



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Desenvolvimento de aplicação descentralizada para aprendizado online

Autor: Arthur de Melo e Eliás Yousef
Orientador: Dr. Orientador

Brasília, DF
2025



Arthur de Melo e Eliás Yousef

Desenvolvimento de aplicação descentralizada para aprendizado online

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Orientador

Coorientador: (quando houver, Titulação Acadêmica e Nome do Orientador)

Brasília, DF

2025

Arthur de Melo e Eliás Yousef

Desenvolvimento de aplicação descentralizada para aprendizado online/ Arthur de Melo e Eliás Yousef. – Brasília, DF, 2025-
59 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Orientador

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2025.

1. Descentralização. 2. IPFS. I. Dr. Orientador. II. Universidade de Brasília.
III. Faculdade UnB Gama. IV. Desenvolvimento de aplicação descentralizada
para aprendizado online

CDU 02:141:005.6

Errata

Elemento opcional da ??, 4.2.1.2). **Caso não deseje uma errata, deixar todo este arquivo em branco.** Exemplo:

FERRIGNO, C. R. A. **Tratamento de neoplasias ósseas apendiculares com reimplantação de enxerto ósseo autólogo autoclavado associado ao plasma rico em plaquetas:** estudo crítico na cirurgia de preservação de membro em cães. 2011. 128 f. Tese (Livre-Docência) - Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, São Paulo, 2011.

Folha	Linha	Onde se lê	Leia-se
1	10	auto-conclavo	autoconclavo

Arthur de Melo e Eliás Yousef

Desenvolvimento de aplicação descentralizada para aprendizado online

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

Dr. Orientador
Orientador

Titulação e Nome do Professor
Convidado 01
Convidado 1

Titulação e Nome do Professor
Convidado 02
Convidado 2

Brasília, DF
2025

**A dedicatória é opcional. Caso não deseje uma, deixar todo este arquivo em
branco.**

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

A inclusão desta seção de agradecimentos é opcional, portanto, sua inclusão fica a critério do(s) autor(es), que caso deseje(em) fazê-lo deverá(ão) utilizar este espaço, seguindo a formatação de *espaço simples e fonte padrão do texto (arial ou times, tamanho 12 sem negritos, aspas ou itálico*.

Caso não deseje utilizar os agradecimentos, deixar toda este arquivo em branco.

A epígrafe é opcional. Caso não deseje uma, deixe todo este arquivo em
branco.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto. O texto pode conter no mínimo 150 e no máximo 500 palavras, é aconselhável que sejam utilizadas 200 palavras. E não se separa o texto do resumo em parágrafos.

Palavras-chaves: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Wavelets correlation coefficients	41
--	----

Lista de tabelas

Tabela 1 – Propriedades obtidas após processamento	42
--	----

Lista de abreviaturas e siglas

Fig. Area of the i^{th} component

456 Isto é um número

123 Isto é outro número

lauro cesar este é o meu nome

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence

Sumário

1	INTRODUÇÃO	27
	Introdução	27
1.1	Objetivos	27
1.1.1	Objetivo Geral	27
1.1.2	Objetivos Específicos	28
2	FUNDAMENTAÇÃO TEÓRICA	29
	Fundamentação Teórica	29
2.1	O problema das aplicações centralizadas	29
2.2	O <i>InterPlanetary File System</i> (IPFS)	29
2.2.1	OrbitDB: Banco de Dados Distribuído	30
2.2.2	Arquitetura de desenvolvimento web	30
2.2.3	Principais vantagens do modelo Cliente-Servidor	30
2.3	Desenvolvimento Cliente-Servidor	31
2.3.1	NestJS	32
2.3.2	OrbitDB	32
2.3.3	Next.js	32
2.3.4	PostgreSQL	33
3	METODOLOGIA	35
3.1	Extreme Programming (XP)	35
3.1.1	Particularidades do XP no Projeto	35
3.2	Kanban	35
3.2.1	Trello	35
3.2.2	Uso do Kanban com Trello	35
I	TEXTO E PÓS TEXTO	37
4	ELEMENTOS DO TEXTO	39
4.1	Corpo do Texto	39
4.2	Títulos de capítulos e seções	39
4.3	Notas de rodapé	39
4.4	Equações	40
4.5	Figuras e Gráficos	40

4.6	Tabela	41
4.7	Citação de Referências	42
5	ELEMENTOS DO PÓS-TEXTO	45
5.1	Referências Bibliográficas	45
5.2	Anexos	45
	REFERÊNCIAS	47
	APÊNDICES	49
	APÊNDICE A – PRIMEIRO APÊNDICE	51
	APÊNDICE B – SEGUNDO APÊNDICE	53
	ANEXOS	55
	ANEXO A – PRIMEIRO ANEXO	57
	ANEXO B – SEGUNDO ANEXO	59

1 Introdução

A Web (World Wide Web) foi concebida como uma plataforma aberta e acessível para todos, promovendo liberdade de acesso e compartilhamento de informações de qualquer lugar e hora. Porém, o funcionamento da Web está atrelado ao modelo cliente-servidor, e isso acarreta que a maioria das máquinas que participam do processo de compartilhamento de informações sejam apenas coadjuvantes, consumindo serviços de outras máquinas servidoras (1). Para contrapor esta limitação, o projeto XXX tem por finalidade desenvolver uma aplicação segura e descentralizada através do paradigma P2P (peer-to-peer), para aprendizado comunitário de assuntos gerais.

A descentralização surge como uma alternativa inovadora, promovendo resiliência, maior privacidade e autonomia dos usuários ao permitir que dados sejam distribuídos e armazenados em redes ponto a ponto, sem depender de servidores centralizados. Com isso, o projeto XXX foi arquitetado para garantir a integridade e privacidade dos dados sensíveis do material didático, onde o proprietário do material possui total controle sobre seu conteúdo, sendo capaz de acessá-lo e distribuí-lo da forma como quiser, protegendo o material contra acessos não autorizados e manipulações, promovendo segurança e confiança no ambiente online.

O projeto adota uma arquitetura cuidadosamente planejada para alcançar esses objetivos. Como peça central da arquitetura, temos o protocolo IPFS (InterPlanetary File System), que possibilita o armazenamento e a distribuição de arquivos de forma descentralizada, e o OrbitDB, que se trata de uma base de dados distribuída construída sobre o IPFS, que garante a persistência e a integridade das informações em um ambiente peer-to-peer. Para facilitar a implementação do IPFS na linguagem JavaScript, foi utilizada a Helia, uma biblioteca que permite a interação simplificada com a rede descentralizada, além do Nest para construção de APIs escaláveis. A escolha dessas tecnologias para o desenvolvimento da aplicação foi motivada pela flexibilidade e desempenho.

Dessa forma, a utilização dessas tecnologias torna possível a criação de uma aplicação inovadora e descentralizada, que redefine a forma como o aprendizado online é conhecido.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver uma plataforma de aprendizado comunitário que utilize uma arquitetura moderna e segura, garantindo integridade e privacidade dos dados sensíveis através

de técnicas de criptografia e tecnologias descentralizadas.

1.1.2 Objetivos Específicos

- Implementar uma arquitetura back-end eficiente e escalável utilizando o framework NestJS integrado com banco de dados Postgres e outras soluções como OrbitDB e Helia, para o gerenciamento seguro dos dados.
- Garantir a privacidade e segurança dos dados dos usuários por meio de técnicas de criptografia, armazenando apenas informações necessárias e sensíveis de forma protegida.
- Utilizar o IPFS (InterPlanetary File System) para o armazenamento descentralizado de materiais de curso, otimizando a disponibilidade e a proteção dos arquivos.
- Implementar um sistema de controle de acesso robusto que permita o compartilhamento seletivo de materiais com base nos níveis de permissão dos usuários, assegurando que conteúdos privados sejam acessíveis apenas para os usuários autorizados.
- Fornecer uma interface amigável e responsiva para os administradores e alunos, desenvolvida com NextJS, que facilite o acesso aos cursos e materiais disponibilizados na plataforma.
- Documentar e testar a solução para validar a conformidade da plataforma com os requisitos funcionais e de segurança estabelecidos.

2 Fundamentação Teórica

2.1 O problema das aplicações centralizadas

A arquitetura tradicional em que a internet se baseou, cliente-servidor, tem sido predominante em várias aplicações, como redes sociais, mensageiros, aplicações de busca. Apesar deste modelo ter se mostrado eficiente no decorrer do tempo, ele apresenta desafios significativos como a centralização dos dados. Segundo Wanner ([WANNER, 2024](#)), problemas como a vulnerabilidade dos dados, a privacidade dos usuários e a censura são aspectos a serem avaliados nessa arquitetura que podem acarretar problemas de segurança e confiança. A forte tendência à centralização pode ser confirmada através do estudo da LogicMonitor ([LOGICMONITOR, 2023](#)) que confirma que com a pandemia COVID-19, a migração de dados das empresas para a nuvem foi acelerada 87% e planeja seguir com esse processo. Diante dessa problemática, soluções como o *InterPlanetary File System* (IPFS), bitcoin, nostr foram pensadas com o cerne na descentralização e democratização dos dados.

Neste projeto, o foco é desenvolver uma plataforma descentralizada para cursos e materiais educativos, utilizando o IPFS para garantir que os dados permaneçam sob o controle direto de seus criadores. A aplicação será responsável por gerenciar os dados dos usuários e no controle dos meios de pagamento através da rede *Lightning Network* ([NETWORK,](#)) para os conteúdos privados criptografados. Tanto os materiais públicos quanto os privados serão persistidos na rede IPFS. Dessa forma, mesmo que eventualmente essa aplicação saia de circulação ou seja bloqueada, os dados permanecerão disponíveis globalmente ao público.

2.2 O *InterPlanetary File System* (IPFS)

O IPFS é um protocolo distribuído para armazenamento e compartilhamento de dados, que organiza arquivos em uma estrutura baseada em conteúdo, tornando a web mais eficiente e resistente a falhas ([LABS, 2025](#)). Com ele é possível persistir os dados da aplicação entre os nós da rede. Isso o torna público, de fato, e replicável por quaisquer usuários que quiserem. Diferente do HTTP, que carrega arquivos de um servidor, o IPFS utiliza um modelo de endereçamento de conteúdo, em que os dados não desacoplados de sua localização e tratados a partir da hash única gerada pelo conteúdo do mesmo.

Entre suas principais características, destacam-se:

- **Identificadores de Conteúdo (CIDs - Content Identifiers):** cada conteúdo

ou arquivo é associado a um endereço único baseado em um hash criptográfico, garantindo sua integridade e imutabilidade.

- **Rede Peer-to-Peer (P2P):** milhares de nós interconectados facilitam a localização e recuperação dos dados distribuídos na rede.
- **Armazenamento em Cache:** os dados são armazenados temporariamente na memória cache dos nós, otimizando a largura de banda e melhorando a eficiência na redistribuição de conteúdos.

2.2.1 OrbitDB: Banco de Dados Distribuído

O OrbitDB é um banco de dados *serverless* e ponto a ponto (*peer-to-peer*), distribuído construído para atuar em aplicações descentralizadas. Ele usa o IPFS para o armazenamento dos dados e garante a sincronia dos dados entre os diferentes nós. Ele será utilizado no projeto para realizar a comunicação e persistência dos dados com a rede IPFS.

2.2.2 Arquitetura de desenvolvimento web

A arquitetura *cliente-servidor* será adotada no desenvolvimento da aplicação, devido à sua robustez e flexibilidade no gerenciamento de interações entre usuários e servidores. Nesse modelo, os clientes fazem requisições a um servidor centralizado, que processa as solicitações e retorna as respostas adequadas. A utilização dessa abordagem oferece uma série de benefícios importantes para o desenvolvimento de sistemas distribuídos.

A arquitetura escolhida para o desenvolvimento da aplicação é a *cliente-servidor*. Ela será adotada por conta de sua flexibilidade e robustez no gerenciamento de interações entre os usuários e os servidores. Essa parte da aplicação ficará encarregada pelo gerenciamento dos usuários, bem como pelo serviço de gerar as páginas estáticas que serão persistidas na rede ipfs com os dados gerados. O uso de uma abordagem híbrida concilia as vantagens da rede distribuída descritas acima com as vantagens dessa arquitetura:

2.2.3 Principais vantagens do modelo Cliente-Servidor

- **Controle:** Apesar do cerne dessa aplicação ser a descentralização e democratização dos dados, a gestão dos usuários, bem como o controle do meio de pagamento via rede lightning foi pensado em uma aplicação centralizada. Com esse modelo é possível garantir a integridade e a consistência das informações. Isso permite que a administração tenha maior controle, minimizando riscos de dados corrompidos ou acesso não autorizado(COULOURIS et al., 2011).

- **Escalabilidade:** A arquitetura cliente-servidor comporta um grande número de usuários. As aplicações nessa estrutura podem ser aprimoradas com mais recursos de acordo com a demanda (COULOURIS et al., 2011).
- **Desempenho e Eficiência:** A separação clara entre duas entidades corrobora com a performance da aplicação. O cliente foca na interação com o usuário e como esse dado será apresentado a ele. O servidor lida com processamento e armazenamento de dados. (COULOURIS et al., 2011).
- **Facilidade de Manutenção e Atualização:** Como o processamento centralizado ocorre no servidor, qualquer atualização ou manutenção do sistema pode ser realizada em um único ponto. Isso facilita a implementação de novos recursos, correção de falhas ou melhorias no sistema sem a necessidade de alterar os clientes (COULOURIS et al., 2011).
- **Flexibilidade e Compatibilidade:** Com essa arquitetura é possível distribuir a aplicação para uma gama de plataformas e alcançar um número maior de clientes, como aplicativos móveis, web ou desktop. (COULOURIS et al., 2011).

Dessa forma, a utilização desse modelo em conjunto com a abordagem descentralizada, contribuiu para o desenvolvimento de uma plataforma escalável, segura, eficiente e democrática.

2.3 Desenvolvimento Cliente-Servidor

A construção de uma aplicação percorre várias etapas, desde a concepção da ideia e validação até a prototipação e, seguidamente, codificação resultando em uma solução. Para atingir esse objetivo de maneira eficiente, é necessário escolher ferramentas e tecnologias que suportem o tamanho e robustez que a aplicação se propõe. No projeto XXX, serão utilizadas tecnologias que atendam esses critérios.

Do lado do servidor, a aplicação será escrita a partir do framework NestJS, que se destaca por seu conjunto de ferramentas integradas capazes de comunicar-se com o banco de dados e construção de APIs rapidamente. É uma ferramenta contruída em JavaScript / TypeScript que proporcionará uma integração boa com o OrbitDB, haja visto que este é escrito na mesma linguagem.

Para o desenvolvimento do lado do cliente será adotado o nextJS, que atualmente é um framework amplamente utilizado por conta de suas facilidades no desenvolvimento web.

2.3.1 NestJS

NestJS é uma estrutura para construir aplicativos Node.js eficientes e escaláveis. Ele utiliza JavaScript progressivo, sendo construído com e oferecendo suporte total a TypeScript (mas ainda permitindo que os desenvolvedores codifiquem em JavaScript puro) e combina elementos de POO (Programação Orientada a Objetos), FP (Programação Funcional) e FRP (Programação Reativa Funcional) ([FRAMEWORK, 2025](#)). (Texto traduzido por autor).

Pontos Fortes do NestJS:

- **Modularidade:** Arquitetura que facilita a divisão da aplicação em módulos reutilizáveis.
- **Integração simples com bancos de dados:** Compatível com ORM populares, como TypeORM e Prisma.
- **Injeção de Dependência:** Simplifica o gerenciamento de dependências e aumenta a testabilidade.
- **Flexibilidade:** Suporta diversos paradigmas, como programação reativa e micro-serviços.

2.3.2 OrbitDB

O **OrbitDB** é um banco de dados distribuído e descentralizado, construído sobre o protocolo *IPFS* (*InterPlanetary File System*). Sua arquitetura descentralizada permite alta disponibilidade e sincronização eficiente de dados entre nós. Como é escrito em *JavaScript*, oferece uma integração perfeita com o NestJS.

Pontos Fortes do OrbitDB:

- **Descentralização:** Dispensa a necessidade de servidores centralizados.
- **Alta disponibilidade:** Suporta replicação automática e sincronização entre nós.
- **Desempenho escalável:** Ideal para aplicações distribuídas que requerem persistência de dados em tempo real.
- **Fácil integração:** Compatível com tecnologias baseadas em JavaScript.

2.3.3 Next.js

O **Next.js** é um framework *React* para desenvolvimento de aplicações no lado do cliente. Ele oferece recursos como renderização no lado do servidor (*Server-Side Rendering*

- SSR) e geração de sites estáticos (*Static Site Generation* - SSG), que permitem criar interfaces modernas com excelente desempenho.

Pontos Fortes do Next.js:

- **Renderização otimizada:** Suporte a SSR e SSG, melhorando o desempenho e SEO.
- **Rotas automáticas:** Simplifica a estrutura de navegação com rotas baseadas em arquivos.
- **Desenvolvimento eficiente:** Hot reloading e compilação automática durante o desenvolvimento.
- **API Routes:** Permite criar APIs no mesmo projeto do front-end.
- **Suporte ativo da comunidade:** Grande número de extensões e soluções prontas.

2.3.4 PostgreSQL

O **PostgreSQL** é um sistema de gerenciamento de banco de dados relacional de código aberto amplamente reconhecido por sua robustez, flexibilidade e conformidade com padrões SQL. Ele oferece suporte avançado a diversos tipos de dados e permite a criação de extensões personalizadas, tornando-o uma escolha ideal para aplicações modernas e escaláveis.

Pontos Fortes do PostgreSQL:

- **Suporte Avançado a Dados:** Compatível com tipos complexos, como JSON, arrays, e dados geoespaciais (*PostGIS*).
- **Alta Conformidade com Padrões SQL:** Facilita a portabilidade de aplicações entre diferentes bancos de dados.
- **Extensibilidade:** Suporte a funções definidas pelo usuário, tipos de dados customizados e linguagens de programação adicionais.
- **Desempenho e Escalabilidade:** Otimizações para consultas complexas e grandes volumes de dados, com suporte a replicação e particionamento.
- **Segurança:** Autenticação robusta, criptografia e controle granular de permissões.
- **Comunidade Ativa:** Grande quantidade de documentação e suporte da comunidade global.

3 Metodologia

3.1 Extreme Programming (XP)

Extreme Programming é uma metodologia ágil que valoriza a comunicação, simplicidade, feedback, respeito e coragem, com o objetivo de melhorar e desenvolver software enquanto prioriza as pessoas envolvidas no projeto (<http://www.extremeprogramming.org/>). Diante de uma equipe enxuta composta por apenas dois desenvolvedores, foram necessárias adaptações na metodologia para que ela atendesse às necessidades específicas do projeto.

3.1.1 Particularidades do XP no Projeto

Com uma pequena equipe, algumas práticas do XP foram mais valorizadas e amplamente utilizadas pela dupla idealizadora do trabalho. Entre elas, destaca-se a programação por pares, onde periodicamente os desenvolvedores realizaram o trabalho em uma chamada online, realizando a revisão do trabalho de forma cruzada quando isto não ocorria.

3.2 Kanban

O Kanban é uma metodologia ágil que tem como foco a visualização do fluxo de trabalho. A ideal central deste método de gestão é otimizar a entrega de valor otimizando e limitando o trabalho em progresso. (Ref)

3.2.1 Trello

O Trello é uma ferramenta de gerenciamento de projetos que tem por finalidade organizar as tarefas em um quadro visual. Ele é amplamente utilizado de forma conjunta com o método Kanban, dada sua interface intuitiva para manipulação de cartões e monitoramento do fluxo de trabalho.

3.2.2 Uso do Kanban com Trello

Parte I

Texto e Pós Texto

4 Elementos do Texto

4.1 Corpo do Texto

O estilo de redação deve atentar a boa prática da linguagem técnica. Para a terminologia metrological usar o Vocabulário Internacional de Termos Fundamentais e Gerais de Metrologia(??) (Instituto Nacional de Metrologia, 2003).

Grandezas dimensionais devem ser apresentadas em unidades consistentes com o Sistema Internacional de Unidades (SI). Outras unidades podem ser usadas como unidades secundárias entre parênteses se necessário. Exceções são relacionadas a unidades não-SI usadas como identificadores comerciais como pro exemplo “disquete de 3¹/₂ polegadas”.

Na apresentação de números ao longo do texto usar vírgula para separar a parte decimal de um número. Resultados experimentais devem ser apresentados com sua respectiva incerteza de medição.

4.2 Títulos de capítulos e seções

Recomendações de formatação de seções

1 SEÇÃO PRIMÁRIA - MAIÚSCULAS; NEGRITO; TAMANHO 12;

1.1 SEÇÃO SECUNDÁRIA – MAIÚSCULAS; NORMAL; TAMANHO 12;

1.1.1 Seção terciária - Minúsculas, com exceção da primeira letra; negrito; tamanho 12;

1.1.1.1 Seção quaternária - Minúsculas, com exceção da primeira letra; normal tamanho 12;

1.1.1.1.1 Seção quinária - Minúsculas, com exceção da primeira letra; itálico; tamanho 12.

4.3 Notas de rodapé

Notas eventualmente necessárias devem ser numeradas de forma seqüencial ao longo do texto no formato 1, 2, 3... sendo posicionadas no rodapé de cada página na qual a nota é utilizada.¹

¹ Como, por exemplo, esta nota

4.4 Equações

Equações matemáticas devem ser numeradas sequencialmente e alinhadas a esquerda com recuo de 0,6 cm. Usar numerais arábicos entre parênteses, alinhado a direita, no formato Times New Roman de 9 pts. para numerar as equações como mostrado na Eq. (4.1).

Referências a equações no corpo do texto devem ser feitas como “Eq. (4.1)” quando no meio de uma frase ou como “Equação (4.1)” quando no início de uma sentença. Um espaçamento de 11 pontos deve ser deixado acima, abaixo e entre equações subsequentes. Para uma apresentação compacta das equações deve-se usar os símbolos e expressões matemáticos mais adequados e parênteses para evitar ambigüidades em denominadores. Os símbolos usados nas equações citados no texto devem apresentar exatamente a mesma formatação usada nas equações.

$$\frac{d\mathbf{C}}{dw} = \frac{du}{dw} \cdot \mathbf{F}_u + \frac{dv}{dw} \cdot \mathbf{F}_v \quad (4.1)$$

O significado de todos os símbolos mostrados nas equações deve ser apresentado na lista de símbolos no início do trabalho, embora, em certas circunstâncias o autor possa para maior clareza descrever o significado de certos símbolos no corpo do texto, logo após a equação.

4.5 Figuras e Gráficos

As figuras devem ser centradas entre margens e identificadas por uma legenda alinhada a esquerda com recuo especial de deslocamento de 1,8 cm, com mostrado na Fig. (4.5). O tamanho das fontes empregadas nos rótulos e anotações usadas nas figuras deve ser compatível com o usado no corpo do texto. Rótulos e anotações devem estar em português, com todas as grandezas mostradas em unidades do SI (Sistema Internacional de unidades).

Todas as figuras, gráficos e fotografias devem ser numeradas e referidas no corpo do texto adotando uma numeração sequencial de identificação. As figuras e gráficos devem ser claras e com qualidade adequada para eventual reprodução posterior tanto em cores quanto em preto-e-branco.

As abscissas e ordenadas de todos os gráficos devem ser rotuladas com seus respectivos títulos em português seguida da unidade no SI que caracteriza a grandeza entre colchetes.

A referência explícita no texto à uma figura deve ser feita como “Fig. (4.5)” quando no meio de uma frase ou como “Figura (4.5)” quando no início da mesma. Referências implícitas a uma dada figura devem ser feitas entre parênteses como (Fig. 4.5). Para

referências a mais de uma figura as mesmas regras devem ser aplicadas usando-se o plural adequadamente. Exemplos:

- “Após os ensaios experimentais, foram obtidos os resultados mostrados na Fig. (4.5), que ...”
- “A Figura (4.5) apresenta os resultados obtidos, onde pode-se observar que ...”
- “As Figuras (1) a (3) apresentam os resultados obtidos, ...”
- “Verificou-se uma forte dependência entre as variáveis citadas (Fig. 4.5), comprovando ...”

Cada figura deve ser posicionada o mais próxima possível da primeira citação feita à mesma no texto, imediatamente após o parágrafo no qual é feita tal citação, se possível, na mesma página.

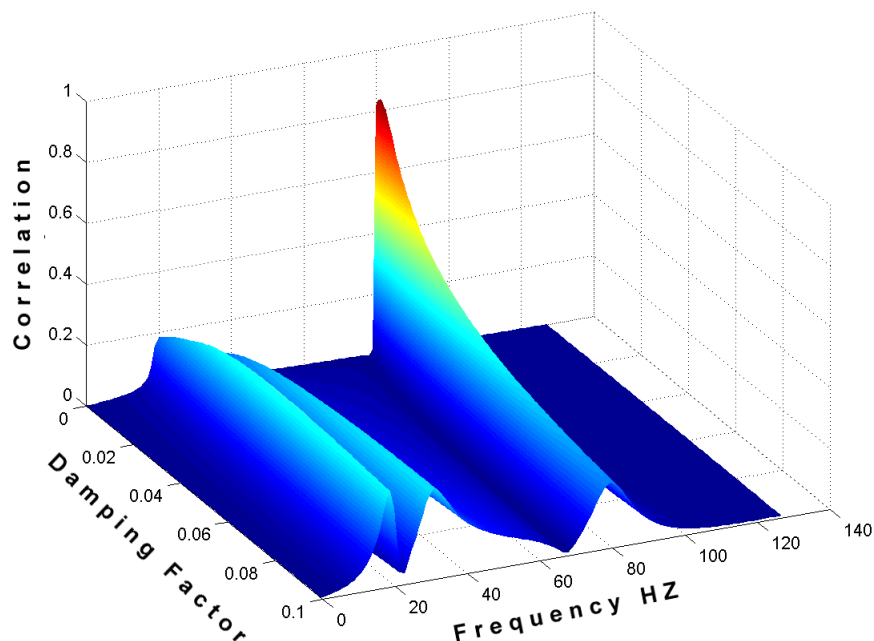


Figura 1 – Wavelets correlation coefficients

4.6 Tabela

As tabelas devem estar centradas entre margens e identificadas por uma legenda alinhada a esquerda, com recuo especial de deslocamento de 1,8 cm, posicionada acima da tabela com mostrado nas Tabs. (4.6) e (2), a título de exemplo. O tamanho das fontes

empregadas nos rótulos e anotações usadas nas tabelas deve ser compatível com o usado no corpo do texto. Rótulos e anotações devem estar em português. Um espaçamento de 11 pts deve ser deixado entre a legenda e a tabela, bem como após a tabela.

As grandezas dimensionais mostradas em cada tabela devem apresentar unidades consistentes com o SI. As unidades de cada variável devem ser mostradas apenas na primeira linha e/ou coluna da tabela, entre colchetes

A referência explícita no texto à uma dada tabela deve ser feita como “Tab. (4.6)” quando no meio de uma frase ou como “Tabela (4.6)” quando no início da mesma. Referências implícitas a uma dada tabela devem ser feitas entre parênteses como “(Tab. 4.6). Para referências a mais de uma tabela as mesmas regras devem ser aplicadas usando-se o plural adequadamente. Exemplos:

- “Após os ensaios experimentais, foram obtidos os resultados mostrados na Tab. (4.6), que ...”
- “A Tabela (4.6) apresenta os resultados obtidos, onde pode-se observar que ...”
- As Tabelas (1) a (3) apresentam os resultados obtidos, ...”
- Verificou-se uma forte dependência entre as variáveis citadas (Tab. 4.6), comprovando ...”

Cada tabela deve ser posicionada o mais próxima possível da primeira citação feita à mesma no texto, imediatamente após o parágrafo no qual é feita a citação, se possível, na mesma página.

Processing type	Property 1 (%)	Property 2 [μm]
Process 1	40.0	22.7
Process 2	48.4	13.9
Process 3	39.0	22.5
Process 4	45.3	28.5

Tabela 1 – Propriedades obtidas após processamento

4.7 Citação de Referências

Referências a outros trabalhos tais como artigos, teses, relatórios, etc. devem ser feitas no corpo do texto devem estar de acordo com a norma corrente ABNT NBR 6023:2002 (ABNT, 2000), esta última baseada nas normas ISO 690:1987:

- “(??), mostraram que...”

- “Resultados disponíveis em (??), (??) e (??), mostram que...”

Para referências a trabalhos com até dois autores, deve-se citar o nome de ambos os autores, por exemplo: “(??), mostraram que...”

5 Elementos do Pós-Texto

Este capítulo apresenta instruções gerais sobre a elaboração e formatação dos elementos do pós-texto a serem apresentados em relatórios de Projeto de Graduação. São abordados aspectos relacionados a redação de referências bibliográficas, bibliografia, anexos e contra-capá.

5.1 Referências Bibliográficas

O primeiro elemento do pós-texto, inserido numa nova página, logo após o último capítulo do trabalho, consiste da lista das referências bibliográficas citadas ao longo do texto.

Cada referência na lista deve ser justificada entre margens e redigida no formato Times New Roman com 11pts. Não é necessário introduzir uma linha em branco entre referências sucessivas.

A primeira linha de cada referência deve ser alinhada à esquerda, com as demais linhas da referência deslocadas de 0,5 cm a partir da margem esquerda.

Todas as referências aparecendo na lista da seção “Referências Bibliográficas” devem estar citadas no texto. Da mesma forma o autor deve verificar que não há no corpo do texto citação a referências que por esquecimento não foram incluídas nesta seção.

As referências devem ser listadas em ordem alfabética, de acordo com o último nome do primeiro autor. Alguns exemplos de listagem de referências são apresentados no Anexo I.

Artigos que ainda não tenham sido publicados, mesmo que tenham sido submetidos para publicação, não deverão ser citados. Artigos ainda não publicados mas que já tenham sido aceitos para publicação devem ser citados como “in press”.

A norma (ABNT), que regulamenta toda a formatação a ser usada na elaboração de referências a diferentes tipos de fontes de consulta, deve ser rigidamente observada. Sugere-se a consulta do trabalho realizado por (ABNT), disponível na internet.

5.2 Anexos

As informações citadas ao longo do texto como “Anexos” devem ser apresentadas numa seção isolada ao término do trabalho, após a seção de referências bibliográficas. Os anexos devem ser numerados seqüencialmente em algarismos romanos maiúsculos (I,

II, III, ...). A primeira página dos anexos deve apresentar um índice conforme modelo apresentado no Anexo I, descrevendo cada anexo e a página inicial do mesmo.

A referência explícita no texto à um dado anexo deve ser feita como “Anexo 1”. Referências implícitas a um dado anexo devem ser feitas entre parênteses como (Anexo I). Para referências a mais de um anexo as mesmas regras devem ser aplicadas usando-se o plural adequadamente. Exemplos:

- “Os resultados detalhados dos ensaios experimentais são apresentados no Anexo IV, onde ...”
- “O Anexo I apresenta os resultados obtidos, onde pode-se observar que ...”
- “Os Anexos I a IV apresentam os resultados obtidos ...”
- “Verificou-se uma forte dependência entre as variáveis citadas (Anexo V), comprovando ...”

Referências

COULOURIS, G. et al. *Distributed Systems: Concepts and Design*. 5th. ed. [S.l.]: Pearson, 2011. Citado 2 vezes nas páginas 30 e 31.

FRAMEWORK, N. *A Progressive Node.js Framework*. 2025. <<https://nestjs.com/>>. Acesso em: 12 jan. 2025. Citado na página 32.

LABS, P. *IPFS - The Distributed Web*. 2025. <<https://ipfs.tech>>. Accessed: 2025-01-11. Citado na página 29.

LOGICMONITOR. *New Study Reveals That 87% of Enterprises Will Accelerate Their Cloud Migration in a Post-COVID World*. 2023. <<https://www.logicmonitor.com/press/new-study-reveals-that-87-of-enterprises-will-accelerate-their-cloud-migration-in-a-post-covid-world>>. Accessed: 2025-01-11. Citado na página 29.

NETWORK, L. *Lightning Network: Scalable, Instant Bitcoin/Blockchain Transactions*. <<https://lightning.network>>. Accessed: 2025-01-11. Citado na página 29.

WANNER, J. *Distributed Applications with IPFS*. 2024. <<https://umm-csci.github.io/senior-seminar/seminars/spring2024/wanner.pdf>>. Accessed: 2025-01-11. Citado na página 29.

Apêndices

APÊNDICE A – Primeiro Apêndice

Texto do primeiro apêndice.

APÊNDICE B – Segundo Apêndice

Texto do segundo apêndice.

Anexos

ANEXO A – Primeiro Anexo

Texto do primeiro anexo.

ANEXO B – Segundo Anexo

Texto do segundo anexo.