

COMPLEXO ESCOLAR LOIDE LAURA

“Privado”

CURSO TÉCNICO DE INFORMÁTICA

TRABALHO FINAL DO CURSO

**PROPOSTA DE DESENVOLVIMENTO DE PLATAFORMA DE CURSO ONLINE**

**Autores:João António Dombele, André Da Silva Barros, Nerilson Gamboa, Nzumba Mafimpa João, Mansaimbundu Suzana.**

**O orientador:**



COMPLEXO ESCOLAR LOIDE LAURA

“Privado”

CURSO TÉCNICO DE INFORMÁTICA

TRABALHO FINAL DO CURSO

13ª CLASSE

**PROPOSTA DE DESENVOLVIMENTO DE PLATAFORMA DE CURSO ONLINE**

**Autores:João António Dombele, André Da Silva Barros, Nerilson Gamboa, Nzumba Mafimpa João, Mansaimbundu Suzana.**

**O orientador:**

**PROPOSTA DE DESENVOLVIMENTO DE PLATAFORMA DE CURSO ONLINE**

Data de Aprovação\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_

Banca Examinadora

Presidente do Júri

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

Primeiro Vogal

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

Segundo Vogal

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

Secretário

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

# DEDICATÓRIA

Dedicamos este trabalho de conclusão de curso ao Colégio Loide Laura, especificamente ao Departamento de Ensino médio Técnico Informática que durante os quatro (4) anos deu-nos a oportunidade de formação nas áreas das tecnologias de informação e especialmente ao Professores, por terem mostrado comprometimento e incentivo para o alcance deste objectivo. A os nossos Encarregados de Educação, colegas e amigos que muito nos apoiaram de forma incondicional no período de

elaboração deste projecto.

# AGRADECIMENTO

Primeiramente agradecemos os nossos familiares que sempre nos apoiaram no

Crescimento escolar. Aos professores e amigos que com suas competências contribuíram e muito No decorrer do desenvolvimento deste trabalho.

Agradecemos especialmente ao orientadores: Eurico Candambo Ângelo, Walter, pelas ideias e tempo disposto para avaliação e evolução no decorrer da confecção deste trabalho.

# RESUMO

# O desenvolvimento de uma plataforma de curso online para um projeto de fim de curso pode ser uma iniciativa estratégica, dada a crescente demanda por educação digital. A plataforma seria um ambiente virtual que não apenas oferece uma diversidade de cursos, mas também proporciona flexibilidade aos estudantes, permitindo-lhes acessar o conteúdo educacional de qualquer localidade com uma conexão à internet.

# Dentro dessa plataforma, é essencial incorporar funcionalidades que facilitem a interação entre alunos e instrutores, promovendo discussões, esclarecimento de dúvidas e colaboração. A inclusão de recursos como fóruns, salas de chat e sessões ao vivo pode enriquecer a experiência de aprendizado, aproximando-a da dinâmica presencial.

# A gama de cursos oferecidos deve ser abrangente, cobrindo desde disciplinas acadêmicas fundamentais, como matemática e ciências, até áreas mais especializadas, como idiomas, artes e negócios. Isso permitirá que a plataforma atenda a diversos públicos e necessidades educacionais.

# A facilidade de gerenciamento do curso para instrutores e a simplicidade de navegação para os alunos são fatores cruciais. Um sistema intuitivo para inscrição em cursos, acesso a materiais de estudo, realização de tarefas e exames contribuirá para uma experiência eficiente e positiva.

# Alem disso, é importante incorporar elementos de avaliação e certificação sólidos. Os alunos devem ter a oportunidade de demonstrar seu conhecimento por meio de avaliações práticas e exames, recebendo certificados de conclusão que tenham reconhecimento no mercado. A relevância dos cursos online tem crescido significativamente, como indicado pela pesquisa da Coursera, que destaca a ênfase em conhecimento prático e habilidades diárias de desenvolvimento em comparação com a educação tradicional. Esse reconhecimento está levando os recrutadores a valorizarem cada vez mais os certificados obtidos por meio de cursos online, o que beneficia os profissionais que buscam novas oportunidades no mercado de trabalho.

# Assim, ao desenvolver uma plataforma de curso online, é fundamental considerar não apenas a diversidade e qualidade dos cursos oferecidos, mas também a experiência do usuário, a interação entre alunos e instrutores, e a validação e reconhecimento efetivo dos certificados, contribuindo para o sucesso e credibilidade da plataforma no cenário educacional e profissional.

# ABSTRACT

Developing an online course platform for a final year project can be a strategic initiative, given the growing demand for digital education. The platform would be a virtual environment that not only offers a diversity of courses, but also provides flexibility to students, allowing them to access educational content from any location with an internet connection.

Within this platform, it is essential to incorporate features that facilitate interaction between students and instructors, promoting discussions, clarifying doubts and collaboration. The inclusion of resources such as forums, chat rooms and live sessions can enrich the learning experience, bringing it closer to in-person dynamics.

The range of courses offered should be comprehensive, covering everything from fundamental academic subjects such as mathematics and science to more specialized areas such as languages, arts and business. This will allow the platform to serve diverse audiences and educational needs.

Ease of course management for instructors and simple navigation for students are crucial factors. An intuitive system for registering for courses, accessing study materials, completing assignments and exams will contribute to an efficient and positive experience.

Additionally, it is important to incorporate solid assessment and certification elements. Students must have the opportunity to demonstrate their knowledge through practical assessments and exams, receiving certificates of completion that are recognized in the market. The relevance of online courses has grown significantly, as indicated by Coursera research, which highlights the emphasis on practical knowledge and daily skills development compared to traditional education. This recognition is leading recruiters to increasingly value certificates obtained through online courses, which benefits professionals seeking new opportunities in the job market.

Therefore, when developing an online course platform, it is essential to consider not only the diversity and quality of the courses offered, but also the user experience, the interaction between students and instructors, and the validation and effective recognition of certificates, contributing to the success and credibility of the platform in the educational and professional scenario.

# LISTA DE FIGURA

Figura 7 - Componentes de interface utilizando framework Bootstrap ...................37

Figura 13 - Exportar Banco de dados para Arquivo SQL. ......................................48

Figura 18 - Itens de menu Cadastros para usuários Administradores. ..................53

Figura 19 - Itens de menu Cadastros para usuários Analistas. ..............................53

Figura 20 - Itens de menu Cadastros para usuários Leitores. ...............................53

Figura 21 - Tela de cadastro e manutenção de perfil de usuário. ..........................54

Figura 22 - Interface de Mensagens Recebidas em ordem de recebimento. .........54

Figura 23 - Interface de Mensagens Enviadas em ordem de envio. ......................55

Figura 24 - Interface de listagem de reservas. .......................................................55

Figura 25 - Encerrando sessão. .............................................................................56

Figura 26 – Interface de listagem de usuários........................................................56

# ÍNDICE DE TABELAS

Quadro 1 - Requisito Software web: Manter Perfil de Usuário ...............................38

Quadro 2 - Requisito Software web web: Manter Mensagens Recebidas .............38

Quadro 3 - Requisito Software web: Manter Mensagens Enviadas .......................39

Quadro 4 - Requisito Software web web: Apresentar Itens Reservados............... 39

Quadro 5 - Requisito Software web web: Manter Cadastro de Usuários ..............39

Quadro 6 - Requisito Software web web: Manter Cadastro de Atores ...................39

Quadro 7 - Requisito Software web web: Manter Cadastro de Projetos ............... 40

Quadro 8 - Requisito Software web web: Manter Cadastro de Pastas ..................40

Quadro 9 - Requisito Software web: Manter Cadastro de Documentos ................41

Quadro 10 - Requisito Software web web: Aplicar ferramenta Bootstra.................41

Quadro 11 - Requisito Software web web: Aplicar linguagens para multiplataform.41

# LISTA DE ABREVIATURA E SIGLAS LISTA DE ABREVIATURA E SIGLAS

CSS *Cascading Style Sheets*

HTML *Hypertext Markup Language*

JS *JavaScript*

PHP *Hypertext Preprocessor*

SQL *Structured Query Language*

XAMP, Apache, MySQL, PHP

Sumário

[DEDICATÓRIA 4](#_Toc158678416)

[AGRADECIMENTO 5](#_Toc158678417)

[RESUMO 6](#_Toc158678418)

[ABSTRACT 7](#_Toc158678419)

[LISTA DE FIGURA 8](#_Toc158678420)

[ÍNDICE DE TABELAS 8](#_Toc158678421)

[LISTA DE ABREVIATURA E SIGLAS LISTA DE ABREVIATURA E SIGLAS 9](#_Toc158678422)

[CAPITULO I. INTRODUÇÃO 12](#_Toc158678423)

[1.1. Justificativa 13](#_Toc158678424)

[**.2.** Formulação do problema 13](#_Toc158678425)

[1.3. Objectivos 14](#_Toc158678426)

[1.3.1. Objectivo geral 14](#_Toc158678427)

[1.3.2. Objectivos específicos 14](#_Toc158678428)

[**1.4.** Hipóteses 14](#_Toc158678429)

[**1.5.** Organização do trabalho 14](#_Toc158678430)

[CAPÍTULO II. ENQUADRAMENTO TEÓRICO 16](#_Toc158678431)

[2.1. Educação a distância 16](#_Toc158678432)

[2.2 TÉCNICAS DE AVALIAÇÃO E DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL. 17](#_Toc158678433)

[2.2.1 DEFINIÇÃO 19](#_Toc158678434)

[2.2.2 FERRAMENTAS DE AVALIAÇÃO 19](#_Toc158678435)

[2.2.4 Modelo Escola Internet 21](#_Toc158678436)

[2.2.5 Avaliação Segundo a Concepção Construtivista de Aprendizagem 24](#_Toc158678437)

[2.3.3 MODELOS DE DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL 26](#_Toc158678438)

[2.3.4 Diretrizes para Interfaces Educacionais 28](#_Toc158678439)

[2.3.5 Paradigma Lições-Construtor-Analisador 30](#_Toc158678440)

[2.4.4 CONCLUSÃO DO CAPÍTULO 34](#_Toc158678441)

[CAPÍTULO III. METODOLOGIA 35](#_Toc158678442)

[3.1. Metodologia de desenvolvimento de software 35](#_Toc158678443)

[3.2.1. Pesquisa bibliográfica 35](#_Toc158678444)

[3.3. Etapas da pesquisa 35](#_Toc158678445)

[3.4. Tecnologias e ferramentas utilizadas 36](#_Toc158678446)

[3.4.1. Ferramentas utilizadas para programação e estilização 36](#_Toc158678447)

[3.4.1.1. Html 36](#_Toc158678448)

[3.4.2. Ferramentas utlizadas para base de dados 37](#_Toc158678449)

[3.7. Análise do Software Web 38](#_Toc158678450)

[3.7.1. Casos de uso 38](#_Toc158678451)

[3.7.2. Actores do Software Web 39](#_Toc158678452)

[3.7.3. Diagramas 40](#_Toc158678453)

[3.7.5. Diagrama de classe 49](#_Toc158678454)

[Figura 3.4: Digrama de classe 49](#_Toc158678455)

[**3.8**. Segurança 50](#_Toc158678456)

[**3.8.2.** Segurança lógica 51](#_Toc158678457)

[CAPÍTULO IV. RESULTADO E DISCUSSÃO 52](#_Toc158678458)

[CAPÍTULO V. CONCLUSÕES E RECOMENDAÇÕES 53](#_Toc158678459)

[**5.1. Conclusão** 53](#_Toc158678460)

[5.2. Recomendações 54](#_Toc158678461)

[REFERÊNCIAS BIBLIOGRÁFICAS 55](#_Toc158678462)

[APÊNDICE 57](#_Toc158678463)

# **CAPITULO I. INTRODUÇÃO**

Na contemporaneidade, a educação é reconhecida como ubíqua, permeando praticamente todos os locais, momentos e situações em que estamos envolvidos, abrangendo grande parte das atividades quotidianas. No entanto, para que ocorra um esforço efetivo de educação e formação, é crucial que a intencionalidade se manifeste por meio da explicitação de conhecimentos e informações, seja por transmissão direta ou por meio de atividades que implicitamente resultem em aprendizagem.

Nesse contexto, as tecnologias digitais emergem como recursos essenciais e inseparáveis de iniciativas educacionais, especialmente nas modalidades de e-learning ou b-learning. O e/b-learning é comummente percebido como uma transição da aula presencial para o ambiente virtual, transformando a voz do professor em uma apresentação disponibilizada na web. O desenvolvimento de iniciativas de educação e formação nesse formato implica um processo extenso e complexo, orientado por critérios de qualidade e eficácia, destacando-se como uma área de conhecimento e desenvolvimento em constante expansão.

O interesse significativo na temática e a perspetiva de aumento nas demandas por formação em e/b-learning e ensino a distância motivaram nossa decisão de realizar um estágio nessa área específica de atividade. Diante das diversas opções de estágio apresentadas, optamos por explorar as oportunidades oferecidas por esse campo em constante evolução.

## **1.1. Justificativa**

A plataforma de Desenvolvimento de Curso Online visa supera barreira geográficas e económicas, proporcionando acesso igualitário á educação para pessoas em qualquer lugar do Mundo. Flexibilidade de aprendizado a educação online oferece flexibilidade de horário, permitindo que os alunos aprendem no seu próprio ritmo. A plataforma oferecerá uma ampla gama de cursos para a tender ás diversas necessidades educacionais, desde habilidades práticas até conhecimento académico avançado.

* Atualização continua da plataforma de curso online:

Permite a constante atualização de conteúdo para refletir as mudanças demandas do mercado de trabalho garantindo que os alunos estejam preparado para os desafios contemporâneos.

* Acesso a Instrutores Especializados:

A possibilidade de interagir com instrutores especializados, muitas vezes localizado em diferentes partes do mundo. Enriquece a experiência de aprendizado. Essa diversidade de perspetivas contribuirá significativamente para minha compressão dos temas abordados no curso.

* + Custo-Benefício:

Ao optar por um curso online, evita despesas com deslocamento e hospedagem, tornando a participação mais acessível financeiramente. Isso permite que nós invistamos em nosso desenvolvimento sem comprometer excessivamente o nosso orçamento.

## **2.** **Formulação do problema**

* Como melhorar plataforma de curso online dos processos a acrescenta demanda por educação online destaca a importância de plataformas eficientes para cursos virtuais. Nesse com texto, identifica-se a necessidade de abordar os seguintes desafios:
* Engajamento do aluno
* Personalização do Conteúdo
* Avaliação de Desempenho
* Acessibilidade e Inclusão

## 

## **1.3. Objectivos**

Nesta secção vamos formular os objectivos levados a cabo para o desenvolvimento do trabalho.

## **1.3.1. Objectivo geral**

* Desenvolver um plataforma de curso online que promova a democratização do acesso á educação, oferecendo uma experiência de aprendizado flexível, personalizada e inclusiva.

## **1.3.2. Objectivos específicos**

* Torna a plataforma acessível em diferentes dispositivos e conexões,Permitindo que alunos de diversas regiões geográficas participem sem restrições.
* Implementar estratégias interativas, como fórum de discussão, quizzes e atividades práticas, para promover o engajamento e a participação ativa do alunos.
* Desenvolver algoritmos de personalização de conteúdo, levado em consideração o histórico de aprendizado, preferências e ritmo individual de cada aluno.

## **1.4.** **Hipóteses**

* Diminuir o tempo de espera dos usuários;
* Aumento do Engajamento;
* Plataforma de curso online vai gerir os processos com maior eficiência e eficácia;
* Plataforma de curso online vai melhorar o controlo do sistema;
* Melhoria da retenção do Conhecimento;
* Feedback Construtivo para Aprimoramento Continuo.

## 

## **1.5.** **Organização do trabalho**

Este trabalho está organizado em cinco capítulos: no capítulo I em que estamos, é feita de forma resumida breves considerações sobre o tema, a justificativa; formulação do problema e seus principais objectivos. No capítulo II, é apresentado os conceitos necessários, enquadramento teórico, nomeadamente: Educação a distância, Diferenças entre a educação presencial e a educação a distância, definição, ferramenta de avaliação, entre outros. O capítulo III**,** aborda sobre a metodologia de pesquisa adotada para o trabalho, bem como, o processo de desenvolvimento de software utilizado para realização do sistema. No capítulo 4, faz referência aos resultados e as discussões da análise dos dados obtidos. Finalmente no capítulo 5 são apresentadas as considerações finais resultante a partir da análise dos resultados e as recomendações futuras.

# **CAPÍTULO II. ENQUADRAMENTO TEÓRICO**

**1 Conceito Fundamentais.**

A base do nosso trabalho em estágio foi a construção de cursos em *e-learning* o que nos remete para a Educação a Distância (EAD) ou educação *online*. Como tal, importa começar por caracterizar a noção de educação assumida neste relatório.

A educação é um conceito que tem múltiplas definições, mas todas têm um aspeto em comum ou seja, “colocam o indivíduo como sujeito no centro da atividade e caracterizam a educação como um processo de influência sobre as pessoas que conduz à

Sua transformação e a capacita para interagir como o meio.

Ainda segundo Calleja(2008) a educação “é ação que desenvolvemos sobre as pessoas que formam a sociedade, com o fim de capacitá-las de maneira integral, consciente, eficiente e eficaz, que lhes permita formar um valor dos conteúdos adquiridos, significando-os em vínculo direto com seu quotidiano, para atuar consequentemente a partir do processo educativo assimilado. No entanto, existe uma tendência geral nas pessoas para assumir a educação como atividade realizar na escola. Por exemplo, para Machado (s.d.), a educação “refere-se ao trabalho que se desenvolve no contexto das unidades educacionais que conhecemos.

Nos dias de hoje a educação é cada vez mais considerada ubíqua, isto é, acontece

em praticamente todos os locais, tempos e situações em que estivermos e em todas as

Atividades em que estamos envolvidos. A educação que é concretizada numa sala de

aula com um professor centrado na transmissão unidirecional de conteúdos temáticos

perante uma turma de alunos a assistir, pode ser facilmente realizada através da

gravação em vídeo e transmissão mediada pelas tecnologias digitais e da internet.

## **2.1. Educação a distância**

No que respeita à variável espaço e tempo, podemos considerar três tipos de educação: presencial, semi presencial e a distância. A educação presencial é a educação realizada na sua forma habitual expositiva, centrada no professor ou seja, tanto o professor como os alunos se encontram no mesmo espaço físico designado de sala de aula. A educação a distância é a educação em que o professor e os alunos não se encontram no mesmo espaço físico, podendo ou não estar ao mesmo tempo em comunicação através das tecnologias (comunicação síncrona). A educação semi presencial é a articulação da educação presencial com a educação a distância, ou seja, neste tipo de educação a comunicação é realizada tanto em sala de aula como a distância através das tecnologias.

De forma a perceber as diferenças entre educação presencial e a educação a distância, construímos as seguintes tabelas (adaptada de Costa, Schaurich, Stefanan, Sales & Richter, 2014).

Tabela 1: Diferenças entre a educação presencial e a educação a distância

|  |  |
| --- | --- |
| Educação presencial | Educação a Distância |
| - Tempo limitado; | - Flexibilidade do tempo; |
| - Centrado no professor; | - Centrado no aluno; |
| - O aluno está dependente do professor e  do programa curricular; | - O aluno estuda ao seu ritmo; |
| - Comunicação síncrona. | Comunicação síncrona e assíncrona |

# **2.2 TÉCNICAS DE AVALIAÇÃO E DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL.**

A diferenciação de um software tradicional, de um software com características

Educacionais, é uma questão problemática, genérica e ampla, bastante complexa e

Controversa. Quando é que um software torna-se educacional? O software

tecnologicamente avançado, com som, animação, cores, profundidade de telas e uma

grande variedade de recursos com conteúdo apenas informativo, é um software

educacional? Muitos autores propõem a divisão em classes e subclasses, para facilitar a

análise e a avaliação de softwares para uso educacional. Na prática, os produtos não

funcionam isoladamente, atualmente os sistemas costumam ser híbridos.

A qualidade de software educacional deve ser determinada pelas teorias de aprendizagem, que distinguem ambientes educacionais, mais ou menos interativos, com

maior ou menor grau de participação e controle do aluno no processo de construção do

conhecimento. O desenvolvimento de um software educacional, portanto, deve

contemplar as características da educação, que levam à formação global do aluno, que

necessita aprender a aprender e a pensar para melhor intervir, inovar e questionar, deve

trabalhar com as funções de cognição do aluno.

Um critério para classificação e avaliação de software educacional bastante

valorizado é o de grau de interatividade conferido a ele. Nesse aspecto, a ergonomia de

interface homem-computador fornece a base teórica e metodológica para estudar as

Dificuldades de relacionamento homem-máquina. A ergonomia utiliza métodos e

técnicas para observar o trabalho humano, utilizando estratégias para apreender a

Complexidade do trabalho e decompor a atividade em indicadores observáveis, como a

Postura, exploração visual e o deslocamento. Como em todo processo científico de

Investigação, a espinha dorsal de uma intervenção ergonômica é a formulação de

Hipóteses. Segundo LEPLAT (in CAMPOS, 2001), o pesquisador trabalha, em geral, a

Partir de uma hipótese, é isso que lhe permite ordenar os fatos.

Cabe aqui analisar como a informática vem sendo utilizada no processo de

ensino-aprendizagem, como a tecnologia interfere no processo cognitivo; quais os

critérios estabelecidos para as aplicações serem consideradas educacionais; quais os

seus pressupostos pedagógicos; cumpre os critérios da *ecologia cognitiva*?

O termo ecologia cognitiva foi cunhado recentemente para indicar a importância que as tecnologias têm no processo cognitivo. Para Lévy (1993), as Sociedades orais dependem apenas do que pode ser inscrito na mente, tendo quase todo o seu edifício cultural fundado sobre as lembranças, concluindo-se que a inteligência se Associa à memória, estando mais embasada na intuição e compreensão do que na explicação. Para Lévy (1993), as formas de representação que mais chance têm de Sobreviver à memória humana, nessa sociedade, são os mitos, as narrações, as danças, As rimas e os cantos, usando as melhores estratégias de codificação culturais.

No entanto, diz que sem escrita não há datas, nem arquivos, nem códigos

Legislativos, nem sistemas filosóficos, nem diagramas, concluindo que a palavra escrita

é o elemento constitutivo indispensável ao pensamento humano.

A tecnologia por si só não é um instrumento para resolver todos os males, mas

auxilia no processo de cognição, usando tanto a escrita como a oralidade das soluções,

criando novos ambientes, com grandes possibilidades de serem explorados; a esses

ambientes chamamos de ecologia cognitiva, em que convivem as tecnologias de

informação, com as engenharias cognitivas e pedagógicas e principalmente com o

elemento humano e suas interações.

# **2.2.1 DEFINIÇÃO**

Quando um software é considerado educacional? Para Chaves (2001), quando é usado para um objetivo educacional ou pedagogicamente defensável, qualquer que seja a natureza ou finalidade para a qual tenha sido criado.

Para Campos (2001), um software educacional contempla as características da

educação que levam à formação global do aluno, que necessita aprender a aprender e a pensar para melhor intervir, inovar e questionar, trabalhando com as funções da

cognição.

# **2.2.2 FERRAMENTAS DE AVALIAÇÃO**

**2.2.3 TICESE**

Técnica de Inspeção de Conformidade Ergonômica de Software Educacional-

TICESE, elaborada a partir da revisão dos vários modelos de avaliação de software

Propõe a integração entre os aspectos pedagógicos e de usabilidade no processo de

Avaliação ergonômica do software educacional.

A técnica é composta por um conjunto de critérios e subcritérios de inspeção,

Baseados em recomendações ergonômicas para o serviço de avaliação. É formada por 3

Seções, formando o Manual do Avaliador: a seção 1 contém a Descrição e Justificativas

Dos Critérios, a seção 2, as diretrizes para o Tratamento Quantitativo da Informação e a

Seção 3, o Formulário de Inspeção Ergonômica de Software Educacional,

Complementados por dois módulos introdutórios: o Módulo de Classificação, com o

Objetivo de classificar, em uma proposta pedagógica, dentro de atributos específicos de

Cada categoria de software, e o módulo de avaliação que constitui o módulo principal da técnica em que se avalia a conformidade com os padrões ergonômicos de qualidade de software. Dentre os critérios de apontados estão:

a) Na avaliação da documentação: dados de identificação, qualidade da

informação impressa, presteza, legibilidade, densidade informacional,

consistência e significado dos códigos e denominações;

b) Na avaliação do produto: condução, presteza, qualidade da opção de ajuda,

legibilidade, feedback imediato, agrupamento e distinção de itens por localização e por formato, adaptabilidade, observando a flexibilidade e consideração da experiência do usuário, controle explícito, recursos de apoio à compreensão dos conteúdos, gestão de erros, avaliação da aprendizagem, carga de trabalho, significado dos códigos e denominações, homogeneidade e compatibilidade.

No módulo específico denominado Módulo de Avaliação Contextual, verificase

a adequabilidade do software em um dado contexto pedagógico ou situação

Específica. Sendo um módulo complementar na avaliação, auxiliando no processo de

Tomada de decisão sobre a adoção e implementação em um contexto particular de Ensino.

A seção dois, em que se trata de propostas de diretrizes para o tratamento quantitativo da informação é composta das seguintes atividades:

1. definição do sistema de valores para tratamento quantitativo dos dados;

b) definição do sistema de ponderação para as respostas ao formulário de

inspeção;

1. definição de pesos a atribuir para cada questão;
2. definição de uma equação matemática para o tratamento da informação;

1. aplicação do sistema de métrica e interpretação dos resultados.

# **2.2.4 Modelo Escola Internet**

Esta técnica foi baseada na coluna de treinamentos on-line do site da Escola

Internet, mantida pela professores, sobre qualidade de software educacional, o qual parte de um artigo anterior desenvolvido e já publicado em parceria com os professores.

Conforme a professores, o software educacional deve ser determinado pelas teorias de aprendizagem, que distinguem ambientes educacionais, mais ou menos interativos, com maior ou menor grau de participação e controle do aluno no processo de construção do conhecimento. Nos ambientes de aprendizagem construtivista, está na autonomia do aluno, que interage com o ambiente, e o foco está no processo de construção do conhecimento e não apenas em um domínio predefinido do conhecimento a ser adquirido.

Um critério muito usado para classificar os tipos de software educacional é o

grau de iniciativa permitido ao aluno ou o grau de direcionamento conferido a ele:

a) alta interatividade: permite a descoberta imprevista e a descoberta com

exploração livre;

b) média interatividade: permite a descoberta guiada;

c) baixa interatividade: privilegia a aprendizagem de recepção direcionada, a

exposição indutiva, a exposição dedutiva. O modelo construtivista de software é caracterizado por uma definição de objetivos macros e os contextos para incentivar a construção do conhecimento com a participação do aluno no processo, a avaliação é qualitativa, considerar a nãolinearidade, a escolha de caminhos de navegação por parte do aprendiz e a liberdade na busca da informação, propor problemas realistas, interessantes e relevantes para os alunos e permite testar diversas soluções e por estimular a colaboração, o diálogo e a negociação no trabalho em grupo.

Para avaliar um software educacional é preciso considerar, além das

características citadas, os atributos inerentes ao domínio e às tecnologias específicas,

bem como as diversas visões das teorias de aprendizagem e cognição.

Os parâmetros mínimos de avaliação referem-se às características pedagógicas ao aspecto técnico da questão:

a) características pedagógicas: formam um conjunto de atributos que

evidenciam a conveniência e a viabilidade de uso em situações

educacionais. Tem como subcaracterísticas: o ambiente educacional

identificando o ambiente e o modelo de aprendizagem, a pertinência em

relação ao programa curricular em que analisa a adequação em relação ao

contexto educacional ou a uma disciplina específica, e também os aspectos

didáticos que analisa a contribuição para que o aluno alcance o objetivo e os

meios motivacionais respeitando a individualidade;

b) facilidade de uso: conjunto de atributos que evidenciam a usabilidade de

software. Inclui as subcaracterísticas como: facilidade de aprendizado,

facilidade de memorização e robustez em que analisa se o software mantém

o processamento corretamente a despeito de ações inesperadas;

c) adaptabilidade: avalia a capacidade do software de adaptar-se às

necessidades e preferências do usuário e ao ambiente educacional

selecionado. Inclui atributos como a customização, em que avalia a

facilidade de adaptação da interface para diferentes usuários e a adequação

ao ambiente que avalia a facilidade de adequação do software aos objetivos

educacionais;

d) documentação: avalia se a documentação está completa, consistente, legível

e organizada. Inclui atributos como *help on line,* em que avalia a existência

de auxílio *on line* e a documentação do usuário em que avalia a documentação sobre o uso do sistema, sobre a instalação e sobre a clareza de compreensão.

A autora destaca as características para ambientes e sites educacionais apoiados

na WEB, incluindo subcaracterísticas como conteúdos corretos, fontes fidedignas, carga

informacional compatível, pertinência, temas transversais, entre outros.

# 

# **2.2.5 Avaliação Segundo a Concepção Construtivista de Aprendizagem**

Essa proposta está baseada na idéia de que o uso do computador na educação

tem o objetivo de promover a aprendizagem dos alunos e ajudar na construção do processo de conceituação e no desenvolvimento de habilidades importantes para que ele

participe da sociedade do conhecimento e não simplesmente facilitar o seu processo de

aprendizagem.

Avaliar significa analisar como um software pode ter um uso educacional,

como pode ajudar o aprendiz a construir seu conhecimento e a modificar sua

compreensão de mundo, elevando sua capacidade de participar da realidade que está

vivendo. Considerando desta forma, uma avaliação criteriosa pode apontar para que tipo

de proposta pedagógica poderá ser mais bem aproveitado.

Classifica os softwares educacionais de acordo com seus objetivos educacionais em: tutoriais, programação, aplicativos, exercícios e práticas, multimídia e internet, simulação e modelagem e jogos. Também, classifica-os quanto ao nível de aprendizagem:

a) seqüencial: a preocupação é transferir a informação; o aprendiz recebe a

informação, memoriza e repete quando solicitado, levando a um aprendiz

passivo;

b) relacional: objetiva a aquisição de algumas habilidades, permitindo que o

aprendiz faça relações com outros fatos ou outras fontes de informação.

Esse nível de aprendizagem leva a um aprendizado isolado;

c) criativo: associado à criação de novos esquemas mentais, possibilita a

interação entre pessoas e tecnologias, compartilhando objetivos comuns.

Esse nível de aprendizado leva a um aprendiz participativo. Além da base pedagógica, avaliam-se também os aspectos técnicos orientandoos para uma adequada utilização. Analisam aspectos de mídias empregadas, qualidade de telas, interface disponíveis, clareza de instruções, compartilhamento de rede local e internet, compatibilização com outros softwares, hardware e funcionalidade em rede (importação e exportação de objetos), apresentação auto-executável, recursos *hipertexto* e *hiperlink*, disponibilidade de *helpdesk*, manual técnico com linguagem apropriada ao professor-usuário, facilidade de instalação, desinstalação e manuseio, entre outros. O modelo utiliza-se de uma ficha de registro contendo:

a) identificação do software: autor, firma desenvolvedora, objetivo, breve

resumo, idioma, duração, preço e o meio de armazenamento;

b) base pedagógica: a concepção teórica da aprendizagem, se construtivista ou

behaviorista com justificativa, a possibilidade do ciclo de Valente, a interação aprendiz-agente de aprendizagem, o *feedback*, o processo de construção do conhecimento do aluno, a formulação e verificação dehipóteses, análise e depuração dos resultados e a interdisciplinaridade;

c) classificação: tutorial, exercícios e prática, programação, aplicativo,

multimídia, sistema de autoria, simulação, modelagem, jogos e se aberto ou

fechado; quanto ao nível de aprendizado, seqüencial, relacional ou criativo;

1. aspectos técnicos: a presença ou não de alguns aspectos já citados acima;

e) conclusões: sobre o processo de avaliação, recomendações, sugestões e

equipe avaliadora.

A autora conclui o seu trabalho afirmando que quem determina as

possibilidades de uso dos softwares na educação são os professores, com suas

concepções sobre o que é ensinar e aprender.

# 

# **2.3.3 MODELOS DE DESENVOLVIMENTO DE SOFTWARE EDUCACIONAL**

O desenvolvimento de um software educacional depende da formação de uma

Equipe multidisciplinar, envolvendo programadores, psicólogos, pedagogos,

Cognitivistas e projetistas de interfaces, isso reflete a conclusão unânime dos modelos

Estudados. Complementa-se com alguns autores admitindo a participação do usuário

Nessa equipe.

A dúvida, entretanto, reside na definição da fronteira entre o que é software e o

que é ambiente educacional. Um site informativo é um software educacional? Uma

Enciclopédia em CD-ROM é um software educacional? Essas reflexões levaram a uma

Classificação em algumas categorias, de acordo com os objetivos pedagógicos e as

Características técnicas de construção: tutoriais, programação, aplicativos, exercícios e

Práticas, multimédia e Internet, simulação e modelagem e jogos.

a) tutoriais: funcionam como um guia, um livro animado, um vídeo interativo

ou um professor eletrónico.

A informação é apresentada seguindo uma sequência deixando a escolha para o usuário. Não há um feedbackda aprendizagem, o controle é do usuário e a interação se resume em avançar ou retroceder dentro da informação apresentada na tela e na sua leitura. É uma versão sofisticada da instrução programada, com recursos de animação, com a vantagem de o usuário programar a sua aprendizagem. Os bons tutoriais utilizam as técnicas de inteligência artificial para analisar padrões de erro, avaliar estilo e a capacidade de aprendizagem do usuário;

b) programação: permite que os usuários criem seus próprios programas, sem

o conhecimento de linguagem ou técnicas de programação. A linguagem

LOGO é um exemplo deste tipo de software. O usuário aprende a aprender,

Utilizando o ambiente para espelhar seus conhecimentos e construir novos

**Objetos:**

c) aplicativos: são considerados os softwares de produtividade, como os

Processadores de texto, planilhas eletrónicas, gestores de banco de

Dados, softwares gráficos. São de uso restrito, mas podem ser explorados

Utilizando-se de macro-comandos embutidos nos pacotes. Destinam-se a uma

Atividade com conteúdo preciso, como esquematizar, classificar objetos ou

Resolução de problemas numéricos;

d) exercícios e prática: são do tipo pergunta e resposta, com a avaliação

Monitorada pelo próprio computador. As atividades exigem apenas o fazer,

O memorizar informação, não analisando a compreensão e nem a avaliação

da aprendizagem. É considerada a forma mais comum de uso do computador na educação. São utilizados para revisar o material visto em sala de aula, especialmente aqueles que exigem memorização e repetição.

São indicados para o próprio aluno avançar no conteúdo ou para auxiliar os

mais atrasados, com dificuldade de acompanhamento;

e) multimídia e internet: o uso de multimídia já pronta e internet são parecidos

Com os tutoriais, com a diferença que o usuário pode interagir, navegando

Com mais facilidade e refletir sobre o conteúdo, mas não oferece

Oportunidade de compreensão e a aplicação da informação recebida. Já os

Sistemas de autoria são sistemas em que o usuário pode desenvolver o

Sistema de multimídia, selecionando as diversas fontes e programas,

Construindo a sua aprendizagem, possibilitando reflexões sobre os

Resultados obtidos, comparando-os e depurando a qualidade, profundidade e

o significado da informação apresentada;

f) simulação e modelagem: implicam a criação de modelos simplificados do

Mundo real e são riquíssimos, pois envolvem modelos de sistemas

Complexos e dinâmicos. As simulações oferecem ao aprendiz a

possibilidade de desenvolver hipóteses, testa-las, analisar resultados e refinar os conteúdos, favorecendo o trabalho em grupo. A interação com o computador é do controle do aprendiz. Esse tipo de software favorece o uso do computador e facilita o trabalho de ensino do professor, pois possibilita que situações difíceis e perigosas sejam reproduzidas em aula, como laboratórios virtuais de química, física, biologia, balística, dissecação de

cadáveres e outros, apenas simulando a realidade virtualmente;

g) Jogos: os jogos pedagógicos têm como alvo a promoção da aprendizagem. A

pedagogia, por trás desta modalidade, é a exploração auto dirigida, em vez da instrução explícita e direta. Há alguns desvios no seu uso, devido à competição, como em qualquer jogo, desviando a atenção em relação ao conceito pedagógico para o objetivo de vencer no jogo. Há também a questão motivacional, após vencido o jogo, a repetição torna-se maçante e o aprendiz perde o interesse.

Segundo Galvis (1999), a qualidade do software não se limita ao cumprimento dos padrões associados ao tipo ou combinações de tipos aos quais pertence. Acima disso, estão os critérios de pertinência, relevância (até onde o software é coerente com os outros elementos do ambiente de aprendizagem?) e unicidade (em que medida são aproveitadas as qualidades únicas do computador como meio?). Assim a qualidade é algo que não só está ligada ao produto, mas sim enraizada em todo o

processo de desenvolvimento.

# 2.3.4 Diretrizes para Interfaces Educacionais

Esse modelo propõe a discussão do projeto de interfaces com o usuário para

ambientes educacionais, analisando-se a literatura sobre a interação homem-máquina e

sobre as interfaces educacionais, com o propósito de definir algumas diretrizes no

desenvolvimento de projeto de interfaces, sob a ótica do projetista e não do usuário

.

As diretrizes que nortearão os projetos de interfaces dos protótipos educacionais são:

a) Utilizar, tanto quanto possível, uma abordagem conceitual: seja apoiada em enfoques teóricos, em aprendizagem baseada em problemas, aprender fazendo, aprender explorando e navegando, para encontrar respostas, em metáforas do mundo real ou em narrativas e histórias;

b) Integrar as tecnologias disponíveis no projeto de interface: a partir da

abordagem conceitual escolhida. Narrativas e histórias podem ser mais bem

representadas em interfaces com componentes hipermídia e com espaços de

comunicação e cooperação. A adoção de uma metáfora do mundo real pode

ser mais bem viabilizada como realidade virtual. Aprender resolvendo

problemas, explorando e navegando, pode ser mais bem resolvido com

interfaces para a Web, pois esses enfoques podem requerer acesso a

múltiplas fontes de consulta;

c) Dar ao usuário a possibilidade de adequar a interface a suas necessidades:

através dos níveis mais simples e fáceis de customização, como a definição

de preferências e a escolha de modelos;

d) Considerar, tanto quanto possível, a questão cultural no projeto da interface:

observando duas vertentes: a primeira, soluções padronizadas, como as

usadas no padrão Windows, podem ser adotadas, já que grande parte dos

usuários de software está familiarizada com a metáfora, com os ícones e

com as formas de trabalho empregadas; a segunda, são soluções baseadas

nas especificidades culturais;

e) Considerar na interface educacional os princípios gerais para projeto de

interfaces, tais como: funcionalidade, confiabilidade, disponibilidade,

segurança, integridade dos dados, padronização, integração, portabilidade,

consistência, integridade estética, estabilidade, estilo de *input see-and-point*,

manipulação direta, uso de *feedbacks* e diálogos;

.

# **2.3.5 Paradigma Lições-Construtor-Analisador**

O modelo proposto está fundamentado na combinação dos elementos das

teorias pedagógicas comportamental e construtiva. Inicialmente projetado e

experimentado na construção do software ASA (Animação e Simulação de Algoritmos)

para o ensino de algoritmos no SENAC; estruturado em três módulos: Lições,

Construtor e Analisador (GUIMARÃES, 2001).

O módulo Lições, semelhante às lições no livro-texto, ou uma aula expositiva

ou uma fita de vídeo, é o primeiro contato do aluno, apresentado com imagens e texto.

As animações são divididas em duas categorias: Conceito e Simulação. Conceitos são

representações concretas de algo abstrato a ser ensinado. Por exemplo: representar a

memória do computador (abstrato) através de gavetas (concreto). Simulações

correspondem à representação mais fiel possível do mundo real. Tanto nos conceitos

como nas animações, o aluno interage com o sistema, controlando a velocidade e o número de vezes que cada animação será executada, fornecendo dados de entrada,

quando solicitado. Por meio das animações, o aluno recebe *feedback* implícito; e, por

meio de exercícios, o *feedback* explícito em qualquer tipo de mídia (texto, som,

imagem, vídeo) e ainda registra a resposta na ficha do aluno.

No módulo Construtor, o aluno aplica o que foi apresentado no módulo Lições.

O aluno cria e executa a simulação, podendo ser criada mais de uma forma de representação. O sistema também envia o *feedback* ao aluno e registra seu trabalho na

Ficha do Aluno.

No Módulo Analisador, o instrutor pode analisar os resultados da interação do

aluno através de um *log* (para um arquivo texto), de todas as atividades do aluno. O

instrutor pode analisar, além dos resultados, os passos intermediários, constituindo uma

forma de acompanhar o desenvolvimento dos alunos. Uma forma de aperfeiçoamento

pode ser trabalhar esse log, classificando os resultados, gerando estatísticas e, baseado

nisso, interagir mais efetivamente com o aluno.

Segundo o autor, o paradigma descrito permite que os desenvolvedores,

independente do conteúdo temático da aplicação, sua natureza ou grau de dificuldade,

lancem mão de recursos de visualização, operem com volumes expressivos de

exercícios (o que implica o uso de simulações gráficas e requer participação ativa e

intensiva dos usuários na sua resolução) e empreguem meios para rastrear o processo de

aprendizagem e construção do conhecimento, inclusive com a possibilidade de gerar e

fornecer registos estatísticos.

· Os construtivistas privilegiam os meta objetivos e as estratégias internas

para a produção do conhecimento. É a pedagogia de projetos;

· Nos ambientes de aprendizagem construtivas, os alunos possuem

muito mais responsabilidades sobre o gerenciamento de suas tarefas e seu

papel no processo é de colaborador ativo;

· A ênfase é centrada no pensamento crítico;

· A progressão da aprendizagem é não-linear;

· O processo educacional é centrado no aluno;

· A interação se faz com o mundo real.

c) solução: definir os macro objetivos e os contextos, para incentivar a construção do conhecimento e incentivar a participação do aluno no processo. A avaliação é qualitativa. Considerar a não-linearidade, a escolha de caminhos navegacionais por parte do estudante e a liberdade na busca da informação. Propor problemas realistas, interessantes e relevantes aos alunos e permitir testar diversas soluções. Estimular a colaboração, o diálogo e a negociação no trabalho em grupo, como forma de encorajar múltiplas interpretações.

O ambiente educacional construtiva deve seguir algumas heurísticas

que irão contribuir para a construção do conhecimento: propor ambientes que permitam produção e conhecimento e a compreensão sob múltiplas perspectivas;

propor problemas contextualizados e compatíveis com o conhecimento externo à sala de aula;

· Permitir interpretação significativa e reflexiva;

· Incentivar o pensamento crítico;

· Encorajar a troca de idéias e testagem das alternativas;

· Fornecer assistência ao aluno, ao contexto da aprendizagem e ao

processo.

Os autores observam a importância do estabelecimento de padrões, como uma

forma de disseminação de soluções obtidas através de muitas práticas, reduzindo a

complexidade da realidade e podendo fornecer ações que levem à solução de problemas.

Ressalta a necessidade da utilização desses padrões, no entendimento, modificação e adaptação às necessidades específicas do usuário, para que se torne uma atividade colaborativa e em que a experiência de um venha a contribuir para o aperfeiçoamento e ampliação deste sistema.

**2.4.2 Método da ULBRA**

Este modelo (SILVEIRA, 2001) descreve o desenvolvimento de software

educacional em forma de *slides* de apresentação. Para o autor, o desenvolvimento de

softwares educacionais com qualidade é necessário à observação de algumas etapas,

desde a concepção até a implantação dos mesmos:

1) definição do ambiente de aprendizagem: seguindo a filosofia de

aprendizagem e o modelo de ensino-aprendizagem;

2) análise da viabilidade: definição de hardware, o cálculo dos custos e

cronogramas;

3) seleção do tipo de documento: documentos que podem ser utilizados por

diversos usuários (base de conhecimento sólida e consistente) ou produtos

que não têm compromisso didático-pedagógico (exploratório);

4) seleção do método de autoria:

· Linguagens de programação: necessidade de incluir um programador na

equipe de desenvolvimento;

· Linguagens de autoria: possuem rotinas específicas para o desenvolvimento de softwares em multimídia, com recursos como editor de texto, editor gráfico, inserção de sons e imagens, entre outros. São exemplos os softwares Multimedia Toolbook, Macro media Director, Visual Class e Everest;

·Sistemas de autoria: são ambientes em que o usuário pode criar um

software multimédia, sem ter que dispor de um conhecimento aprofundado de programação. Com um conhecimento básico é possível desenvolver softwares, tais como: apostilas eletrónicas, exercícios de fixação e tutoriais.

5) Planeamento da interface:

6) Planeamento do documento:

7) Seleção do sistema de autoria e das ferramentas,

**2.4.3 Modelo de Ciclo de Vida**

O modelo foi publicado e disponibilizado na Internet, desenvolvido, traça uma visão geral sobre as teorias de aprendizagem, esse modelo e noções básicas sobre software educacional, culminando na descrição do modelo proposto (SANTOS, 2000).

A proposta segue o modelo do ciclo de vida abordado pela Engenharia de software clássica compreendendo as seguintes fases:

a) Análise: fase em que se analisa qual é a melhor solução para o problema,

definindo o ambiente educacional inclusive;

b) Projeto: fase em que se define o plano de desenvolvimento do software

(especificação, design, diretrizes de interface). Nesta fase, é sempre conveniente usar algum modelo/método para suporte da modelagem;

c) Codificação: fase da escolha da plataforma de hardware e software em que o

software será implementado e a própria implementação;

d) Avaliação: definição dos critérios de avaliação e marcos de avaliação do

processo de desenvolvimento e do produto;

1. Manutenção: implantação e controle das versões do software.

O ponto crítico dessa proposta está na definição do ambiente de aprendizagem

e na modelagem da aplicação. Nessa etapa, deve-se garantir a definição do grau de

interatividade do usuário com o software, o atingimento dos objetivos educacionais e

sua adequação ao público-alvo e o respeito às características do ambiente de

aprendizagem escolhido.

a) Padrões conceituais: são padrões em que as formas são descritas por meio de

termos e conceitos de um domínio de aplicação, são baseados em metáforas

e restritos a um domínio de aplicação;

b) Padrões de projeto: são padrões cujas formas são descritas por meio de

construções de projetos de software, por exemplo, objetos, classes, herança,

agregação e uso-relacionamento. Padrões de projeto complementam, ou são

construídos sobre padrões conceituais;

c) Padrões de programação: são padrões cujas formas são descritas por meio de

construções de linguagens de programação. Padrões de programação

implementam padrões de projeto, utilizando uma linguagem de programação específica.

Os padrões disponibilizam para muitos um conhecimento adquirido, validado e

medido, porém, não especificam uma solução detalhada. Ajudam na redução da

complexidade de muitas situações da vida real.

# **2.4.4 CONCLUSÃO DO CAPÍTULO**

Para avaliar um software educacional, temos que considerar além das características inerentes ao produto, os atributos de domínio e as tecnologias específicas. Sabe-se que as teorias de aprendizagem refletem visões profundamente diferentes sobre como ocorre o aprendizado e essas visões têm impacto nos modelos de software educacional. Acredita-se que a avaliação começa pela identificação do seu ambiente, ou seu potencial uso para um determinado ambiente educacional.

# **CAPÍTULO III. METODOLOGIA**

Na organização Para a resolução dos problemas encontrados primeiramente foram levantados alguns requisitos para a criação de um novo software e como o mesmo poderia ser utilizado no dia-a-dia, em sequência foi projetado uma ideia de solução, e que por final fora executado e aplicado ação.

As metodologias utilizadas consistem em coletar e analisar informações sobre

uma determinada organização, a fim de estudar aspectos relacionados ao tema de

Plataforma de curso online de especificação de software.

# **3.1. Metodologia de desenvolvimento de software**

Um projeto de software é um esforço no sentido de construir um produto,

Dentro de determinadas especificações, que atenda às necessidades dos usuários para

que executem processos operacionais e gerenciais de negócios. É a junção de Objetivos

+ Atividades + Prazos + Recursos Envolvidos + Riscos e Incertezas, conforme.

**3.2. Metodologia de investigação científica**

A metodologia científica adotada para este trabalho é a indutiva. Sendo este, um método que considera casos particulares para chegar a uma determinada conclusão, que é o caso geral.

# **3.2.1. Pesquisa bibliográfica**

Partindo do ponto de que a pesquisa feita partiu de material publicado por outros autores, como livros, artigos e outros material encontrados na Internet.

# **3.3. Etapas da pesquisa**

Todo o projeto começou com a recolha de dados, de forma a conhecer como funciona os serviços prestados por outras plataformas reconhecidas a nível mundial no mercado, e outras fontes (sites de internet).

# **3.4. Tecnologias e ferramentas utilizadas**

Nesta secção descrevem-se as tecnologias e ferramentas levadas a cabo para a elaboração do software Web.

.

# **3.4.1. Ferramentas utilizadas para programação e estilização**

# 3.4.1.1. Html

HTML (HyperText Markup Language - Linguagem de Marcação de Hipertexto) é uma linguagem considerada a base de todas as outras linguagens de desenvolvimento de projetos para WEB. Com ela você pode compartilhar fotos, vídeos, musicas, textos e fazer muitas coisas.

**3.4.1.2. Css**

CSS significa em ingles: Cascading Style Sheet, que traduzindo para o português fica: Folha de Estilo em Cascata. E usado para alterar cores, fontes, tamanho, espaçamentos, etc.

**3.4.1.3. Javascript**

JavaScript é uma linguagem de programação interpretada. JavaScript, podemos injetar logica em nossas paginas HTML.

**3.4.1.4. Php**

PHP (*Hypertext Preprocessor*) é uma linguagem de programação para web de

script *open-source,* trabalha mesclado ao HTML (*Hypertext Markup Language*) e é

executado no lado servidor, o que possibilita que o site seja dinâmico, e garante

mais segurança nas informações, já que o código PHP é todo processado no lado

servidor e o lado cliente fica responsável apenas de apresentar os dados a tela do

usuário.

# **3.4.2. Ferramentas utilizadas para base de dados**

**3.4.2.1. Sql**

A linguagem SQL (Structured Query language) é um padrão para linguagens de declaração de BDR, isto é, para especificar quais tabelas fazem parte da Base de Dado e o esquema de cada uma, bem como se especificar o acesso que se deseja fazer aos dados da Base de Dado.

**3.4.2.2. MySql**

Banco de dados é um local no qual é possível armazenar informações, para

consulta ou utilização, quando necessário. Todos os bancos de dados são constituídos por três elementos básicos: campos, registros e tabelas. Cada banco de dados possui complexas estruturas internas de funcionamento, uma diferente da outra, a fim de facilitar o acesso aos elementos do banco de dados foi criada e distribuída pela IBM em 1981 uma linguagem de consulta, o SQL.

**3.4.3.2. Visual studio code (VSC)**

Visual Studio Code é um editor de código-fonte desenvolvida pela Microsoft para Windows, Linux e macOs. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, completação inteligente de código, snippets e reformação de código.

**3.4.3.3. Xamp**

O Xamp (Extend Apache MariaDB PHP Perl) é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, base de dados MYSQL e Apache com suporte às linguagens PHP e Perl. É possível rodar sistemas como WordPress e Drupal localmente, o que facilita e dá uma eficiência no desenvolvimento. Como o conteúdo armazenado numa rede local, o acesso aos arquivos é realizado instantaneamente. O XAMPP está disponível para quatro sistemas operacionais: Windows, Linux, MAC OS X e Solaris.

**3.6. Requisitos não funcionais**

* Software Web deve ser desenvolvido na linguagem php, JavaScript, html e CSS.
* Software Web de base de dados utilizado é o MYSQL.
* O tempo de resposta do Software Web não deve ultrapassar os 15 segundos.
* O Software Web deve ser utilizado somente pelos utilizadores registados com as respectivas permissões.
* O Software Web deve ser operacional em qualquer plataforma
* O Software deve ser acedido via navegadores web.
* O Software deve apresentar uma interface amigável.
* O Software deve garantir a integridade das informações armazenadas na base de dados.

# **3.7. Análise do Software Web**

A análise do Software foi feita com base no problema apresentado, com o intuito de dar solução ao mesmo e alcançar os objectivo esperados.

### **3.7.1. Casos de uso**

Os casos de uso representam uma técnica baseada em cenário para licitação de requisitos. Ou ainda um caso de uso identifica as interações com o Software. Um caso de uso abrange vários cenários. Existirá um cenário para representar a interação normal e outros para cada possível exceção.

# **3.7.2. Autores do Software Web**

O primeiro elemento a ser feito antes mesmo de se construir o diagrama de caso de uso, é o de identificar os autores do Software. Para todo efeito temos a destacar os seguintes autores: Administrador, Instrutor e usuário.

É definido como actor aqueles que interagem com o Software. Pode ser uma pessoa, organização ou Software Web que tem contacto com o seu site.

**Administrador:** é a pessoa responsável pela plataforma de desenvolvimento de curso online. Tem maiores privilégios na utilização do software em relação a outros utilizadores. Pode então fazer alteração de qualquer conteúdo no Software, visualizar, eliminar ou fazer uma revisão no cótigos.

**Instrutor:** é o responsável pelo atendimento ao usuário numa plataforma de desenvolvimento de curso online que desempenha algumas funções tais como: prestar informações, registrar um pedido, oferecer conhecimento e fornecer feedback construtivo sobre o trabalho dos alunos.

**Usuário:** é a pessoa que interage com o site , onde ele pode se inscrever ou fazer fazer o seu cadastro e alterar o seu perfil, etc.

### **3.7.3. Diagramas**

Um diagrama é uma representação gráfica usada para demonstrar um, esquema simplificado ou um resumo sobre um assunto.

Normalmente é formado por palavras chaves ou conceitos que são ligados por linhas e setas que definem o raciocínio a ser seguido para que seja possível entender o tema de plataforma de desenvolvimento de curso online.

**3.7.3.1 Manter Perfil de Usuário**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **F1** Manter Perfil de Usuário | | | **Oculto ( )** | | |
| **Descrição:** O software deve possibilitar a edição do perfil do usuário. Alterar Nome, Email, Usuário e  Senha. | | | | | |
| **Requisitos Não-Funcionais** | | | | | |
| **Nome Restrição Categoria Desejável Permanente** | | | | | |
| NF1.1  Identificação do  Usuário | O código único referente ao  usuário deve ser gerado  automaticamente pelo software, e  não pode ser editado. | Automação | | ( ) | ( x) |
| NF1.2 Foto do  usuário | O software deve manter  relacionada uma imagem do  usuário. | Interface | | ( X ) | ( ) |
| NF1.3 Projeto  Default | O software deve manter  relacionado um projeto default  para o usuário. | interface | | ( X ) | ( X ) |
| NF1.4 Nível de  Usuário | O usuário Administrador poderá no  software manter relacionado um  nível de usuário que define o papel  entre Administrador, Analista,  Leitor ou Bloqueado | Segurança | | ( ) | ( x) |

**Quadro 1 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

3.7.4.Manter Mensagens Recebidas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F2 Manter Mensagens Recebidas | | | Oculto ( ) | | |
| Descrição: O software deve manter as mensagens recebidas de outros usuários. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF2.1  Apresentação de  mensagens | O software deve apresentar as mensagens recebidas de outros  usuários para o usuário logado. | Interface | | ( ) | (X ) |
| NF2.2 Manter por  tempo  indeterminado | O software não deve permitir a  exclusão das mensagens, apenas identificar se esta pendente de leitura, se já foi lido ou se foi respondido | Histórico | | ( ) | ( X) |
| NF2.3 Responder | O software deve possibilitar a  resposta na mesma tela de leitura. | Interface | | ( ) | ( X) |

**Quadro 2 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F3 Manter Mensagens Enviadas | | | Oculto ( ) | | |
| Descrição: O software deve manter as mensagens enviadas para outros usuários. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF3.1  Apresentação de  mensagens | O software deve apresentar as mensagens enviadas para outros usuários do usuário logado. | Interface | | ( ) | (X ) |
| NF3.2 Manter por  tempo  indeterminado | O software não deve permitir a  exclusão das mensagens, apenas dentificar se esta pendente de leitura, se já foi lido ou se foi respondido. | Histórico | | ( ) | ( X) |
| NF3.3 Responder | O software deve possibilitar o envio de uma nova resposta da própria mensagem. | Interface | | ( ) | ( X) |

**Quadro 3 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F4 Apresentar Itens Reservados | | | Oculto ( ) | | |
|  | | | | | |
| Descrição: O software deve apresentar os itens reservados, a fim de facilitar o acesso ao mesmo. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF4.1  Apresentação dos itens | O software deve apresentar os itens reservados do usuário logado. | Interface | | ( ) | (X ) |
| NF4.2 Acesso ao documento | O software deve possibilitar o  acesso ao documento ao simples clique do item reservado. | Histórico | | ( ) | ( X) |
|  |  |  | |  |  |

**Quadro 4 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F5Manter Cadastro de Usuários | | | Oculto ( ) | | |
| Descrição: O software deve manter o cadastro de usuários. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF5.1 Acesso | O software só deve permitir acesso a manutenção de cadastro de usuários pelo usuário administrador. | Interface | | ( ) | (X ) |
| NF5.2  Apresentação de usuários | O software deve apresentar os  usuários e suas características  particulares. | Histórico | | ( ) | ( X) |
| NF5.3 Ocultação  de senha | O software deve manter oculta a senha dos usuários,  apresentando apenas o caractere \* na quantidade equivalente a quantidade de caracteres da senha descriptografada. | Segurança | | ( ) | ( X) |
| NF5.4 Acesso ao cadastro de  usuário | O sistema deve possibilitar o  acesso ao cadastro do usuário ao simples clique nos usuários  listados. | Usabilidade | | ( ) | ( X) |

**Quadro 5 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F6 Manter Cadastro de Atores | | | Oculto ( ) | | |
| Descrição: O software deve manter o cadastro de atores. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF6.1  Apresentação dos atores | O sistema deve apresentar os atores. | Interface | | ( ) | (X ) |
| NF6.2 Acesso | O software só deve permitir acesso à manutenção de cadastro de atores para usuários Administrador  ou Analista. | Histórico | | ( ) | ( X) |
| NF6.3 Tipo de  Ator | O software deve manter  relacionado um tipo para o ator,entre Pessoa, Hardware e Software. | Interface | | ( ) | ( x) |

**Quadro 6 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F7Manter Cadastro de Projetos | | | Oculto ( ) | | |
| Descrição: O software deve manter o cadastro de projetos. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF7.1  Apresentação dos  projetos | O software deve apresentar os  Projetos. | Interface | | ( ) | (X ) |
| NF7.2 Acesso | O software só deve permitir acesso à manutenção de cadastro de projetos para os usuários Administrador e Analista. | Histórico | | ( ) | ( X) |
|  |  |  | |  |  |

**Quadro 7 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F8 Manter Cadastro de Pastas | | | Oculto ( ) | | |
| Descrição: O software deve manter o cadastro de pastas. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF8.1  Apresentação | O software deve apresentar as pastas de forma hierárquica,  resultando numa arvore de pastas. | Interface | | ( ) | (X ) |
| NF8.2  Manutenção | O software deve permitir a  manutenção do cadastro de  pastas para os usuários  Administrador e Analista. | Histórico | | ( ) | ( X) |
| NF8.3 Listagem  de Documentos | O software deve listar os  documentos relacionados as  pastas quando acessado a  listagem de pastas através do  menu principal. | Interface | | ( ) | ( x) |
| NF8.4 Inativos | O software deve exibir as pastas  com status inativo, e as pastas  posteriores a ela não serão  exibidas. | Interface | | ( ) | ( x) |

**Quadro 1 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F9Manter Cadastro de Documentos | | | Oculto ( ) | | |
| Descrição: O software deve manter o cadastro de Documentos. | | | | | |
| Requisitos Não-Funcionais | | | | | |
| Nome | Restrição | Categoria | | Desejável | Permanente |
| NF9.1  Identificação do  Usuário | O software deve identificar se o usuário logado esta com o documento reservado e disponibiliza interface de edição. | Interface | | ( ) | (X ) |
| NF9.2 Opção de  Reserva | O software ao identificar que o  documento não esta reservado  deve disponibilizar a opção de  reservar documento caso o  usuário seja Administrador ou  Analista. | Histórico | | ( ) | ( X) |
| NF9.3 Leitura de  Documento em  edição | O software deve alertar o usuário  leitor que o documento esta sendo  editado. | Interface | | ( ) | ( x) |
| NF9.4 | O sistema deve identificar a | Interface | | ( ) | ( x) |

**Quadro 8 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificação de  versão que esta  sendo visualizada | versão que o usuário está lendo. | Usabilidade |  | ( |
| NF9.5 Layout  responsivo | O software deve identificar a  largura da tela do dispositivo e  redesenhar a tela. | Usabilidade | ( ) | ( ) |
| NF9.6 Guia | O software deve dispor de um guia  para facilitar o acesso aos tópicos  do documento | Histórico | ( ) | ( ) |
| NF9.7 Histórico | O software deve manter registros  de alterações a fim de diferenciar  versões. | Interface | ( ) | ( ) |
| NF9.8  Identificação de  Alteração | O software deve disponibilizar a  cada tópico uma opção de  gravação quando uma alteração  for feita. | Interface | ( ) | ( ) |
| NF9.9 Validações | O software deve validar campos  importantes, identificando com  visual avermelhado ou  apresentando balões de aviso,  impedindo que a gravação  continue. | Segurança | ( ) | ( ) |
| NF9.10  Publicação de  versão | O software deve no tópico de  Histórico de revisões,  disponibilizar opção para Concluir  alterações e lançar versão do  documento. | Interface | ( ) | ( ) |

**Quadro 9 – Software Web: Manter Perfil de Usuário**

**Fonte: Autoria Própria**

# 3.7.5. Diagrama de classe

Os diagramas de classes são aqueles que permitem especificar a estrutura estática de um Software Web segundo abordagem orientada por objectos. Para o nosso software.

# Figura 3.4: Digrama de classe

Instrutor

instrutor\_id : int(11)

instrutor\_nome : varchar(255)

instrutor\_email : varchar(255)

instrutor\_semha : varchar(255)

Aulas

Id : int(11)

Nome : varchar(255)

Descrição : text

Link \_aula : vachar

Curso\_id : int(11)

Curso

Id : in(11)

Nome: varcha(255)

Descrição : text

Imagem : varchar(255)

Categoria\_id : int(11)

Instrutor\_id : int(11)

Inscrições

Id : int(11)

Usuário\_id : int(11)

Curso\_id : int(11)

Data\_inscricao timestamp

Usuário

Usuário\_id : int(11)

Nome : varchar(255)

Email : varchar(255)

Senha : varchar(255)

Categoria

Id : int(11)

Nome\_vachar(255)

Descrição : text

**3.7.6. Definição das chaves primárias**

Uma chave primária é uma coluna da tabela de base de dados, designada para identificar exclusivamente todos os registros da tabela.

Do ponto de vista de base de dados referem-se aos conjuntos de um ou mais campos, cujos valores, considerando a combinação de valores em caso de mais de uma chave primária, nunca se repetem na mesma tabela e, desta forma, podem ser usadas como um índice de referência para criar relacionamentos com as demais tabelas. Todavia, uma chave primária nunca pode ter valor nulo, nem repetição.

A seguir são apresentadas a lista das tabelas com a definição das suas chaves primárias:

* **Instrutor:** idistrutor, é a chave primária desta classe;
* **Curso**: idcurso, é a chave primária desta classe;
* **Inscrições:** idinscricóes, é a chave primária desta classe;
* **Aula:** aula, é a chave primária desta classe;
* **Categoria:** idcategoria, é a chave primária desta classe;
* **Usuário:** idusuario, é a chave primária desta classe

## **3.8. Segurança**

A segurança da informação é um dos requisitos essenciais a se ter em conta na hora da construção de qualquer sistema (software), pois, permite a protecção de um conjunto de informações contra tentativas de acessos não autorizados feitas por utilizadores ou outras entidades não cadastrados.

Ao nível do software tem-se a destacar as seguintes medidas de segurança:

* Políticas de senhas;
* Validação de dados;
* Protecção a injecção de código;
* Criptografia de dados;

## 

**3.8.2.l. Autenticação**

É um dos pré-requisitos para utilizadores que desejam aceder ao ssoftware. Para tal, é necessário que cada utilizador esteja registado, para ter acesso de acordo as suas permissões. Esse processo ocorre através da inserção do nome do utilizador e da senha.

## **3.8.2. Segurança lógica**

O objectivo da segurança lógica é proteger os dados (informações), programas e softwares contra tentativas de acessos não autorizados, feitas por utilizadores ou outros programas. Para garantir a segurança lógica do software estará salvaguardo os seguintes princípios:

**3.8.2.2. Confidencialidade**

Com isso, garantiu-se que as informações armazenadas no software não devem ser divulgadas a pessoas não autorizadas.

# CAPÍTULO IV. RESULTADO E DISCUSSÃO

Neste capítulo são apresentados os resultados obtidos na fase de construção, através dos protótipos das telas.

# CAPÍTULO V. CONCLUSÕES E RECOMENDAÇÕES

## **5.1. Conclusão**

1. Após a análise dos capítulos precedentes assim como a plataforma de desenvolvimento de curso online, Como diz o ditado, **a primeira impressão é a que fica**. Por isso, o **site**onde você disponibilizará seus cursos online deve ser **atrativo**e passar **credibilidade** para os clientes. A maioria das pessoas julga o livro pela capa!

Com isso em mente, é importante procurar por plataformas para cursos online que possibilitem a **personalização**do seu site de forma simples e intuitiva, sem que você precise contratar programadores ou webdesigners terceirizados. Personalizações de domínio do site, cores de fundo, imagens, favicon e logomarca são requisitos básicos para que você selecione a opção mais completa para o seu negócio.

1. Ter um design responsivo significa funcionar em diferentes tipos dedispositivos. Você já teve dificuldade de navegar em sites no celular que funcionam perfeitamente no seu computador? Certamente não é uma experiência muito agradável. E você não quer que o seu cliente passe por isso quando for acessar seus cursos, não é?

Afinal, hoje os smartphones são a principal forma de acesso à internet, e muito provavelmente é a partir deles que seus alunos irão navegar na plataforma de cursos. Por isso, quando estiver pesquisando opções de plataformas para cursos online, dê prioridade para aquelas que estão preparadas para funcionar nos mais diversos tipos de dispositivos. Assim, a experiência do seu cliente será mais positiva.

1. Dependendo do curso que você quer ofertar, pode ser mais vantajoso comercializá-lo como um pacote único (quando o cliente compra e pode acessar o curso integralmente por tempo ilimitado), cobrar por mês, por semestre, por ano, e assim por diante. Qualquer que seja a modalidade que você deseja oferecer, certifique-se de que a plataforma escolhida tenha essa possibilidade.

É bom lembrar também que algumas pessoas têm uma certa resistência em efetuar compras na internet. Para solucionar esse empecilho, você pode contar com ferramentas conhecidas para [assegurar a segurança do pagamento](https://bit.ly/2DbXalF). Assim, você garante a transparência e a segurança que seu cliente precisa para concluir a negociação pela internet.

1. Ainda falando um pouco sobre segurança na internet, queremos chamar atenção para a segurança dos conteúdos que você utiliza para dar as aulas. Para evitar que o curso vaze e seja alvo de plágio ou pirataria, é importante ter certeza que a plataforma que você utiliza para distribuí-lo tenha várias camadas de**proteção** **e impeça que os usuários repliquem o conteúdo** em outras páginas.

## **5.2. Recomendações**

Estamos abertos para possíveis contribuições. Numa visão futura, espera-se implementar o maior número de funcionalidades possíveis que vão desde canal de comunicação com os clientes, capacidades para rodar em múltiplas plataformas, como sistemas de pagamentos para compras de cursos, sistemas de avaliações, chats entre outras.

# REFERÊNCIAS BIBLIOGRÁFICAS

1. Subcomitê de Software-SC-21:10, CE-21:101.01 – Qualidade de software. URL: http://www.pr.gov.br/celepar/abntsoftware/ce10101 (21/05/2000).

Produto de Software-Características de Qualidade e Diretrizes para o seu Uso.

1996.

3. NBR ISO 9000-3-Diretrizes para aplicação da BNR 19001 ao desenvolvimento, fornecimento e manutenção de software. 1993.

4. NBR ISO/IEC 12119-Pacotes de Software-Teste e Requisitos de Qualidade.

1. <http://sitio.dgidc.minedu.pt/Paginas/default.aspx>
2. <http://eden.dei.uc.pt/~adf/Mitos&des.pdf>.
3. FormatFactory, [on-line] [consulta em 12-12-2008]. Disponível em:

<http://www.formatoz.com/>.

8. MoodlePT, “Página da comunidade *Moodle* Portuguesa”. [online] . Disponível : <http://web.educom.pt/moodlept/>.

1. ARGUELHES, Renato de Brito et al. Startup data quântica para cursos de engenharia de dados. 2021
2. CASTILHOS, Leandro Felipe; SANTOS, Yan Neto dos. Plataforma para organização sistemática de material didático online. 2021.
3. FIGUEIREDO, Paulo N. Programa de pesquisa em aprendizagem tecnológica e inovação na indústria no Brasil. Revista de Administração Pública, v. 34, n. 5, p.
4. GOHN, Daniel Marcondes. A realidade das redes sociais: uma discussão acerca da educação musical nas comunidades virtuais. Revista da Abem, v. 28, 2020.
5. GOMES, Rafael Peduzzi; RIBEIRO, Vinicius Gadis; SILVEIRA, Sidnei Renato. Educação não-formal online em design: uma análise conceitual dos cursos de Design Gráfico da plataforma Udemy. DAPesquisa [recurso eletrônico]. Florianópolis: UDESC, 2021. Vol. 16 (2021), 22 p., 2021.
6. LOVATTO, Paulo Alberto et al. Meta-análise em pesquisas científicas: enfoque em metodologias. Revista Brasileira de Zootecnia, v. 36, p. 285-294, 2007.
7. MARTINEZ, Fernanda. Busca por cursos on-line dispara na pandemia; veja dicas para empreender no universo digital, G1, Matéria jornalística publicada em 03 de agosto de 2021. Disponível em: https://g1.globo.com/economia/pme/noticia/2021/08/03/busca-por-cursos-on-line-dis para-na-pandemia-veja-dicas-para-empreender-no-universo-digital.ghtml.

# APÊNDICE

**Apêndice A- Cronograma de actividades**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Actividades |  |  |  |  |  |  |
| Pesquisa  Bibliografia | x |  |  |  |  |  |
| Revisão de  Literatura |  | x |  |  |  |  |
| Coleta dos  Dados |  |  | x |  |  |  |
| Analise dos  Dados |  |  |  | x |  |  |
| Redação  Preliminar |  |  |  |  | x |  |
| Revisão e  Correção |  |  |  |  |  | x |
| Redação  Final |  |  |  |  |  |  |