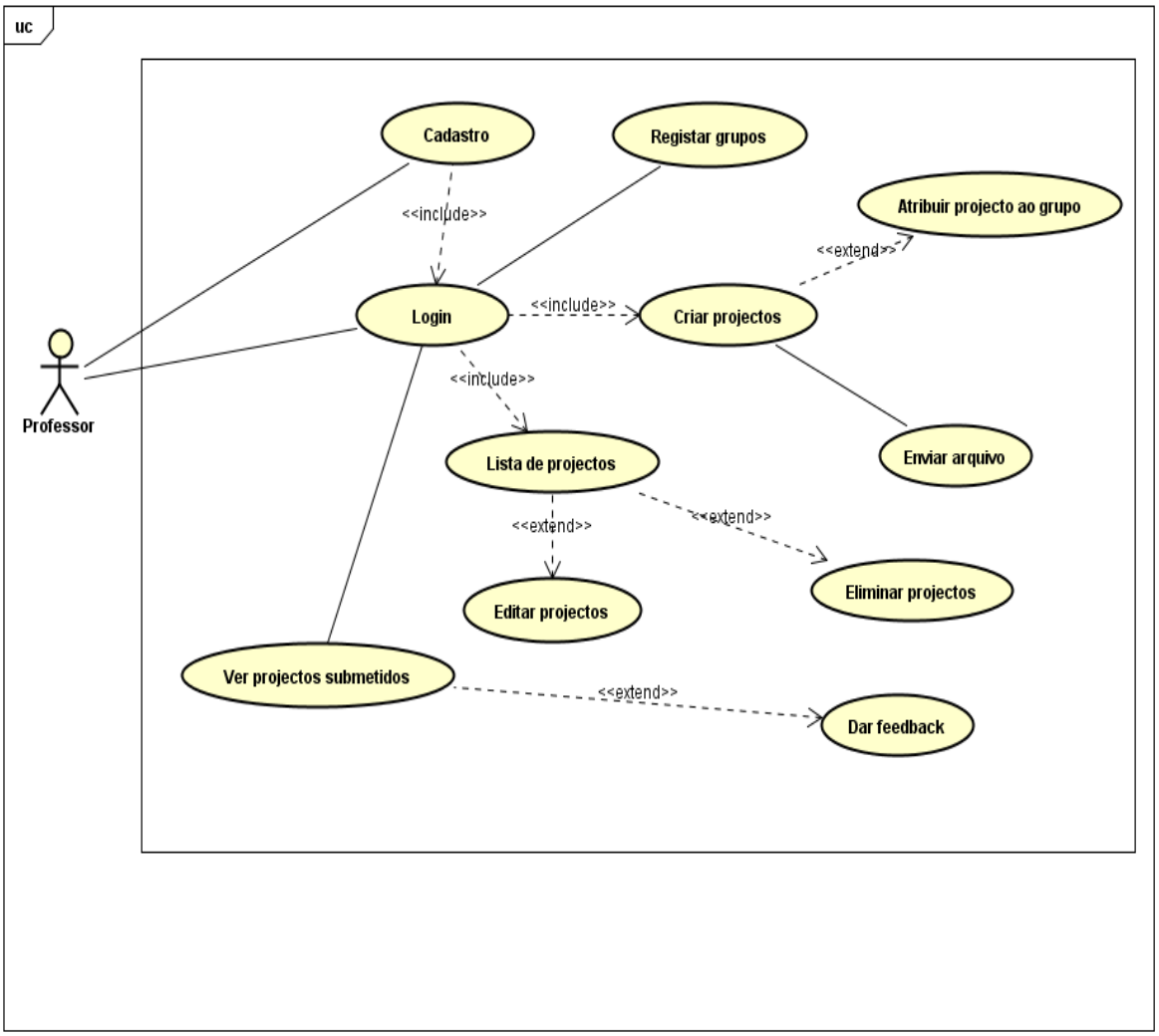

Figura 1-Diagrama de caso de uso (Professor)

Fonte: Aut

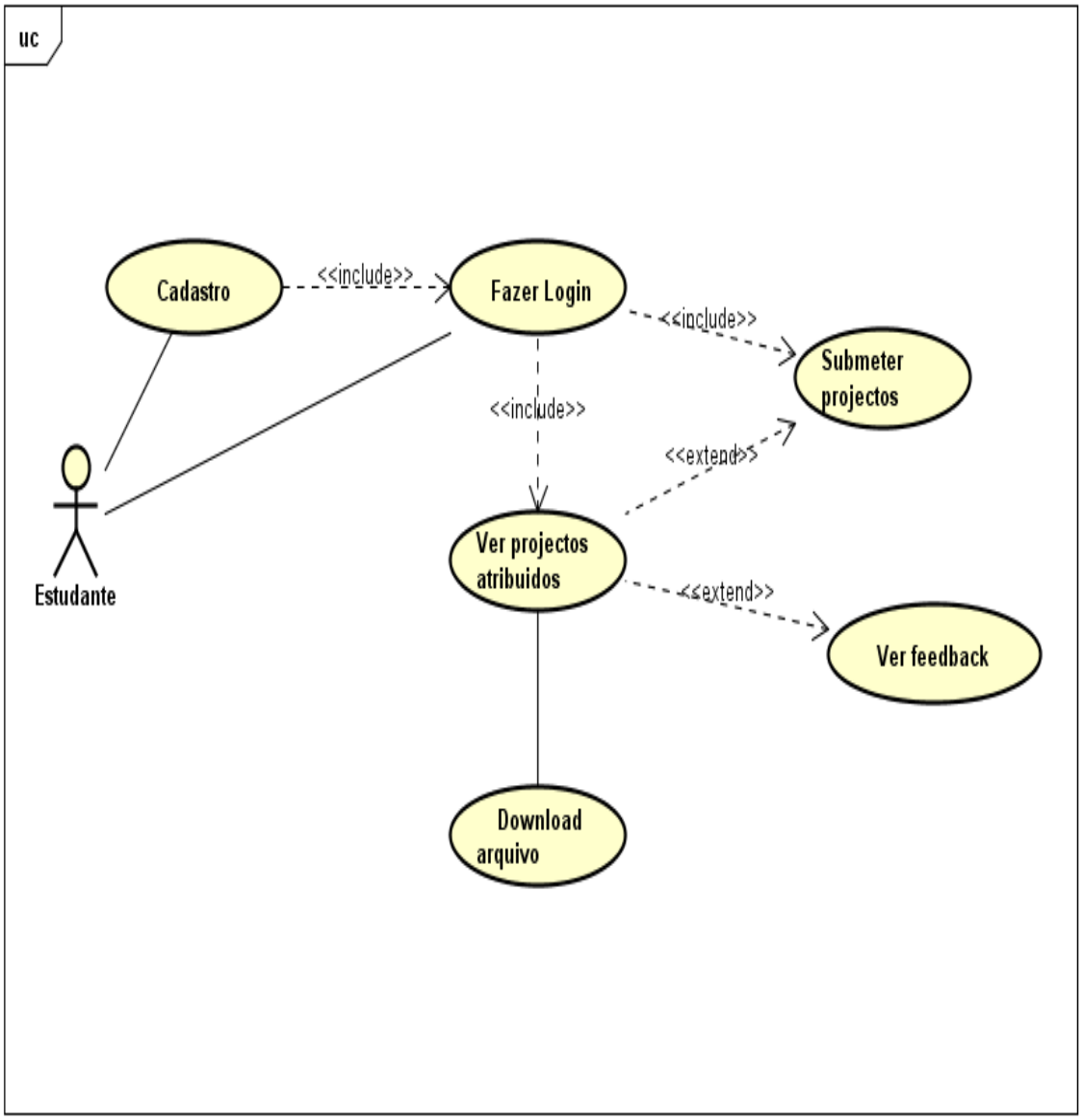

Figura 1-Diagrama de caso de uso (Estudante)

Fonte: Autor

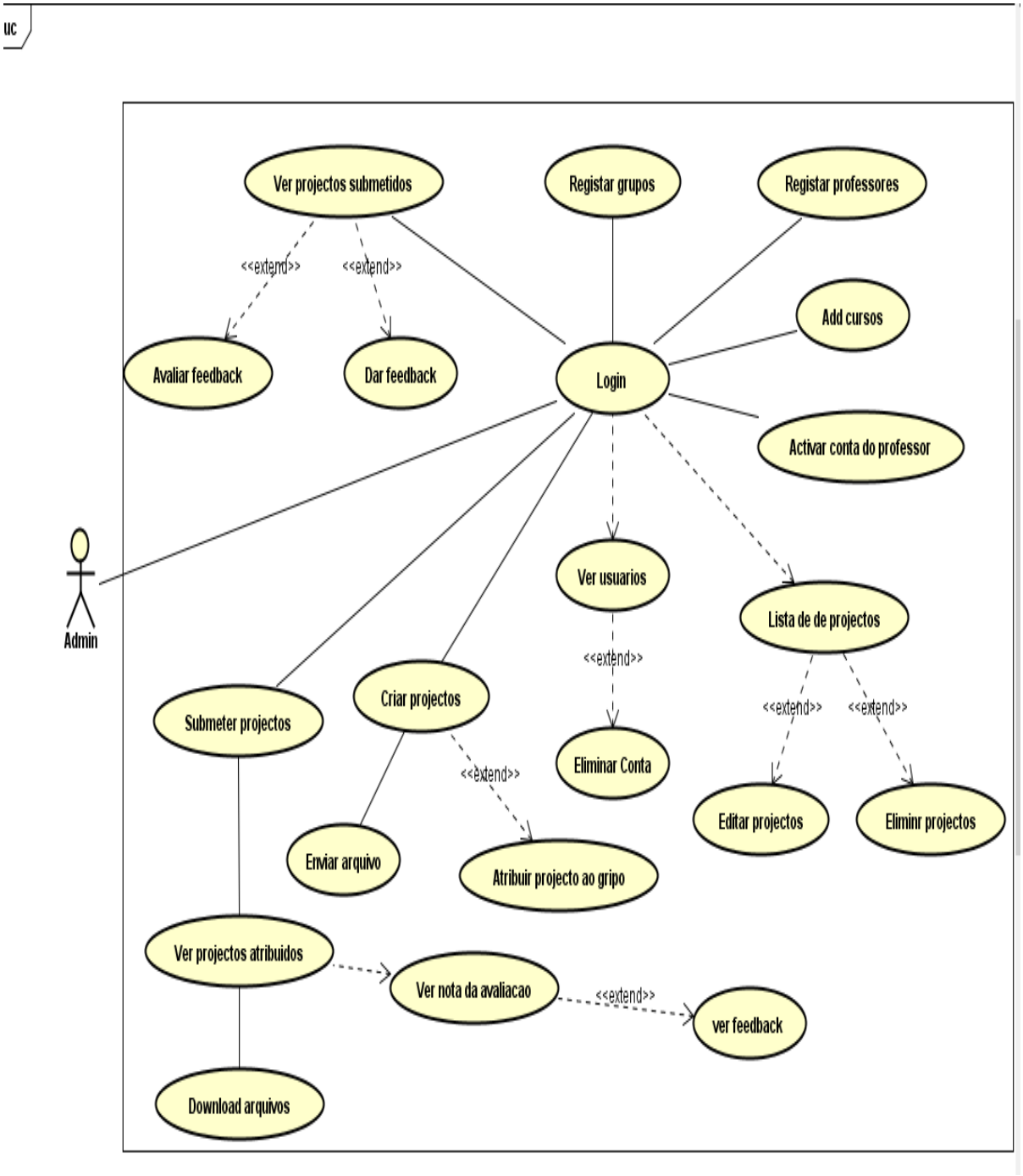

Figura 1-Diagrama de caso de uso (Admin)

Fonte: Autor

3.2.6. Diagrama de Classes

Permite a visualização de um conjunto de classes, detalhando atributos e operações (métodos) presentes nesta ultima, assim como prováveis relacionamentos entre essas extruturas. Este tipo de representação pode incluir ainda definições de interfaces. [32]

3.2.7. Diagrama de Actividades

Contempla as diversas tarefas desempenhadas na execução de uma actividade, sendo utilizado geralmente na representação de processos dentro de uma empresa/organização. [32]

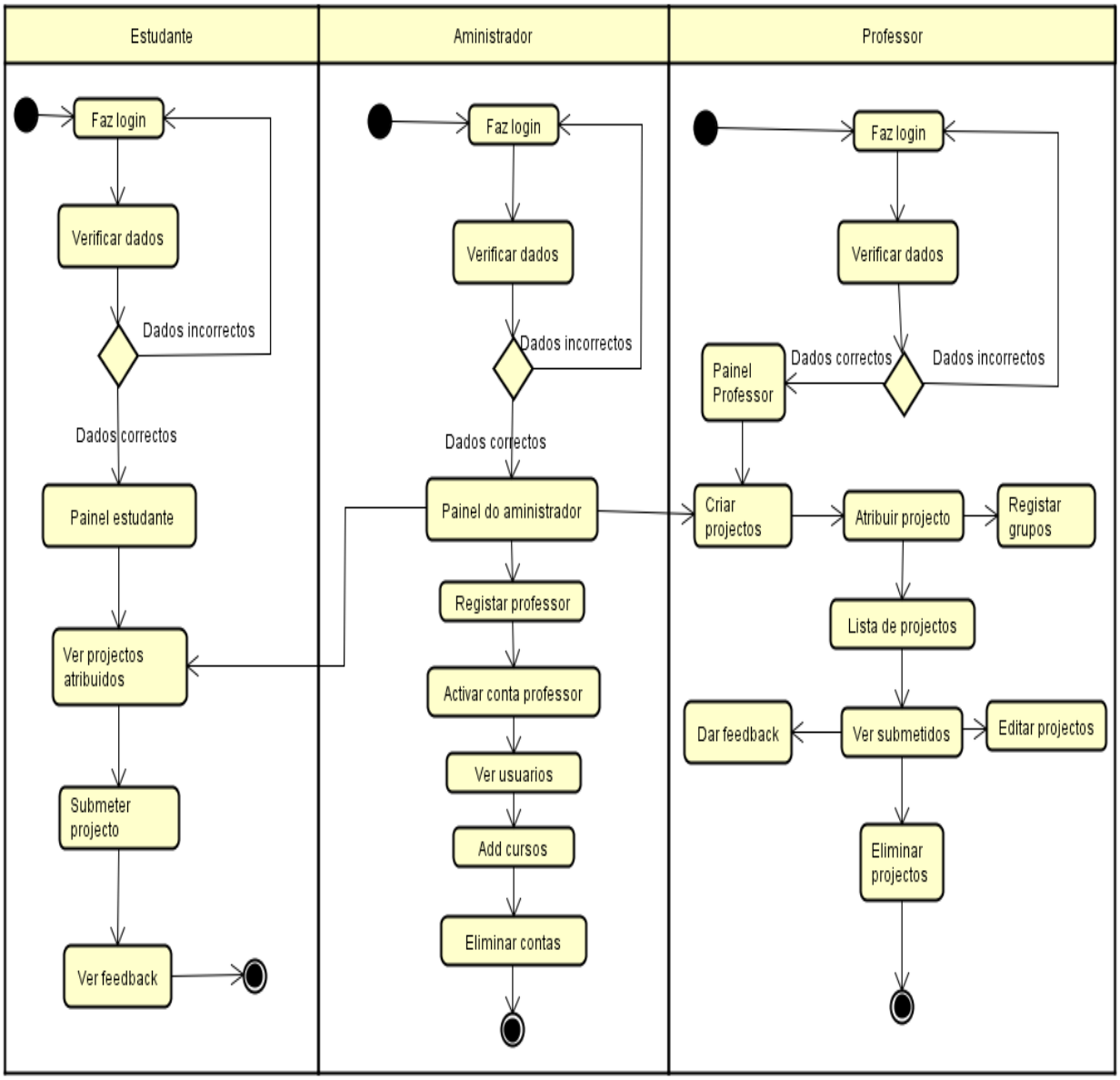

Figura 1-Diagrama de actividade

Fonte: Autor

3.2.8. Diagrama de Sequência

Demostra as interações entre diferentes objectos na execução de uma operação, destacando ainda em que tais acções acontecem num intervalo de tempo. A sequência em que as diversas operações são executadas ocorre na vertical, de cima para baixo. [32]

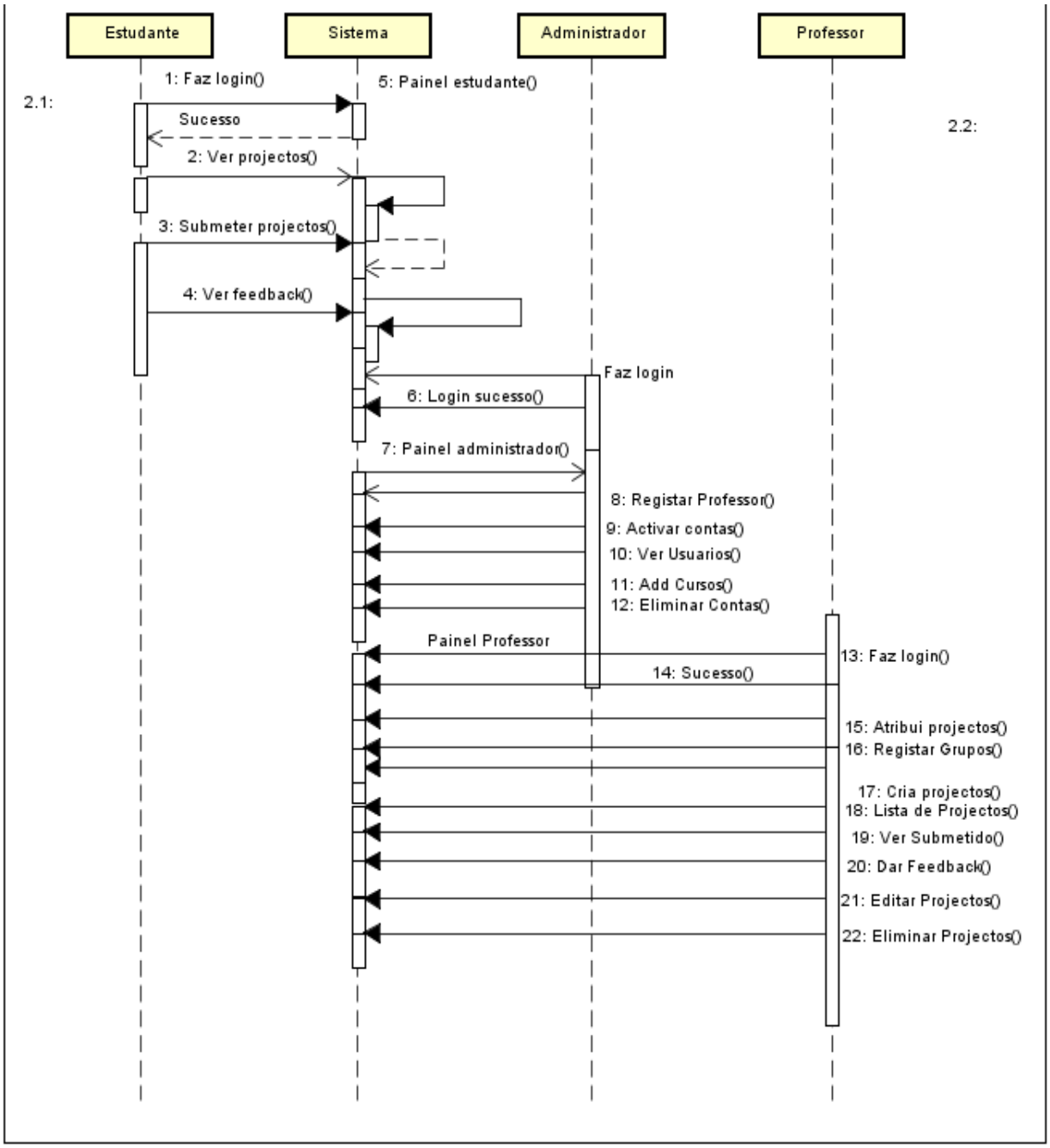

Figura 1-Diagrama de sequência

Fonte: Autor

3.2.9. Diagrama de Componente

É um tipo de diagrama na linguagem UML que representa a estrutura física de um sistema, focando-se em como os componentes de software estão organizados e interagem entre si. Permite visualizar a arquitectura do sistema e compreender as suas dependências. [32]

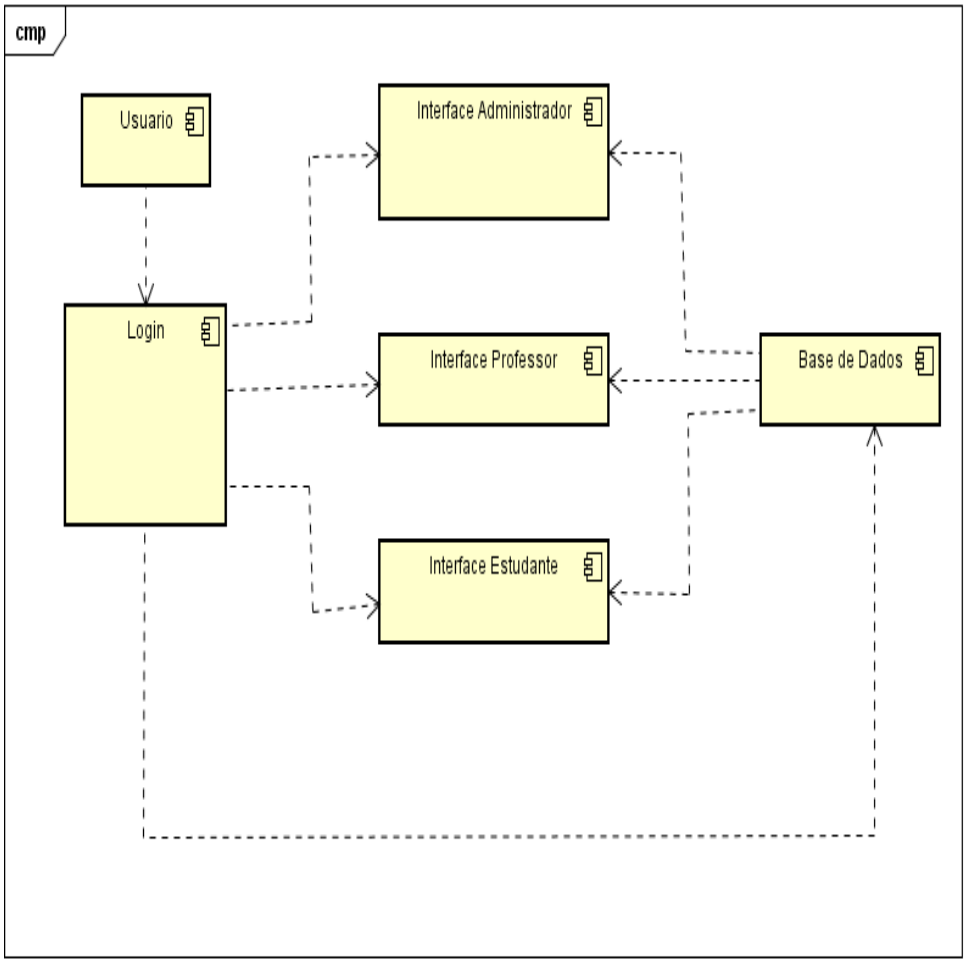

Figura 1-Diagrama de componentes

Fonte: Autor

3.3. Resultados

3.3.1. Analise de Resultados

Com a conclusão do desenvolvimento, foi possível validar os requisitos estabelecidos na fase de análise. A seguir estão os principais resultados alcançados:

- Implementação completa do fluxo de cadastro e login com controle por tipo de usuário (Administrador, Docente e Estudante).
- Painéis personalizados conforme o perfil do usuário, garantindo acessos específicos.
- Cadastro e gerenciamento de projetos por parte dos docentes.
- Vinculação de estudantes a grupos e projetos de forma dinâmica.
- Envio de projetos pelos estudantes, com upload de arquivos e validações.
- Submissões visíveis apenas para os docentes responsáveis pelos projetos atribuídos.
- Utilização de modais e Ajax para melhorar a experiência do usuário sem recarregamento de páginas.
- Responsividade assegurada pelo uso do Bootstrap.

Esses resultados demonstram que o sistema atende aos objetivos propostos, proporcionando uma plataforma funcional para a gestão de projetos acadêmicos em ambiente universitário.

4. Conclusões

4.1. Conclusão

O desenvolvimento do Sistema de Gestão de Projetos Acadêmicos (SGPA) permitiu alcançar os objetivos propostos neste trabalho, oferecendo uma solução eficiente e intuitiva para o gerenciamento de projetos no ambiente universitário. Através da aplicação da metodologia de desenvolvimento estruturado, aliada ao uso de tecnologias modernas, foi possível criar uma plataforma funcional, segura e de fácil usabilidade.

Conclui-se, portanto, que o sistema desenvolvido não só atende à necessidade de modernizar a gestão de projetos acadêmicos, como também contribui para a melhoria da comunicação entre docentes e estudantes, promovendo um ambiente mais organizado, interativo e eficiente. Espera-se que, futuramente, o SGPA possa ser expandido com novas funcionalidades, tais como notificações automáticas, relatórios estatísticos e integração com sistemas institucionais, ampliando ainda mais sua utilidade e impacto na comunidade acadêmica.

Trabalhos Futuros

Referências Bibliográficas

[1] PMI. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 6th ed. Project Management Institute, 2017.

[2] SOUZA, Flávio. Metodologias Ativas e Tecnologias Educacionais. Penso Editora, 2020.

[3] PRESSMAN, Roger S. Engenharia de Software. 8ª ed. McGraw Hill Brasil, 2016.

[4] MORAN, José Manuel. A educação que desejamos: novos desafios e como chegar lá. Papirus, 2018.

[5]KENSKI, Vani Moreira. Educação e Tecnologias: o novo ritmo da informação. Campinas: Papirus, 2012.

[6] NIXON, Robin. Learning PHP, MySQL & JavaScript: with jQuery, CSS & HTML5. 5ª ed. O'Reilly Media, 2018.

[7] NIELSEN, Jakob. Usabilidade na Web. Alta Books, 2005.

[8] STALLINGS, William. Criptografia e segurança de redes: princípios e práticas. Pearson Prentice Hall, 2014.

[9] MARCOTTE, Ethan. Responsive Web Design. A Book Apart, 2010.

[10] DEVMEDIA. “Política de Segurança da Informação e Firewall”, Revista Infra Magazine, [Online]. Disponível em: https://www.devmedia.com.br/politica-de-seguranca-da-informacao-e-firewall-revista-infra-magazine-2/22234. [Acesso em: 9 abr. 2022].

[11] DEVMEDIA. “Introdução às Linguagens de Programação”, [Online]. Disponível em: https://www.devmedia.com.br/introducao-as-linguagens-de-programacao/25111. [Acesso em: 27 maio 2022].

[12] Nixon, R. (2018). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (5th ed.). O'Reilly Media.

[13] Welling, L., & Thomson, L. (2017). PHP and MySQL Web Development (5th ed.). Addison-Wesley Professional.

[14] DATE, C. J. Introdução a Sistemas de Bancos. Elsevier Editora, 2004, 2022.

[15] ELMASRI, R.; NAVATHE, S. Fundamentals of Database Systems. Pearson, 2015.

[16] CORONEL, C.; MORRIS, S. Database Systems: Design, Implementation, & Management. Cengage Learning, 2022.

[17] MAURER, A. Criação de páginas web com HTML5 e CSS3. Kindle Unlimited, 2021.

[18] PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. McGraw-Hill Education, 2021.

[19] VALENTE, M. T. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Kindle Edition, 2020.

[20] WAZLAWICK, R. S. Engenharia de Software: Conceitos e Práticas. Federal University of Santa Catarina: Elsevier Editora, 2013.

[21] PRESSMAN, Roger S. Engenharia de Software. 7. ed. São Paulo: McGraw-Hill, 2011.

[22] SOMMERVILLE, Ian. Engenharia de Software. 10. ed. São Paulo: Pearson, 2019.

[23] BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 3. ed. Rio de Janeiro: Elsevier, 2011.

[24] Apache Friends. XAMPP. Disponível em: https://www.apachefriends.org. Acesso em: jul. 2025.
[25] MySQL. MySQL Workbench. Disponível em: https://dev.mysql.com/workbench. Acesso em: jul. 2025.
[26] ASTAH. Modelagem com UML usando Astah. Disponível em: https://astah.net. Acesso em: jul. 2025.
[27] Microsoft. Visual Studio Code. Disponível em: https://code.visualstudio.com. Acesso em: jul. 2025.
[28] Sublime HQ. Sublime Text. Disponível em: https://www.sublimetext.com. Acesso em: jul. 2025.
[29] Bootstrap. Documentação Oficial do Bootstrap. Disponível em: https://getbootstrap.com. Acesso em: jul. 2025.

[30] LAZZERI, Alexandre. Padrões de Arquitetura de Software: Teoria e Prática com Java e .NET. São Paulo: Novatec, 2016.

[31] WANG, Paul. AJAX: Creating Web Pages with Asynchronous JavaScript and XML. Boston: Course Technology PTR, 2007.

[32] LARMAN, Craig. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientados a Objetos e ao Processo Unificado. 3. ed. Porto Alegre: Bookman, 2007.

Apêndice

Anexos