



INSTITUTO POLITÉCNICO INDUSTRIAL DE LUANDA (IPIL)
ÁREA DE FORMAÇÃO DE INFORMÁTICA
CURSO TÉCNICO DE GESTÃO DE SISTEMAS INFORMÁTICOS

IMPLEMENTAÇÃO DE UM WEB-SITE DE GESTÃO DE TRABALHOS
ESCOLARES (AISHA SCHOOLWORKS)

Trabalho elaborado para a prova
de aptidão profissional (PAP).

Nome: Rodolf Ommar Quimuanga Gouveia

Nº 09

Turma: IG13A

Luanda

2020/2021



INSTITUTO POLITÉCNICO INDUSTRIAL DE LUANDA (IPIL)
ÁREA DE FORMAÇÃO DE INFORMÁTICA
CURSO TÉCNICO DE GESTÃO DE SISTEMAS INFORMÁTICOS

IMPLEMENTAÇÃO DE UM WEB-SITE DE GESTÃO DE TRABALHOS
ESCOLARES (AISHA SCHOOLWORKS)

Luanda
2020/2021

FOLHA DE APROVAÇÃO

Rodolf Ommar Quimuanga Gouveia

IMPLEMENTAÇÃO DE UM WEB-SITE DE GESTÃO DE TRABALHOS ESCOLARES (AISHA SCHOOLWORKS)

Data de Aprovação ____/____/____

Mesa de Jurados

Júri

1º Vogal

2º Vogal

3º Vogal

Avaliação Final: _____

Orientadores

NELSON TITO MÁRIO

NDINGA MACAIA HERMES

Orientador Regente

Co-Orientador

AGRADECIMENTOS

Agradeço à DEUS pelas conquistas até o momento, aos meus companheiros de jornada que trilharam o mesmo caminho árduo mas nunca se submeteram a desistência, e à todos que directa e indirectamente contribuíram para a realização deste trabalho.

Agradecimento especial ao meu Orientador Professor Eng.º Nelson Tito Mário pelo acompanhamento e orientação, à minha família a quem dedico este trabalho, pelo apoio e força que têm me dado, sem esquecer, ao Professore Eng.º Lucas Pazito Miguel Abel.

RESUMO

Os trabalhos escolares são uma das coisas mais comuns na vida escolar e por vezes a recolha desses mesmos trabalhos apresentam problemas como a perda do mesmo, um só indivíduo entregando mais de um trabalho ou o mesmo fazendo o trabalho de outro indivíduo. E na maioria das vezes a recolha dos trabalhos escolares é feita pelo representante/delegado da classe ou turma e assim entregue ao professor, podendo resultar no atraso ou mistura na dos mesmos perdendo assim a identidade de seu autor quando chegados na posse do Professor/Docente.

Gestão de Trabalhos Escolares é o foco do presente trabalho que se encontra dividido em três partes, respetivamente: Fundamentação Teórica, Fundamentação Prática e Implementação. Na fundamentação teórica falou-se propriamente sobre trabalhos escolares, em seguida foi a fundamentação prática, onde foi abordado sobre as metodologias e tecnologias utilizadas para a execução do projecto e por último a implementação onde foi apresentada uma série de ilustrações relacionado ao funcionamento do website.

Para a criação do website utilizou-se uma plataforma de edição de Código-fonte, sendo que os dados mais importantes do website provêm de um servidor externo.

Palavras-chaves: Trabalhos escolares, professor, aluno, internet, projecto, plataforma, web.

ABSTRACT

Schoolwork is one of the most common things in school life and sometimes the collection of such work presents problems such as the loss of it, a single individual delivering more than one job, or the same doing the work of another individual. And most of the time the collection of schoolwork is done by the representative/delegate of the class or class and thus delivered to the teacher, which may result in delay or mixing in them thus losing the identity of its author when they arrive in the possession of the Teacher.

Schoolwork's Management is the focus of the present work, which is divided into three parts, respectively: Theoretical Foundation, Practical Foundation, and Implementation. On the theoretical basis, we talked about schoolwork, followed by the practical basis, where it was discussed about the methodologies and technologies used for the execution of the project and finally the implementation where a series of illustrations related to the functioning of the website was presented.

To create the website, a source code editing platform was used, with the most important data on the website coming from an external server.

Keywords: Trabalhos escolares, professor, aluno, internet, projecto, plataforma, web.

ÍNDICE DE FIGURAS

Figura 1: Representação Gráfica do Modelo MVC (TREINAMENTOWEB)	27
Figura 2: Imagem do aspecto gráfico do RUP (RODRIGUES DA SILVA e ESCALEIRA VIDEIRA, 2001)	28
Figura 3: Princípio fundamental da DSDM.....	31
Figura 4: Ciclo de Vida de um Projeto em DSDM.....	33
Figura 5: Actores do Sistema	41
Figura 6: Diagrama de Contexto	46
Figura 7: Diagrama de Entidades e Relacionamentos	47
Figura 8: Formulário de Login	59
Figura 9: Formulário de Cadastro.....	59
Figura 10: Página Inicial do Professor	60
Figura 11: Área de Notificações do Professor	60
Figura 12: Perfil do Professor.....	61
Figura 13: Formulário de Criação de Evento para Áreas	61
Figura 14: Formulário para Cadastro de Áreas	62

ÍNDICE DE TABELAS

Tabela 1: Requisitos funcionais do aplicativo	40
Tabela 2: Requisitos não funcionais do aplicativo	40
Tabela 3: Identificação dos casos de uso por actor	43

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

ÍNDICE

AGRADECIMENTOS	I
RESUMO.....	II
ABSTRACT.....	III
ÍNDICE DE FIGURAS	4
ÍNDICE DE TABELAS.....	5
LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS	6
INTRODUÇÃO.....	9
PROBLEMA	10
JUSTIFICATIVA	11
OBJECTIVOS.....	11
OBJECTIVO GERAL	11
OBJECTIVO ESPECÍFICO	11
MÉTODO	12
ESTRUTURA DO TRABALHO	13
CAPÍTULO 1 – FUNDAMENTAÇÃO TEÓRICA	14
1. ESCOLA.....	14
1.1. HISTÓRIA DA ESCOLA	14
1.2. TRABALHO ESCOLAR	16
1.2.1. TIPOS DE TRABALHOS ESCOLARES	19
1.2.2. PROFESSOR.....	21
1.2.3. ALUNO	22
CAPÍTULO 2 – PROGRAMAÇÃO.....	23
1. FRAMEWORKS	23
2. BANCO DE DADOS	24
3. MYSQL	25
4. SERVIDORES WEB.....	25
5. APACHE HTTP SERVER	25
6. PADRÕES DE ARQUITETURAS DE SOFTWARES.....	26
6.1. MVC.....	26
Model ou Modelo	26
View ou Visão.....	27
Controller ou Controlador.....	27
7. METODOLOGIAS DE DESENVOLVIMENTO	27

7.1. RUP	27
7.1.1. CARACTERÍSTICAS PRINCIPAIS	28
7.2. METODOLOGIAS ÁGEIS	29
7.2.1. DSDM	30
CAPÍTULO 3 – IMPLEMENTAÇÃO DO PROJECTO.....	38
1. IMPLEMENTAÇÃO DE UM WEB-SITE DE GESTÃO DE TRABALHOS ESCOLARES.....	38
1.1. DESCRIÇÃO DA PLATAFORMA.....	38
1.2. OBJECTIVO DO PROJECTO.....	38
1.3. RISCOS NO PROCESSO DE DESENVOLVIMENTO DA PLATAFORMA 38	
2. ANÁLISE ESTRUTURADA DO SISTEMA	39
2.1. REQUISITOS	39
2.1.1. REQUISITOS FUNCIONAIS.....	39
2.1.2. REQUISITOS NÃO FUNCIONAIS.....	40
2.1.3. IDENTIFICAÇÃO DE ACTORES	41
2.1.4. CASOS DE USO.....	41
2.1.4.1. IDENTIFICAÇÃO DOS CASOS DE USO POR ACTORES.....	43
3. ANÁLISE DE SISTEMAS.....	44
3.1. DESCRIÇÃO DO OBJECTIVO.....	44
3.2. LISTA DE EVENTOS.....	44
3.3. DIAGRAMA DE CONTEXTO	46
3.4. DIAGRAMA DE ENTIDADES E RELACIONAMENTOS (DER)	47
3.5. DICIONÁRIO DE DADOS	48
4. TECNOLOGIA ENVOLVIDA.....	50
5. RECOMENDAÇÕES	54
6. DIFICULDADES ENCONTRADAS	54
CONCLUSÃO.....	56
REFERÊNCIAS BIBLIOGRÁFICAS.....	57
APÊNDICE.....	59
APÊNDICE A – FORMULÁRIOS E PÁGINAS.....	59

INTRODUÇÃO

Os trabalhos escolares são uma das coisas mais comuns na vida escolar e por vezes a recolha desses mesmo trabalhos apresentam problemas como a perda do mesmo, um só indivíduo entregando mais de um trabalho ou mesmo fazendo o trabalho de outro indivíduo, fazendo que demore a chegar na posse do professor ou até mesmo a chegar adulterado.

Trabalho escolar ou uma pesquisa escolar é uma forma do professor avaliar o aluno. De acordo com o tema dado da disciplina, o professor direciona o aluno a desenvolver de acordo com aquilo que aprendeu ou com o fruto de suas pesquisas um trabalho escolar.

O presente projecto visa implementar um website para a gestão de trabalhos escolares usando uma plataforma digital no intuito de facilitar a recolha dos mesmos. Está hierarquizado em três capítulos:

Falar de Trabalhos Escolares é falar de Escola, no primeiro capítulo será abordado conceitos ligados a escola, como por exemplo, o que é um trabalho escolar, o que o categoriza e quais os seus tipos.

No segundo capítulo será fundamentada as metodologias e tecnologias utilizadas para implementar o projeto. Abordar-se-á sobre a Metodologia RUP, Padrão MVC, Linguagem de Marcação de Hipertexto (HTML), Linguagem Interpretada PHP, Linguagem de Programação Javascript, Linguagem Cascading Style Sheet (CSS), Frameworks, etc.

No terceiro e último capítulo terá como foco principal apresentar uma série de ilustrações relacionadas ao funcionamento do website propriamente dito, como o Modelo Entidade Relacionamento (MER) da estrutura do banco de dados usando MySQL, descrição das tabelas, as páginas de interação do website junto com as explicações de cada.

PROBLEMA

Difícilmente fala-se de escola sem falar de trabalhos escolares. O trabalho escolar é um documento que representa o resultado de um estudo/pesquisa sobre um assunto. Sua produção pode envolver um ou mais alunos e, necessariamente deverá ter a coordenação de um orientador.

E para recolha ou recepção desses trabalhos escolares, os orientadores geralmente instruem o envio dos mesmos em seus correios electrónicos, e assim ele poder acessá-los e depois avaliá-los.

Mas esse método é trabalhoso, porque o orientador tem que acessar os trabalhos, copiar para a sua máquina local, organiza-los por áreas, cursos, turmas ou grupos, quando quiser que um aluno/estudante refaça e reenvie o trabalho, o mesmo terá que contactar o aluno pelo correio electrónico ou por meio de terceiros (colegas ou companheiros do mesmo aluno), por vezes resultando em falha na recepção da mensagem. E depois de avaliá-los, o orientador terá que fazer uma lista com os identificador do aluno (número de processo, nome ou número do aluno) e as avaliações ou/e notas, em seguida imprimi-las para depois anunciá-las aos alunos/estudantes ou enviar na coordenação da instituição para depois fixa-las no painel de informações (vitrine). Por vezes resultando no atraso da entrega e avaliação dos trabalhos e comprometendo a agenda do orientador.

Problemas como esses acima citados têm acontecido também com os variados orientadores e professores em nosso país, onde ainda se nota e desorganizada prática de avaliação de trabalhos escolares (como Monografias, Resenhas ou Relatórios), principalmente em orientadores e professores iniciantes. Mas, e se tivéssemos uma plataforma capaz de resolver esses problemas?

JUSTIFICATIVA

Nos últimos tempos a tecnologia tem sido usada nos mais diferentes sectores em todomundo, tornando ainda maior e cada vez mais comum o seu uso e expansão. Tendo como base os trabalhos escolares ou acto de receber e avaliar os mesmos, este como um dos pilares constituintes na área educacional e empresarial no geral, e, com base nos problemas acima delineados, decidi desenvolver a AISHA Schoolworks, que consiste em uma plataforma web, do tipo de armazenamento e gerenciamento de trabalhos escolares.

A mesma plataforma será usada por professores afim de atualizarem os seus métodos para receber trabalhos escolares de maneira mais eficiente e organizada, e por alunos, estes que irão enviar os seus trabalhos escolares para os seus professores.

OBJECTIVOS

OBJECTIVO GERAL

O presente projecto tem como objectivo geral implementar um website para auxiliarna recepção de trabalhos escolares.

OBJECTIVO ESPECÍFICO

O mesmo tem como objectivos específicos:

- Fazer uma análise rigorosa das referências já existentes sobre de sites eaplicativos de repositórios;
- Levantar requisitos (Funcionais e não funcionais);
- Pesquisar ferramentas necessárias para o desenvolvimento web;
- Prototipar o web-site;
- Desenvolver o web-site;
- Realizar testes de funcionalidade e tratar da validação do web-site.

MÉTODO

Utilizando as funcionalidades do sistema anterior como base (CUSTODIO, 2016), identificaram-se as principais necessidades dos alunos e professores no processo de comunicação e escolha de temas para projetos de final de curso. Assim, perceberam-se maneiras de facilitar cada vez mais as tarefas referentes a esse processo. Dessa maneira, o sistema desenvolvido implementa e aplica todas essas funcionalidades em uma interface amigável e fácil de ser utilizada.

Em um primeiro momento, analisou-se as principais dificuldades encontradas pelos discentes no processo de escolha de algum tema para o desenvolvimento do projecto de final de curso. Algumas situações serviram de estímulo para o presente trabalho, como, por exemplo, o trancamento da matéria de Projecto Tecnológico por vários alunos como resultado da falta de boas ideias e da ausência de professores dispostos a orientá-los por conta dos factores ou consequências causadas pela pandemia do Covid-19.

Posteriormente, foram escolhidas as linguagens a serem utilizadas na implementação do sistema e as plataformas em que esse será suportado. As linguagens utilizadas para o desenvolvimento deste trabalho são HTML, CSS, Javascript, PHP e SQL. Quanto à plataforma, optou-se pelo desenvolvimento de um sistema do tipo WEB. Por fim, todas as funcionalidades foram implementadas e, ao final, avaliadas pelos nossos orientadores do projecto de final de curso.

ESTRUTURA DO TRABALHO

Este trabalho está estruturado com os seguintes capítulos:

Fundamentação Teórica

No capítulo 1, serão apresentadas as tecnologias utilizadas no desenvolvimento do sistema, os requisitos funcionais e não-funcionais, descrevendo detalhadamente cada funcionalidade implementada, os diagramas utilizados no desenvolvimento do sistema, os tipos de utilizadores e as funções a que cada tipo de utilizador tem acesso, bem como a forma de comunicação com o banco de dados.

Avaliação

No capítulo 2, será especificado como foi realizada a avaliação do sistema e são apresentadas as sugestões dos voluntários que participaram desse experimento. Além disso, é realizada uma comparação relativa à interface gráfica e às funcionalidades do sistema anterior com o sistema atual.

Implementação

No capítulo 3, será fundamentado as tecnologias e metodologias usadas para a elaboração deste projecto. Nesta iremos falar das linguagens de programação usadas, máquinas e programas utilizados que contribuíram para a finalização do mesmo.

Conclusão

Nesta, serão expostas as considerações finais sobre o sistema e são propostas algumas mudanças que podem ser realizadas em trabalhos futuros.

CAPÍTULO 1 – FUNDAMENTAÇÃO TEÓRICA

1. ESCOLA

A **Escola** (do grego *scholé*, através do termo latino *schola*) tinha como significado, “discussão ou conferência”, mas também “folga ou ócio”. Este último significado, no caso, seria um tempo ocioso onde era possível ter uma conversa interessante e educativa. Hoje é uma instituição concebida para o ensino de alunos sob a direção de professores. A maioria dos países tem sistemas formais de educação, que geralmente são obrigatórios. Nestes sistemas, os alunos progridem através de uma série de níveis escolares e sucessivos. Os nomes para esses níveis nas escolas variam por país, mas geralmente incluem o ensino fundamental (ensino básico) para crianças e o ensino médio (ensino secundário) para os adolescentes que concluíram o fundamental. Uma instituição onde o ensino superior é ensinado é comumente chamada de faculdade ou universidade.

1.1. HISTÓRIA DA ESCOLA

O conceito de unir alunos em um local separado para a aprendizagem existe desde a Antiguidade Clássica. O ensino fundamental existe provavelmente desde a Grécia antiga, Roma antiga, Índia antiga e China antiga. O Império Bizantino tinha um sistema de ensino criado a partir do nível primário. De acordo com Bentley (2006), a fundação do sistema de educação primária começou em 425 quando “... o pessoal militar geralmente tinha pelo menos o ensino primário ...”. Apesar de Bizâncio ter perdido muito da grandiosidade da cultura romana, o Império enfatizou a eficiência nos seus manuais de guerra. O sistema de ensino bizantino continuou até o colapso do império em 1453.

O Islã foi outra cultura que desenvolveu um sistema escolar, no sentido moderno da palavra. Na Europa, foi durante os séculos XIV e XV que ocorreu a expansão das escolas devido, em grande parte, aos esforços catequistas da Igreja na busca de fiéis. “(...) é preciso estudar a Bíblia para chegar a Deus, e as palavras da liturgia não toleram imprecisão. Cabe à Igreja atrair fiéis, que devem conhecer as preces e os preceitos.”. Assim, os dois últimos séculos da Idade Média presenciam a expansão da escrita, tanto em latim quanto na língua vulgar.

Destinadas as crianças entre sete e quatorze anos, aos poucos a escola traz o livro do domínio eclesiástico e político para o uso quotidiano. Expande-se para os estabelecimentos comerciais ("livros de contas") e chega à zona rural nos contratos de venda ou locação, mesmo para posses pequenas. Também nas profissões, a escola exerce grande influência: frequentar a escola constitui uma prova de honradez, útil para conseguir um bom casamento, tornar-se administrador dos bens da paróquia ou magistrado municipal: *scolae scalae* ("a escola é uma escada").

Estas escolas eram presididas por um eclesiástico, *scholasticus*, subordinado ao bispo, daí o nome de escolástica dado à doutrina e à prática de ensino. Há uma forte demanda por elas, mais para moços, citadinos e mercadores do que para moças, camponeses e pequenos vendedores. Apesar do estímulo à formação de clérigos por Carlos Magno (768-814) e da referência às escolas em 1215 no Concílio de Latrão, não há uma estrutura escolar uniforme, como uma escola por paróquia.

O magistério tem maior concentração nas regiões mais desenvolvidas. Os mosteiros beneditinos recebem rapazes e moças e os jovens pensionistas sempre se tornam monges. Os conventos e confrarias também podem manter escolas, assim como hospitais e orfanatos. Fundar, subvencionar e manter uma escola constitui um ato de misericórdia. A escola pode funcionar ainda, sobretudo na Itália, como empresa privada subvencionada pela comuna. Em uma escala mais reduzida, os mercadores ensinam, a seus aprendizes, as bases da escrita e do cálculo.

A oferta assume várias formas, bem adaptada à demanda dos pais e inserida na continuidade da educação familiar, centrada na aprendizagem dos valores, na socialização e na aquisição de competências precisas. Este tipo de oferta tem seus inconvenientes: a flexibilidade de suas estruturas resulta em um funcionamento aleatório; se o pároco muda ou o mestre decide viajar, a escola para de funcionar.

Na França, as escolas elementares só surgiram na segunda metade do século XIII, e se multiplicaram entre 1350 e 1450. As escolas rurais são relativamente bem conhecidas no norte, na Champanha e na Normandia, ou em toda região rica e urbanizada que tem muitos clérigos a formar e muitos monges para formá-los. No norte, em 1449, das 156 aldeias

de Flandres, 152 possuem uma escola. Na zona rural, a escola raramente ensina a escrever: "saber ler é uma função intelectual valorizada, saber escrever é uma habilidade manual vagamente desprezada". Na cidade, há todos os tipos de escola (cursos em latim ou em língua vulgar), assim como todos os níveis de ensino. No norte, Lille e Saint-Omer possuem trinta escolas, quase uma por paróquia, e Douai possui sete escolas. Em Valenciennes, que conta com vinte escolas em 1337 e 49 em 1388, há, nessa data, 516 crianças (145 meninas) escolarizadas entre sete e dez anos.

1.2. TRABALHO ESCOLAR

Trabalho escolar ou uma **pesquisa escolar** é uma forma do professor avaliar o aluno. De acordo com o tema dado da disciplina, o professor direciona o aluno, a desenvolver de acordo com aquilo que aprendeu um trabalho escolar.

O **Trabalho Escolar** compreende o conteúdo ou desenvolvimento de um determinado tema, e se constitui uma forma de avaliação à qual se confere nota. Pode ser feito individualmente ou em equipe, e de modo geral, trata de assunto que pode ser encontrado em livros didáticos, revistas, enciclopédias ou outras publicações científicas ou culturais.

Ele é às vezes solicitado para resolver problema de falta de nota de alunos que deixaram de fazer alguma prova. Também serve para melhorar a média de alunos tidos como *fracos* em determinada matéria, costume já amplamente disseminado entre os professores, principalmente com aqueles que não querem deixar ninguém para fazer recuperação.

Como substituto de prova os alunos geralmente adoram fazê-lo, pois ao contrário da prova, que é feita na sala de aula e individualmente, o trabalho escolar pode ser feito em casa com toda comodidade, ou então, numa biblioteca, e com prazo de entrega acessível. Infelizmente, porém, há alunos que pagam para alguém fazer isso por eles.

Levado a sério o Trabalho Escolar é da maior importância para ampliação de conhecimentos, desenvolvimento do gosto pela leitura, extensão do raciocínio e organização de método de trabalho. Pena que muitos alunos não entendem isso...

Um trabalho escolar consta, geralmente, do seguinte:

- **Capa**

A **Capa** é a proteção externa do Trabalho Escolar. Há professores que dão nota em função do visual da Capa e, por isso, muitos alunos capricham na sua confecção. Mas é bom não generalizar. O melhor mesmo é considerá-la Capa apenas como um dos componentes do Trabalho Escolar, e não o mais importante.

- **Folha de Rosto ou Contracapa**

É a folha que vem logo depois da Capa. Ela contém os elementos essenciais para identificação do Trabalho, que são praticamente os mesmos escritos na Capa, acrescidos da identificação estudantil do autor ou autores (curso, turma, turno e número) e da *Nota de Apresentação*.

- **Sumário ou Índice**

Sumário é a relação dos itens ou divisões que compõem o Trabalho, com os respectivos números das folhas em que se encontram distribuídos no texto. Basta citar o número da folha a partir da qual o item ou divisão se inicia. Use **Sumário** em vez de **Índice**, pois este é uma lista detalhada dos assuntos em ordem alfabética com indicação de sua localização no texto. O Sumário é o esquema do trabalho. É bom ter cuidado para não errar na numeração. Um professor exigente pode baixar ponto se constatar que você indicou que tal item do Trabalho começa na folha número tal e isto de fato não acontecer. O nome Sumário deve ser escrito com uma margem de 5cm a partir da borda superior da folha e centralizado.

Apesar de os Manuais Técnicos fazerem diferença entre Sumário e Índice, na maioria dos livros o que se constata mesmo é que os termos são usados indistintamente. De nossa parte, recomendamos o uso do Sumário em

Trabalhos Escolares, que é mais adequado para coisa dessa natureza. Mas não podemos ser intransigente neste assunto. Se o aluno optar por Índice e o professor não fizer nenhuma restrição, que se faça assim e pronto, não há o que se discutir. O importante é que os itens ou divisões do trabalho sejam reunidos numa parte ou seção do trabalho com o nome de Sumário (preferencialmente) ou Índice.

- **Introdução**

Na Introdução – que é a primeira parte do desenvolvimento do tema

-, explique sobre a importância e objetivos do Trabalho, forneça detalhes sobre sua elaboração, a maneira como os dados foram obtidos, etc. Nesta parte o assunto ou tema do Trabalho Escolar fica claramente identificado, dando uma panorâmica ou visão geral do que está sendo abordado. Não precisa ser longa e em hipótese alguma deve conter conclusões. O termo Introdução pode ser substituído por **Apresentação**, sem que isto afete o roteiro do trabalho. Deve ser escrito também obedecendo a margem de 5cm e centralizado.

Um detalhe importante: é a partir da Introdução que as folhas são numeradas. O número da folha pode ser escrito na parte superior, à direita, na mesma margem utilizada pelo texto, ou então na parte de baixo, à direita ou no centro

- **Texto ou Desenvolvimento do Conteúdo**

O **Texto** é o Trabalho propriamente dito. É o conteúdo do tema abordado e, portanto, a parte mais substancial do trabalho. Há várias maneiras de se começar a desenvolver o tema. Tudo vai depender do assunto. Há casos em que se pode iniciar com um Histórico, no qual se detalha a evolução do assunto. Em outra situação pode-se começar com o Conceito ou Definição do assunto. Isto, na verdade, funciona como uma espécie de Introdução ao assunto.

Os alunos geralmente se limitam a transcrever o que está nos livros consultados. Mas o melhor mesmo é que você faça algumas modificações, coloque alguma coisa sua, substituindo palavras, invertendo frases, etc

- **Conclusão**

É sem dúvida a parte mais importante do Trabalho Escolar, pois retrata deduções lógicas do autor fundamentadas no conteúdo do texto. Também é a parte que os alunos acham mais difícil. Uma maneira prática é apresentar a conclusão ou conclusões sob a forma de tópicos, empregando-se frases curtas, objetivas.

- **Bibliografia**

Também se diz Fontes de Consulta. Nesta parte devem ser relacionadas todas as publicações (livros, revistas, jornais, etc.), que foram consultadas e das quais se retirou alguma coisa para elaboração do Trabalho Escolar. Existem normas oficiais sobre este assunto. Aqui vamos apresentar um resumo do que consideramos de maior importância para trabalho de nível secundário.

1.2.1. TIPOS DE TRABALHOS ESCOLARES

Os trabalhos escolares podem ser divididos em:

1. Resumo

O resumo nada mais é do que uma síntese das ideias de um determinado material. Ele pode ser dividido em quatro categorias:

- Resumo simples — sintetiza um texto;
- Resumo indicativo — foca nos principais temas da obra;
- Resumo informativo — explica a metodologia do trabalho ou a conclusão obtida;
- Resumo crítico — apresenta a opinião do autor sobre a obra.

2. Relatório

O relatório é um trabalho exclusivamente descritivo. Ele explica o andamento de uma pesquisa ou projeto, sem apresentar a opinião, a análise ou as reflexões do autor.

Essa produção costuma ser realizada após pesquisas científicas e aulas em campo, por exemplo. Quem faz iniciação científica, normalmente, apresenta relatórios como forma de pedir à universidade um financiamento para o projeto.

3. Fichamento

Além de ser cobrado como trabalho acadêmico, o fichamento também é muito útil para a produção dos demais textos. Seu objetivo é fazer com que o aluno separe, de forma dinâmica, os principais pontos de uma obra e escreva-os em tópicos.

Assim como os resumos, ele também pode ser dividido em categorias:

- Fichamento de conteúdo — destaca a ideia do autor e sua visão sobre o tema;
- Fichamento bibliográfico — apresenta uma síntese e análise do texto;
- Fichamento de citações — transcreve citações do original, entre aspas.

• Artigo Científico

O artigo científico apresenta e discute, de forma sintética, ideias, métodos e resultados em diferentes áreas, sobre problemas específicos. Assim, é uma forma rápida de publicar uma pesquisa realizada, com seu referencial, sua metodologia e seus resultados.

Principal forma de publicação em revistas e periódicos da área, o artigo científico só é validado após ser aprovado por uma banca examinadora.

- **Resenha**

Muita gente confunde a resenha com o resumo crítico, uma vez que, nas duas produções, é preciso sintetizar uma obra e opinar sobre ela. Porém, a resenha tem o objetivo de ser publicada e, por isso, deve seguir as normas da ABNT. Assim, é preciso:

4. Identificar os dados da obra;
5. Descrever o texto, com rápida síntese dos capítulos;
6. Fazer uma crítica clara e direta.

6. **Monografia** ou **TCC**

Em alguns cursos, a monografia entregue como Trabalho de Conclusão de Curso é obrigatória para a formação em uma graduação ou pós-graduação. Por isso, há regras de formatação mais rígidas.

É importante que o TCC apresente uma fundamentação teórica, quer seja uma produção de caráter experimental ou uma análise comparativa mais aprofundada. Ele deve demonstrar a consolidação dos conhecimentos adquiridos pelo aluno naquela área.

Por conta de sua importância, os cursos costumam reservar uma matéria para a sua elaboração no último ano da graduação. Após concluída, a monografia deverá ser apresentada a uma banca avaliadora, mesmo se a defesa verbal não for necessária.

1.2.2. **PROFESSOR**

Professor ou **docente** é uma pessoa que ensina ciência, arte, técnica ou outros conhecimentos. Para o exercício dessa profissão, requer-se qualificações acadêmicas e pedagógicas, para que consiga transmitir/ensinar a matéria de estudo da melhor forma possível ao aluno.

1.2.3. ALUNO

Aluno (do latim *alumnus* 'pupilo, aluno'), **aluno** ou **discente** é o indivíduo que recebe formação e instrução de um ou vários professores ou mestres para adquirir ou ampliar seus conhecimentos, geralmente nas áreas intelectuais, levando em conta que existem diferentes aptidões e estilos de aprendizado para cada aluno - principalmente à medida em que avança na vida escolar.

CAPÍTULO 2 – PROGRAMAÇÃO

1. FRAMEWORKS

Atualmente, é esperado que o desenvolvimento de softwares e aplicações seja realizado rapidamente. No entanto, sabe-se que desenvolver algo com qualidade em pouco tempo não é uma tarefa trivial. A utilização correta de frameworks implica na redução do tempo gasto desenvolvendo ao mesmo tempo que melhora a qualidade do produto final. Frameworks são estruturas genéricas reutilizáveis que implementam funcionalidades utilizadas por tipos similares de aplicações. Segundo Schmidt, Gokhale e Natarajan (2004), eles são um conjunto de artefatos integrados (classes, objetos e componentes) que fornecem uma arquitetura reusável à aplicações com domínios semelhantes.

2.2.1 Frameworks e Desenvolvimento Web

A utilização de frameworks no desenvolvimento web é imensa e se faz essencial para o sucesso da aplicação. A maioria dos frameworks para desenvolvimento web se baseiam no padrão de projeto Model-View-Controller (MVC) visto que websites são extremamente dependentes de como é realizada a interação com os dados e o modo que os mesmos são apresentados nas páginas web. “O MVC foi proposto em 1980 como uma abordagem para design de GUI2 , permitindo múltiplas representações de um objeto e diferentes interações com estas representações” (SOMMERVILLE, 2011, p. 432, tradução nossa). Considerando o domínio dos websites, cada componente lógico do MVC pode ser definidos da seguinte maneira:

- **Model:** responsável por gerenciar os dados do sistema.
- **View:** responsável por definir como os dados serão representados e exibidos ao utilizadores. É a partir da view que o utilizadores faz a interação com o sistema.
- **Controller:** responde às interações do utilizador, definindo quais operações serão realizadas, quais dados serão manipulados e quais views serão utilizadas para representá-los.

Sommerville (2011) também afirma que, apesar das diferenças, todos os frameworks para aplicações web geralmente oferecem as seguintes funcionalidades:

- **Segurança:** inclusão de classes para auxiliar o processo de autenticação de utilizadores e gerenciamento de permissões.
- **Páginas web dinâmicas:** capacidade de criação de templates, onde somente determinadas partes dos mesmos são alteradas.
- **Suporte a Banco de Dados:** Permitem a interação da aplicação com um banco de dados externo, facilitando a execução de consultas e CRUD (Create, Read, Update, Delete) de registros.
- **Gerenciamento de sessões:** possuem classes para definir e gerenciar sessões dos utilizadores.
- **Interações de Utilizador:** suporte à AJAX para criação de páginas mais interativas.

2. BANCO DE DADOS

Um banco de dados é uma estrutura, física ou lógica, responsável pelo armazenamento de registros com a finalidade dos mesmos serem consultados quando necessário. Um dos principais recursos utilizados por SIs é o banco de dados digital. Estes possuem inúmeros dados referentes a um tipo de informação tais como clientes, fornecedores, produtos, etc. Cada um destes tipos é comumente chamado de entidade, enquanto que, cada característica que determinada entidade possui é denominada atributo (LAUDON; LAUDON, 2010, p. 145).

Dentre os diversos tipos de banco de dados, os bancos de dados relacionais são os mais utilizados mundialmente. Eles “organizam os dados em tabelas bidimensionais (denominadas relações) com colunas e linhas. Cada tabela contém dados referentes a uma entidade e seus atributos” (LAUDON; LAUDON, 2010, p. 145).

Com o objetivo de gerenciar os bancos de dados computacionais, Sistemas de Gestão de Banco de Dados (SGBD) foram criados. Os SGBDs permitem que os utilizadores executem tarefas tais como armazenamento, organização, adição, acesso e processamento dos dados de maneira mais simples e eficiente. Durante o trabalho, foram utilizados os SGBDs MySQL,

que estava instalado no servidor para testes, e MariaDB, que estava instalado no ambiente de desenvolvimento.

3. MYSQL

O MySQL, segundo Oracle (2017), é desenvolvido, distribuído e mantido pela Oracle Corporation⁵, sendo o SGBD com a versão open-source mais popular do mundo, fazendo uso da Structured Query Language (SQL), que é a linguagem de manipulação de dados utilizada pela maioria dos SGBDs.

4. SERVIDORES WEB

Servidores web permitem que as pessoas acessem os websites na internet. Eles processam as requisições dos utilizadores via Hypertext Transfer Protocol (HTTP), que é o protocolo utilizado para transferência de informações pela internet. A expressão Servidor Web pode referenciar duas tecnologias: computador (hardware) ou programa (software).

O servidor web (computador) possui diversos websites hospedados e permanece conectado a todo instante, permitindo que os utilizadores consigam acessar os websites nele armazenados sempre que necessitarem.

O servidor web (programa) é o software que fica em execução no computador responsável por hospedar os websites. A principal função dele é servir as páginas web, ou seja, ele fica sempre no aguardo de requisições feitas pelos utilizadores (clientes) e responde a essas requisições.

5. APACHE HTTP SERVER

Segundo a Apache (2016), o Apache HTTP Server Project⁷ é um projeto que conta com a colaboração de desenvolvedores de várias partes do mundo, buscando o desenvolvimento e manutenção constante de um servidor HTTP open-source para atender sistemas operacionais modernos. O objetivo do projeto é a disponibilização de um servidor HTTP seguro, eficiente e extensível, sempre estando a par com os padrões atuais HTTP.

O servidor desenvolvido veio a ser chamado Apache HTTP Server (httpd). Ele foi lançado em 1995 e ganhou popularidade em um curto espaço de tempo, se tornando o servidor web mais popular na Internet (APACHE, 2016). Atualmente, o mesmo se encontra na versão 2.4, a mesma utilizada durante o desenvolvimento deste trabalho.

6. PADRÕES DE ARQUITETURAS DE SOFTWARES

6.1. MVC

MVC é o acrônimo de **Model-View-Controller** (em português: **Arquitetura Modelo-Visão-Controle - MVC**) é um padrão de projeto de software, ou padrão de arquitetura de software formulado na década de 1970, focado no reuso de código e a separação de conceitos em três camadas interconectadas, onde a apresentação dos dados e interação dos usuários (front-end) são separados dos métodos que interagem com o banco de dados (back-end).

Normalmente usado para o desenvolvimento de interfaces de usuário que divide uma aplicação em partes (camadas/componentes) interconectadas. Isto é feito para separar representações de informação internas dos modos como a informação é apresentada para e aceita pelo usuário, levando ao desenvolvimento paralelo de maneira eficiente.

O MVC sugere uma maneira para você pensar na divisão de responsabilidades, principalmente dentro de um software web.

O princípio básico do MVC é a divisão da aplicação em três camadas:

- **Model ou Modelo**

A camada de manipulação dos dados. A responsabilidade dos modelos é representar o negócio. Também é responsável pelo acesso e manipulação dos dados na sua aplicação.

- **View ou Visão**

A camada de interação do utilizador. A view é responsável pela interface que será apresentada, mostrando as informações do model para o usuário/utilizador.

- **Controller ou Controlador**

A camada de controle. O Controller é responsável por ligar o model ea view, fazendo com que os models possam ser repassados para as views e vice-versa.

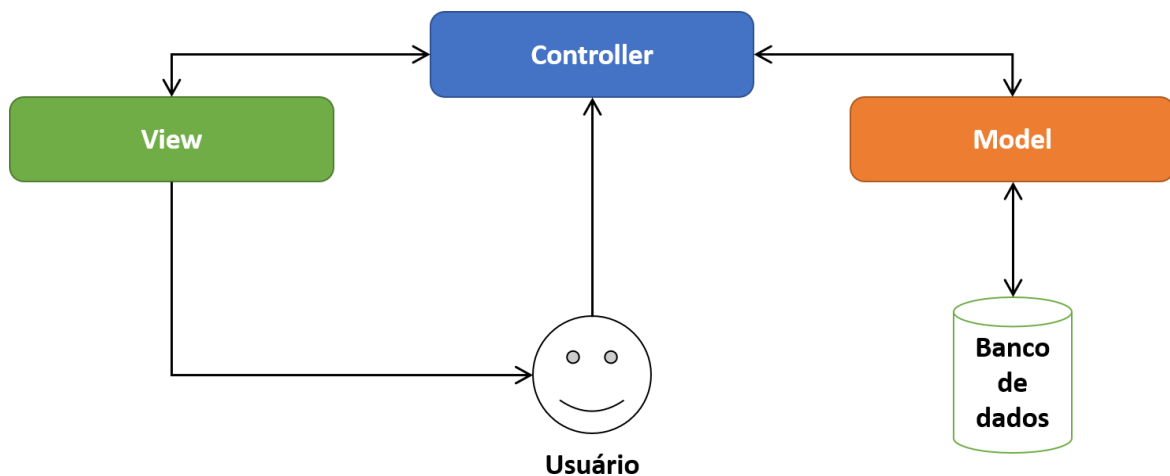


Figura 1: Representação Gráfica do Modelo MVC (TREINAMENTOWEB)

7. METODOLOGIAS DE DESENVOLVIMENTO

7.1. RUP

A RUP (Rational Unified Process), “é uma metodologia de engenharia de software desenvolvida e comercializada pela empresa Rational Software”. Tendo em conta que a construção de software de qualidade de uma forma repetitiva e previsível é difícil, e que as causas dos problemas associados a este tipo de desenvolvimento têm sido uma constante ao longo do tempo, a RUP propõe várias boas práticas para o desenvolvimento de software e aplicadas de forma integrada (RODRIGUES DA SILVA e ESCALEIRA VIDEIRA, 2001).

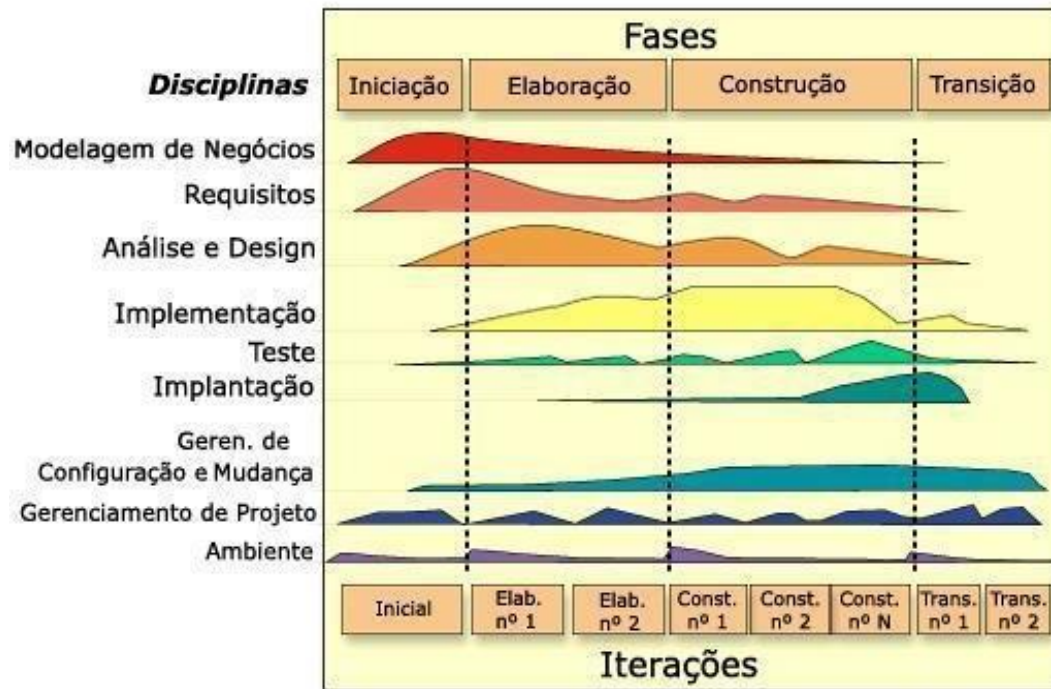


Figura 2: Imagem do aspecto gráfico do RUP (RODRIGUES DA SILVA e ESCALEIRA VIDEIRA, 2001)

7.1.1. CARACTERÍSTICAS PRINCIPAIS

- É uma metodologia de desenvolvimento de *software* iterativa;
- Propõe a gestão integrada de requisitos desde a sua identificação até à implementação;
- Propõe o desenvolvimento de *software* baseado em arquiteturas de software e em componentes;
- Defende a modelação visual;
- O controle de qualidade permanente. Para além destas características, o RUP integra outras ideias fundamentais, nomeadamente o facto de ser orientado

por casos de utilização (RODRIGUES DA SILVA e ESCALEIRA VIDEIRA, 2001).

7.2. METODOLOGIAS ÁGEIS

A maioria do desenvolvimento de software é feita de forma caótica, vulgarmente caracterizada pelo método *programar e corrigir*. O código é feito sem um planeamento aprofundado do problema, sendo muitas vezes agregado a porções de código distinto para a resolução de pequenos problemas que surgem conforme o software vai sendo desenvolvido. (TEIXEIRA, PIRES, *et al.*, 2015).

Este procedimento é eficaz para pequenas aplicações mas, com o aumento do sistema construído, torna-se bastante mais difícil adicionar novas funcionalidades ao programa ou alterar outras já existentes. Além disso, os *bugs* tornam-se bastante mais frequentes e bem mais difíceis de resolver. (TEIXEIRA, PIRES, *et al.*, 2015).

Uma longa fase de testes posterior à conceção do programa é uma característica deste tipo de programação, bem como um sinal de que foi usada. Dado ser muito difícil estabelecer prazos exatos para uma fase de testes e *debugging* tão longa, o tempo previsto para a execução do trabalho é também difícil de determinar. (TEIXEIRA, PIRES, *et al.*, 2015).

Este estilo de desenvolvimento de projetos foi predominante durante muito tempo, tendo como mais forte concorrente o *Desenvolvimento Metódico*. O *Desenvolvimento Metódico* impõe um processo de desenvolvimento disciplinado, com o objetivo de tornar o desenvolvimento de software previsível e eficiente, conseguido através do desenvolvimento de documentação detalhada e dando um grande ênfase ao planeamento inspirado noutras áreas da engenharia de software. (TEIXEIRA, PIRES, *et al.*, 2015).

O desenvolvimento metódico já existe há bastante tempo, mas nunca chegou a ser frequentemente utilizado devido à sua falta de eficácia – é uma metodologia burocrática sendo que, para a seguir, é necessário cumprir diversos itens que tornam o desenvolvimento do software muito mais demorado. (TEIXEIRA, PIRES, *et al.*, 2015).

7.2.1. DSDM

DSDM, “é uma Metodologia Ágil de desenvolvimento de software cujo objetivo é entregar softwares no tempo e com custo estimados através do controle e ajuste de requisitos ao longo do desenvolvimento.” (TEIXEIRA, PIRES, et al., 2015)

Como uma extensão do RAD (Desenvolvimento Rápido de Aplicação), o DSDM é aplicado em projetos de Sistemas caracterizados pelos cronogramas e custos limitados. Aponta *falhas* de informação mais comuns destes projetos, incluindo custos excedentes, perda de prazos, falta de envolvimento de usuários e acompanhamento da alta gerência. Através do uso do RAD, contudo, sem os devidos cuidados com o DSDM pode aumentar ainda mais o risco em outros quesitos. (TEIXEIRA, PIRES, *et al.*, 2015).

7.2.1.1. PRINCÍPIOS DA DSDM

A DSDM segue alguns princípios chave. Estes princípios delimitam as bases do desenvolvimento utilizando DSDM.

- O ponto fundamental desta metodologia prende-se com a entrega de um sistema que se aproxime das atuais necessidades de negócio. Não é uma metodologia tão direta que forneça todas as necessidades de negócio, mas centraliza todo o potencial na concretização final de todos os objetivos do projeto.
- Nenhum sistema é completamente construído na primeira tentativa. Num processo de desenvolvimento de um sistema informático 80% da solução pode ser desenvolvida em 20% do tempo necessário para encontrar a solução perfeita. Para aperfeiçoar a parte final poderá ser necessário que o projeto ultrapasse o seu tempo e orçamento estipulados. Uma vez que a DSDM é caracterizada por realizar exatamente o que a empresa necessita, é muitas vezes desnecessário chegar à solução perfeita.

- A entrega do projeto deve ser feita na data estipulada, dentro do orçamento previsto e com boa qualidade (Fig. 2).
- As exigências para o Sistema de Informação têm que ser flexíveis. Tal como falaremos mais tarde, exigências flexíveis são tópicos importantes da DSDM.
- Esta metodologia apenas requer que cada etapa do desenvolvimento seja completada até que seja possível iniciar o passo seguinte. Isto faz com que cada fase do projeto possa começar sem ter que esperar que as fases que começaram anteriormente sejam totalmente terminadas.
- A comunicação entre todas as partes envolvidas (stakeholders) é também um pré-requisito bastante importante para que o projeto corra com a eficiência desejada.
- O envolvimento dos utilizadores é a chave para esta eficiência.
- As equipas responsáveis têm que ser dotadas de um sentido de decisão, sendo este também um ponto fulcral na progressão do projeto.
- Tal como as equipas de desenvolvimento também as equipas de gestão do projeto estão incorporadas na DSDM.

Após o desenvolvimento do Sistema de Informação, a DSDM pode também ser usado para expandir o Sistema obtido.

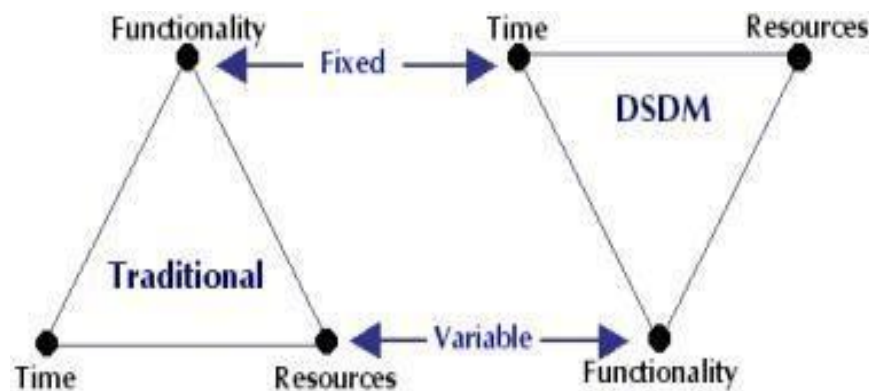


Figura 3: Princípio fundamental da DSDM

7.2.1.2. PRÉ-REQUISITOS PARA A UTILIZAÇÃO DA DSDM

Para que a DSDM seja um sucesso, um número definido de pré-requisitos devem ser implementados.

- Em primeiro lugar, é necessária a existência de uma interatividade entre a equipa de desenvolvimento, os utilizadores finais e os gestores de projeto. A falta de motivação da equipa de gestão ou a falta de envolvimento dos utilizadores finais é uma das mais frequentes causas de falha do desenvolvimento de projetos de Sistemas de Informação.
- Um segundo, e importante, pré-requisito é a facilidade de decomposição em módulos. Esta possibilidade de decompor o projeto em partes mais pequenas, possibilita as abordagens iterativa e incremental da DSDM. Esta funcionalidade possibilita a criação de diversos projetos mais pequenos, também eles a serem desenvolvidos segundo os princípios da DSDM.
- A terceira, e não menos importante, propriedade relaciona-se com a possibilidade de definir claramente os requisitos do SI, introduzindo-lhes uma ordem segundo a sua prioridade de implementação.

7.2.1.3. AS FASES DA DSDM

O *framework* DSDM consiste em três fases sequenciais: **Pré-projecto**, **Projeto** e **Pós-projecto**. A fase de Projeto do DSDM é a mais elaborada das três fases (TEIXEIRA, PIRES, *et al.*, 2015). Ela consiste em 5 níveis formadas por uma abordagem passo-a-passo e iterativa no desenvolvimento de um SI. As três fases e correspondentes níveis são explicadas exhaustivamente nas secções seguintes:

Fase 1: O Pré-projecto

Na fase do pré-projecto, o projeto candidato é identificado, tratando-se depois do seu plano de financiamento e sendo assegurado um compromisso de realização.

Tratar destas questões numa fase inicial evita problemas futuros em fases mais avançadas do desenvolvimento do projeto. (TEIXEIRA, PIRES, *et al.*, 2015).

Fase 2: O Ciclo de Vida do Projeto

A visão geral de um processo DSDM, presente na Fig. 3, representa o **Ciclo de Vida** do Projeto nesta segunda fase da metodologia. Ela mostra os 5 níveis que a equipa de desenvolvimento terá de percorrer para criar um SI. Os dois primeiros níveis, o Estudo de Viabilidade e o Estudo de Negócio, são fases sequenciais que se complementam. (TEIXEIRA, PIRES, *et al.*, 2015).



Figura 4: Ciclo de Vida de um Projeto em DSDM

Depois destas fases estarem concluídas, o sistema é desenvolvido iterativamente e de forma incremental nos níveis de Análise Funcional, Desenho e Implementação.

Nível 1: O Estudo de Viabilidade

Durante este nível do projeto, a viabilidade de utilização da DSDM para este projeto é examinada. Os pré-requisitos para o uso da DSDM são encontrados respondendo a questões como: “Pode este projeto ir de encontro às necessidades de negócio apontadas?”, “É, este projeto, adequado ao uso da DSDM?” e “Quais são os riscos mais importantes envolvidos?”. As técnicas mais importantes utilizadas nesta fase são os *workshops*. (TEIXEIRA, PIRES, *et al.*, 2015).

Para entrega ao cliente, são preparados neste nível o **Relatório** e o **Protótipo de Viabilidade** que dizem respeito à viabilidade do projeto em mãos. A estes, adicionam-se um esboço global do plano para o resto do projeto e um **Registro de Risco** que identifica os riscos mais importantes no projeto. (TEIXEIRA, PIRES, *et al.*, 2015).

Nível 2: O Estudo do Negócio

O Estudo do Negócio incrementa todo o trabalho realizado no Estudo de Viabilidade. Depois do projeto ser declarado fiável para o uso da DSDM, este nível examina o processo de financiamento, os utilizadores envolvidos e as suas necessidades e desejos respetivos. Uma vez mais, os *workshops* são uma das mais valiosas técnicas. *Workshops* nos quais os diferentes *stakeholders* se reúnam e discutam o sistema proposto. (TEIXEIRA, PIRES, *et al.*, 2015).

A informação retirada destas sessões é combinada numa lista de requisitos. Uma importante propriedade desta lista de requisitos é a possibilidade de se definir prioridades. Estas prioridades são definidas utilizando uma perspectiva **MoSCoW1**. Baseado neste escalonamento, um plano de desenvolvimento é construído como uma linha mestra para o resto do projeto. Uma importante técnica utilizada no desenvolvimento do plano é a técnica de **Timeboxing2**. Esta técnica é essencial para serem atingidos os objetivos da DSDM, nomeadamente a imposição de tempo e orçamento fixos, garantindo no entanto a qualidade desejada. Uma arquitetura de sistema é outro meio para guiar o desenvolvimento do Sistema de Informação. (TEIXEIRA, PIRES, *et al.*, 2015).

No final deste nível, deverão estar prontos para entrega ao cliente: uma definição de área de negócio que descreve o contexto do projeto dentro da companhia, a definição da arquitetura do sistema que fornece uma arquitetura global inicial do SI em desenvolvimento juntamente com o plano de desenvolvimento que realça os passos mais importantes no processo de desenvolvimento. Na base destes dois últimos documentos está a lista de prioridades dos requisitos. A lista define todos os requisitos do sistema, organizados de acordo com o princípio do *MoSCoW*. Por fim,

o Registo de Risco é atualizado com os factos que foram identificados durante esta fase da DSDM. (TEIXEIRA, PIRES, *et al.*, 2015).

Nível 3: Análise Funcional

Os requisitos que foram identificados nos níveis anteriores são convertidos para um **Modelo Funcional**. A **Prototipagem** é uma das técnicas chave dentro deste nível, que ajuda no bom envolvimento do utilizador final com o projeto (TEIXEIRA, PIRES, *et al.*, 2015). O protótipo desenvolvido é revisto pelos diferentes grupos de utilizadores. Para assegurar a qualidade do projeto, os testes são implementados em cada iteração da DSDM. Uma importante parte dos testes são realizados na Análise Funcional (TEIXEIRA, PIRES, *et al.*, 2015). O Modelo Funcional pode ser subdividido em quatro subníveis:

- *Identificar Protótipo Funcional*: determinar as funcionalidades a ser implementadas no protótipo que resulta desta iteração.
- *Acordar Calendário de Tarefas*: definir como e quando desenvolver estas funcionalidades.
- *Criar Protótipo Funcional*: desenvolver o protótipo.
- *Rever o Protótipo*: procurar correções possíveis no protótipo desenvolvido. Isto pode ser feito através de testes com utilizadores finais e revendo a documentação.

Neste nível, é necessário entregar ao cliente o Modelo Funcional e o Protótipo Funcional que, juntos, representam as funcionalidades que podem ser realizadas nesta iteração, prontas para serem testadas pelos utilizadores. Além destes dois documentos, a Lista de Requisições é atualizada, sendo apagados os itens que foram implementados e repensando as prioridades dos requisitos restantes. (TEIXEIRA, PIRES, *et al.*, 2015).

Nível 4: Desenho

O ponto central desta iteração da DSDM é a integração das componentes funcionais do nível anterior num sistema que satisfaça as necessidades do utilizador. Mais uma vez, os testes são uma das atividades mais importante (TEIXEIRA, PIRES, *et al.*, 2015) s. O Desenho pode ser dividido em quatro subníveis:

- *Identificar Protótipo de Desenho*: identificar requisições funcionais e não-funcionais que são necessários no sistema testado.
- *Acordar Calendário de Tarefas*: definir como e quando desenvolver estes requisitos.
- *Criar Protótipo de Desenho*: criar um sistema que possa, com segurança, ser fornecido aos utilizadores finais para um uso diário.
- *Rever Protótipo de Desenho*: verificar a exatidão do sistema desenhado. Mais uma vez, os testes e revisões são peças fundamentais.

Ao utilizador, serão entregues o Protótipo de Desenho para que estes efetuem testes ao produto-protótipo.

Nível 5: Implementação

No nível de Implementação, o sistema testado e a sua documentação são entregues aos utilizadores finais que deverão começar a ser treinados para a futura utilização do novo SI. O sistema a ser entregue foi revisto para incluir todos os requisitos que foram definidos nos primeiros níveis do projeto. O nível de implementação pode ser dividido em quatro subníveis:

- *Aprovação do utilizador*: os utilizadores finais aprovam o sistema testado para implementação e as linhas mestras para a implementação e uso do sistema são criadas.
- *Treinar os utilizadores*: treinar os futuros utilizadores no uso do sistema.
- *Implementação*: implementar o sistema testado no local de trabalho dos utilizadores finais.

Rever Negócio: rever o impacto do sistema implementado no negócio, um problema central será tentar compreender se o sistema vai de encontro aos objetivos definidos no início do projeto. Dependendo disto, o projeto passará para a fase seguinte, o Pós-Projeto ou voltará a uma das fases anteriores para desenvolvimento posterior. (TEIXEIRA, PIRES, *et al.*, 2015)

No final deste nível, o sistema deverá ser entregue e instalado, pronto para o uso de todos os utilizadores finais e a **Documentação de Utilização do Sistema** deverá ser detalhada.

Fase 3: Pós-Projeto

A fase de pós-projeto assegura um sistema de atuação eficiente. Isto é implementado através da manutenção e melhoramentos de acordo com os princípios da DSDM. Até mesmo a iniciação de novos projetos, para atualizar o sistema existente ou desenvolver um novo sistema, é possível. (TEIXEIRA, PIRES, *et al.*, 2015).

CAPÍTULO 3 – IMPLEMENTAÇÃO DO PROJECTO

1. IMPLEMENTAÇÃO DE UM WEBSITE DE AUXÍLIO AO PROFESSOR NA RECEPÇÃO DE TRABALHOS ESCOLARES

1.1. DESCRIÇÃO DA PLATAFORMA

A AISHA Schoolworks é uma plataforma web de gestão de trabalhos escolares a nível nacional. Ela começou a ser desenvolvida em Março de 2020 pelo programador Rodolf Gouveia (Andadeuane221) e apresentada a sua primeira parte funcional em Outubro do mesmo ano. Na AISHA Schoolworks os utilizadores poderão assumir o papel de professores e alunos, podendo assim criar eventos, avaliar e estruturar fichas técnicas (no caso de professores) e enviar trabalhos escolares (no caso de alunos).

1.2. OBJECTIVO DO PROJECTO

Com o objectivo de ajudar os professores em nosso país, A AISHA Schoolworks é uma plataforma web de armazenamento e gerenciamento de trabalhos escolares. A plataforma AISHA Schoolworks permitirá aos seus utilizadores processar fichas informativas, criar pastas (eventos) para o agrupamento e avaliação de trabalhos escolares.

1.3. RISCOS NO PROCESSO DE DESENVOLVIMENTO DA PLATAFORMA

Este projecto comporta 3 riscos principais:

1. O primeiro prende-se com a natureza do projecto. Esta plataforma faz parte de um projecto muito grande com muitos módulos e com muitas interacções entre os módulos, sendo estes desenvolvidos por equipas distintas. As interacções têm que ser bem combinadas e estruturadas.
2. O segundo risco relaciona-se com a tecnologia a utilizar. A equipa propõe-se a utilizar a plataforma Php Storm. Esta plataforma de desenvolvimento, assim como a linguagem de programação a utilizar, é muito antiga. As linguagens de programação web, apesar de antigas, não deixam de ser complexas na sua aplicação e implicará uma adaptação.

3. O terceiro e último risco prende-se com o tempo disponível, uma vez que 6 meses para desenvolver um projecto aplicando os conceitos de Engenharia de Software, não é suficiente.

2. ANÁLISE ESTRUTURADA DO SISTEMA

2.1. REQUISITOS

Os requisitos num sistema, “são funcionalidades ou características consideradas relevantes na ótica do utilizador”. Normalmente, representam o comportamento esperado do sistema, que na prática consiste num serviço que deve ser disponibilizado a um utilizador.

2.1.1. REQUISITOS FUNCIONAIS

Requisitos funcionais, “descrevem o que um sistema faz”. São os requisitos que inicialmente serão levantados, abrangendo a descrição de processamento a efetuar pelo sistema, entradas (inputs) e saídas (outputs) de informação em papel ou no ecrã que derivam da interação com pessoas e outros sistemas. A seguir, é apresentada uma tabela com os requisitos funcionais do sistema.

Identificador	Requisitos Funcional
RF01	Cadastro de utilizadores
RF02	Login de utilizadores
RF03	Notificar trabalhos carregados
RF04	Notificar pedidos de cadastro
RF05	Permitir utilizadores consultar e editar seus perfis
RF06	Permitir utilizadores cadastrar, actualizar, consultar e apagar eventos, áreas, cursos, e turmas
RF07	Permitir utilizadores carregar, actualizar, consultar e apagar trabalhos
RF08	Ver histórico de eventos criados
RF09	Ver histórico de trabalhos carregados
RF10	Enviar notificação de trabalhos carregados
RF11	Enviar notificação de eventos cadastrados

RF12	Terminar sessão

Tabela 1: Requisitos funcionais do aplicativo

2.1.2. REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais “estão relacionados com as características qualitativas do sistema, descrevendo a qualidade com que o sistema deverá fornecer os requisitos funcionais”. Abrange medidas de desempenho como, por exemplo, tempos de resposta, volume de dados ou considerações de segurança.

Na tabela a seguir, é apresentado os requisitos não funcionais:

Identificador	Requisito Não Funcional	Categoria
RNF01	A plataforma deverá ter acesso à Internet para obter os dados	Usabilidade
RNF03	Deverá verificar a autenticação do usuário	Segurança
RNF03	Deverá encriptar os dados a circularem bem como os dados de acesso a plataforma	Segurança
RNF04	Deve garantir que toda ação ocorra a devida confirmação do usuário, para que não haja problemas com modificação ou exclusão de dados	Confiabilidade
RNF05	A mesma deve ser acessada a partir de um browser	Portabilidade

Tabela 2: Requisitos não funcionais do aplicativo

2.1.3. IDENTIFICAÇÃO DE ACTORES

A primeira tarefa a desenvolver para contruir um digrama de casos de uso é a identificação dos atores do sistema. Um Actor representa uma entidade externa que interage com o sistema. A figura mostra os atores identificados no desenvolvimento do aplicativo.

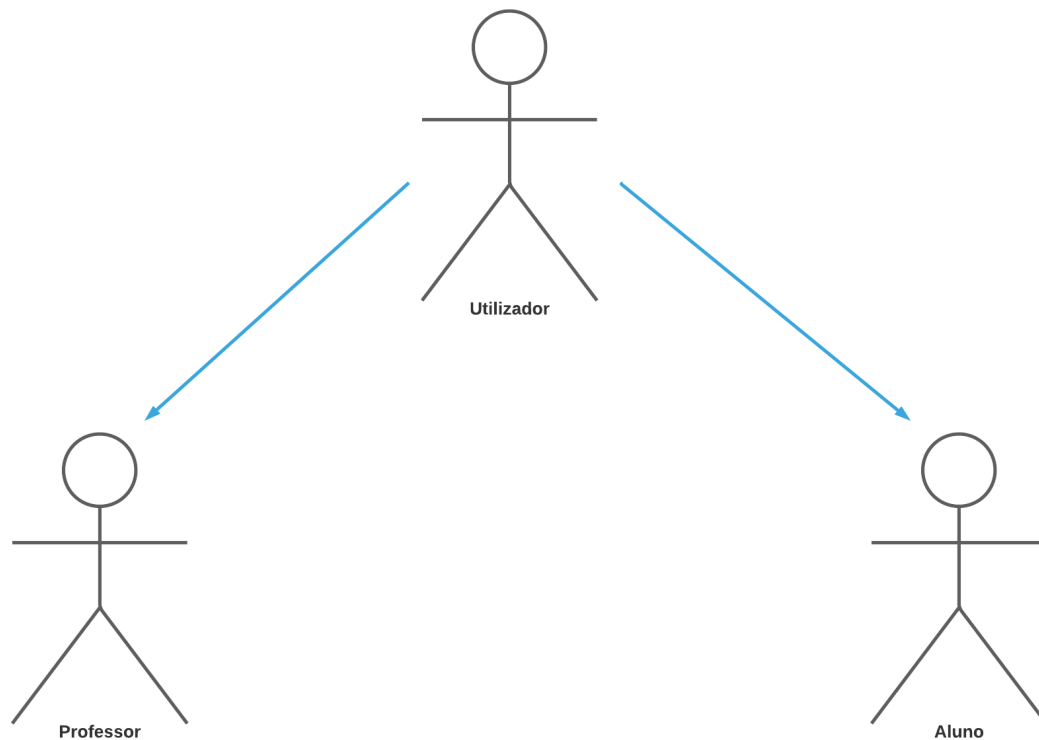


Figura 5: Actores do Sistema

Professor: pessoa que cadastra eventos, turmas, cursos e áreas, avaliaos trabalhos escolares e faz a estruturação da ficha de informações;

Aluno: pessoa que carrega os trabalhos escolares.

2.1.4. CASOS DE USO

Os casos de usos podem ser definidos numa perspetiva de negócio ou desistema. Na primeira perspetiva, procura-se identificar a forma como se responde a

um cliente ou evento em termos de processo de negócio. Na perspectiva do sistema, procura-se caracterizar as funcionalidades que a aplicação a desenvolver deve disponibilizar aos utilizadores.

Um caso de uso corresponde a um conjunto de ações executadas durante a realização de uma funcionalidade do sistema. Casos de uso concentram-se nas relações entre as funções do sistema e os utilizadores que delas participam de alguma forma. Um caso de uso de sistema tem as seguintes características (DE MORAES PEREIRA, 2011):

- Captura as ações para a realização de uma função do sistema, enfocando as interações entre os utilizadores e o sistema;
- É uma unidade coerente de passos, expressa como uma transação entre os atores o sistema, compondo-se tipicamente de várias ações dos atores e respostas do sistema;
- É uma sequência de ações que produzem resultados observáveis de valor para os utilizadores;
- Expressa o que acontece quando um caso de uso é executado, incluindo suas possíveis variações. Não há preocupação em como os participantes executam suas ações, embora a ordem delas seja relevante.

2.1.4.1. IDENTIFICAÇÃO DOS CASOS DE USO POR ACTORES

Actor	Casos de Uso
Professor	Criar eventos na plataforma
	Cadastrar áreas na plataforma
	Cadastrar cursos na plataforma
	Cadastrar turmas na plataforma
	Permitir cadastramento de alunos na plataforma
	Solicitar reenvio de trabalhos escolares na plataforma
	Terminar eventos criados na plataforma
	Solicitar ficha de informação do aluno na plataforma
	Consultar perfil
	Alterar informações do perfil
	Consultar histórico de eventos
	Consultar histórico de trabalhos escolares
Aluno	Carregar trabalhos escolares
	Consultar detalhes do evento
	Consultar Perfil
	Alterar informações do perfil
	Consultar histórico de trabalhos escolares

Tabela 3: Identificação dos casos de uso por actor

3. ANÁLISE DE SISTEMAS

3.1. DESCRIÇÃO DO OBJECTIVO

O objectivo do Web Site de armazenamento e gerenciamento de trabalhos escolares é permitir que cada professor tenha uma conta, permitindo-lhe organizar e fazer a avaliação de trabalhos escolares. O mesmo criará eventos para determinada área, curso ou turma e permitirá que seus alunos carreguem seus trabalhos no evento em questão determinado evento.

3.2. LISTA DE EVENTOS

1. O Sistema permite o cadastro do Professor (I, C)
2. O Sistema notifica o Professor para cadastrar as áreas, cursos, turmas (E, F)
3. O Professor cadastra área no Sistema (E, F)
4. O Professor visualiza área no Sistema (E, C)
5. O Professor actualiza área no Sistema (E, F)
6. O Professor elimina área no Sistema (E, F)
7. O Professor consulta área no Sistema (E, F)
8. O Professor cadastra curso no Sistema (E, F)
9. O Professor visualiza curso no Sistema (E, C)
10. O Professor actualiza curso no Sistema (E, F)
11. O Professor elimina curso no Sistema (E, F)
12. O Professor consulta curso no Sistema (E, F)
13. O Professor cadastra turma no Sistema (E, F)
14. O Professor visualiza turma no Sistema (E, C)
15. O Professor actualiza turma no Sistema (E, F)
16. O Professor elimina turma no Sistema (E, F)
17. O Professor consulta turma no Sistema (E, F)
18. O Professor cadastra evento no Sistema (E, F)
19. O Professor visualiza evento no Sistema (E, C)
20. O Professor actualiza evento no Sistema (E, F)
21. O Professor elimina evento no Sistema (E, F)
22. O Professor consulta evento no Sistema (E, F)
23. O Sistema notifica os eventos cadastrados no Aluno (E, F)
24. O Sistema envia notificação de eventos cadastrados no correio electrónico do Aluno (E, F)
25. O Sistema notifica os eventos actualizados no Aluno (E, F)
26. O Sistema envia notificação de eventos actualizados no correio electrónico do Aluno (E, F)
27. O Professor permite cadastro do Aluno (E, F)
28. O Sistema permite cadastro do Aluno (I, C)
29. O Sistema restringe Aluno a apenas actuar na turma em que pertence (I, C)
30. O Aluno carrega trabalho no Sistema (E, F)
31. O Aluno visualiza trabalho no Sistema (E, F)

32. O Aluno actualiza trabalho no Sistema (E, F)
33. O Aluno elimina trabalho no Sistema (E, F)
34. O Aluno consulta trabalho no Sistema (E, F)
35. O Sistema notifica os trabalhos carregados no Professor (E, F)
36. O Sistema envia notificação dos trabalhos carregados no correio electrónico do Professor (E, F)
37. O Sistema notifica os trabalhos actualizados no Professor (E, F)
38. O Professor demanda recarregamento do trabalho no Sistema (E, F)
39. O Sistema elimina trabalho (I, C)
40. O Sistema notifica demanda de recarregamento do trabalho no Aluno (E, F)
41. O Sistema envia notificação da demanda de recarregamento do trabalho no correio electrónico do Aluno (E, F)
42. O Professor solicita suspensão do Aluno no Sistema (E, F)
43. O Sistema suspende Aluno (E, F)
44. O Sistema notifica Aluno sobre suspensão (E, F)
45. O Sistema envia notificação da suspensão no correio electrónico do Aluno (E, F)
46. O Professor solicita anulação da suspensão do Aluno no Sistema (E, C)
47. O Sistema anula suspensão do Aluno no Sistema (I, C)
48. O Sistema notifica anulação da suspensão no Aluno (E, F)
49. O Sistema envia notificação da anulação da suspensão no correio electrónico do Aluno (E, F)
50. O Professor solicita código de recuperação de conta no Sistema (E, C)
51. O Sistema gera código de recuperação de conta do Professor (I, C)
52. O Sistema envia código de recuperação no correio electrónico do Professor (I, C)
53. O Professor carrega o código de recuperação no Sistema (E, F)
54. O Sistema verifica o código de recuperação (I, C)
55. O Sistema permite redefinição da senha da conta do Professor (I, C)
56. O Professor actualiza senha da conta no Sistema (E, F)
57. O Sistema notifica que a senha da conta foi actualizada no Professor (E, F)
58. O Sistema envia notificação da redefinição da senha da conta no correio electrónico do Professor (E, F)
59. O Aluno solicita código de recuperação de conta no Sistema (E, C)
60. O Sistema gera código de recuperação de conta (I, C)
61. O Sistema envia código de recuperação no correio electrónico do Aluno (I, C)
62. O Aluno carrega o código de recuperação no Sistema (E, F)
63. O Sistema verifica o código de recuperação (I, C)
64. O Sistema permite redefinição da senha da conta do Aluno (I, C)
65. O Aluno actualiza senha da conta no Sistema (E, F)
66. O Sistema notifica que a senha da conta foi actualizada no Aluno (E, F)
67. O Sistema envia notificação da redefinição da senha da conta no correio electrónico do Aluno (E, F)
68. O Aluno solicita remoção da conta no Sistema (E, C)
69. O Sistema remove Aluno (I, C)
70. O Sistema notifica o Professor sobre remoção do Aluno (E, F)
71. O Professor solicita remoção da conta no Sistema (E, F)
72. O Sistema remove Professor (I, C)

3.3. DIAGRAMA DE CONTEXTO

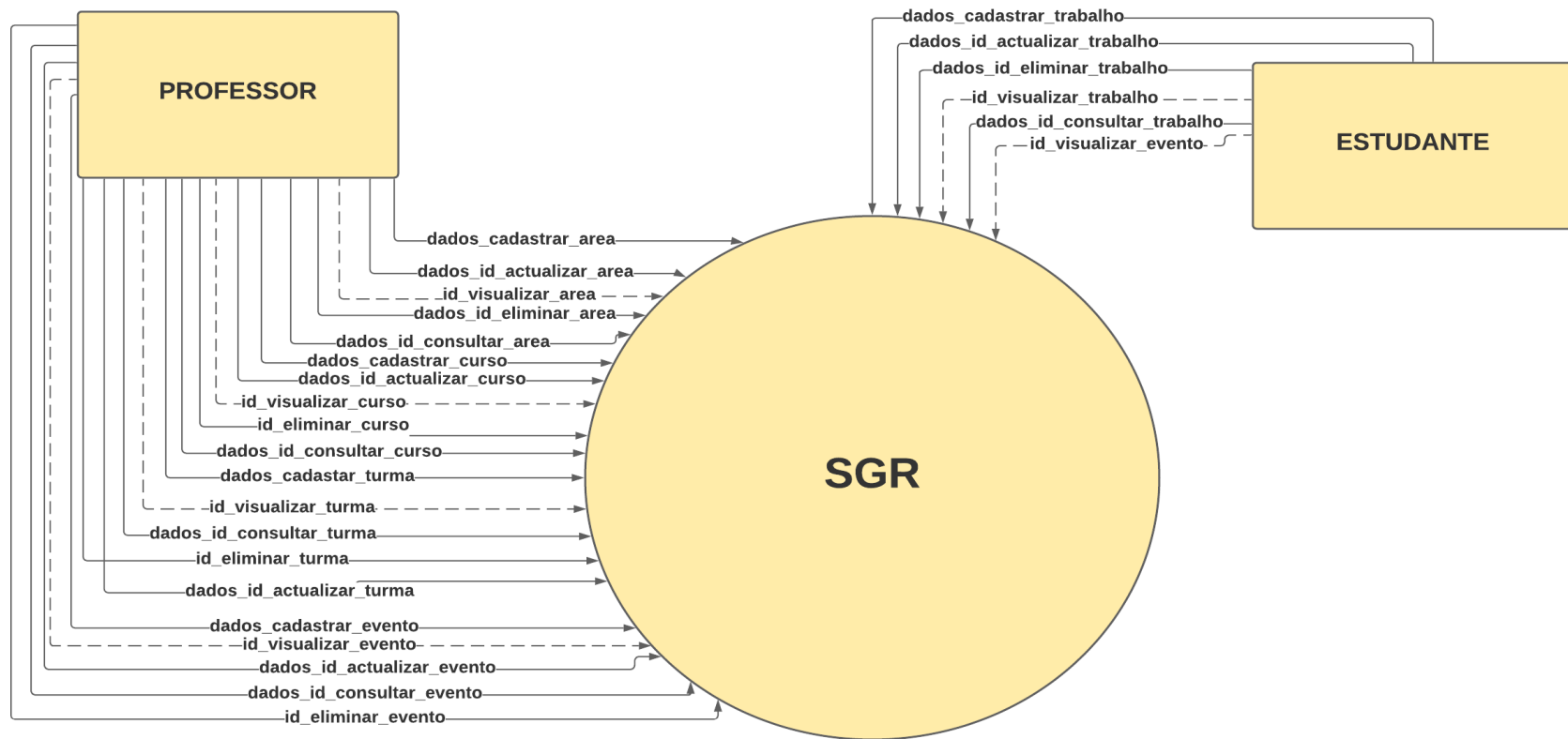


Figura 6: Diagrama de Contexto

3.4. DIAGRAMA DE ENTIDADES E RELACIONAMENTOS (DER)

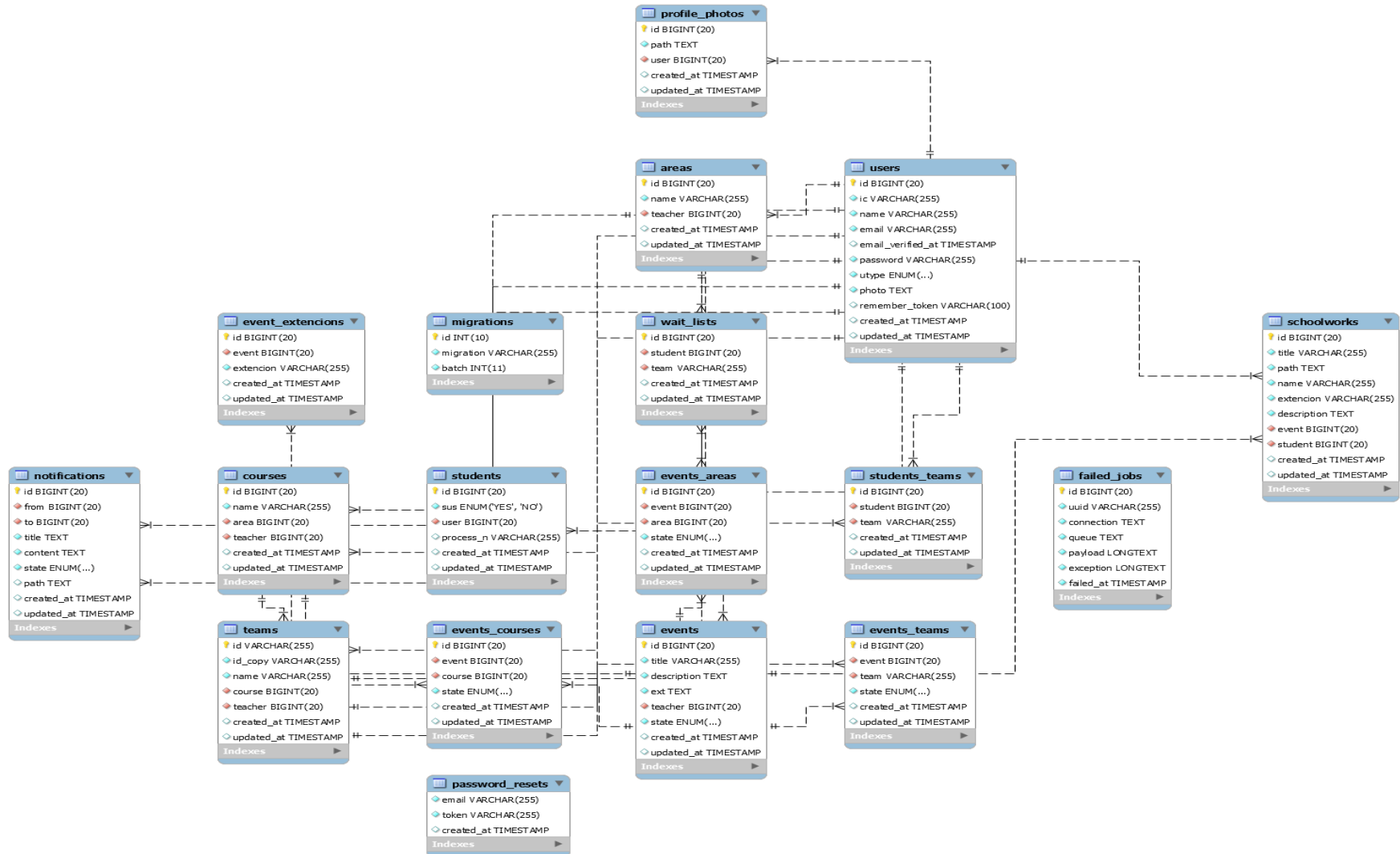


Figura 7: Diagrama de Entidades e Relacionamentos

3.5. DICIONÁRIO DE DADOS

1. User = @id + uuid + name + ic + email + email_verified_at + password + utype + remember_token + created_at + updated_at
2. Areas = @id + name + @teacher + created_at + updated_at
3. Courses = @id + name + @area + @teacher + created_at + updated_at
4. Events = @id + title + @teacher + state ["ATIVO" | "DESATIVO"] + date_end + created_at + updated_at
5. Teams = @id + name + key + @course + @teacher + created_at + updated_at
6. Students = @id + @user + @key + process_n + @photo + created_at + updated_at
7. Events_areas = @id + @event + @area + state ["ATIVO" | "DESATIVO"] + created_at + updated_at
8. Events_courses = @id + @event + @course + state ["ATIVO" | "DESATIVO"] + created_at + updated_at
9. Events_teams = @id + @event + @team + state ["ATIVO" | "DESATIVO"] + created_at + updated_at
10. Schoolworks = @id + title + path + @event + @student + created_at + updated_at
11. Wait_lists = @id + @student + @key + created_at + updated_at
12. Photos = @id + path + @user + created_at + updated_at
13. Migrations = @id + migration + batch

14. Failed_jobs = @id + uuid + connection + queue + payload + exception +failed_at
15. Password_resets = email + token + created_at

4. TECNOLOGIA ENVOLVIDA

Como já havíamos dito antes, para realizarmos este trabalho, usamos os programas ou editores de texto, estes que são:

- Visual Code

Sendo assim, não podemos também deixar de citar os três pilares da programação web que foram bastante essenciais para a elaboração deste projecto, que são:

- HTML
- CSS
- JavaScript
- SQL
- PHP

1. VisualCode

O **VisualCode** também é um IDE ou editor de texto que permite compilar códigos PHP, HTML, CSS e JavaScript, sendo este mais usado na criação de web sites. O VisualCode foi criado pela Microsoft para Windows, Linux e MacOS. Ele contém suporte para debugging, tendo ainda muitas funcionalidades implementadas e melhoradas do que o PhpStorm, sendo este também um dos programas usados no desenvolvimento do presente projecto.

2. HTML

HTML é a sigla para Hypertext Markup Language, que em português quer dizer Linguagem de Marcação de Hipertexto. É uma linguagem de marcação usada para criar Páginas Webs. Quando dizemos Páginas Webs estamos a nos referir á SITES, SISTEMAS WEBS ou até mesmo APLICATIVOS.

3. CSS

CSS é a Sigla para Cascading Style Sheets, que em português significam folhas de estilo em Cascatas. A verdade é que o HTML é uma linguagem que não serve para formatações, ela apenas serve para definir o conteúdo da página e descrever o significado de cada elemento. Não cabe ao HTML fornecer informações sobre apresentação dos elementos, cores, tamanho dos textos e posicionamentos. Isto é trabalho das CSS.

4. JavaScript

JavaScript é uma linguagem de programação. Na verdade a única dentre os 3 pilares. O JavaScript, mais conhecido pela sigla JS é responsável por dar comportamento as páginas web, permitindo não só melhorar elas até um grande certo ponto como também de adicionar as mesmas muitas funcionalidades, estas como cálculos, eventos, animações e outros.

5. XAMPP

XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl.

O nome provem da abreviação de **X** (para qualquer dos diferentes sistemas operativos), **A**pache, **M**ariaDB, **P**HP, **P**erl. É um método que torna extremamente fácil para os desenvolvedores a criar um servidor web local para fins de teste.

Assim como tem o pacote Wamp, que é para Windows, tem o Lamp para Linux e o Mamp para Mac OS X. WampServer é um pacote que instala automaticamente tudo que é necessário para desenvolver aplicações web e é muito intuitivo de usar.

6. Mysql-workbench-community-8.0.13-winx64

MySQL-workbench é um ambiente de desenvolvimento web para Windows que permite criar aplicações com Apache, PHP e Banco de Dados MySQL. Possui o phpMyAdmin que permite gerenciar facilmente os bancos de dados. Nele os dados são agrupados em espécie linha de código e o mesmo ainda possui várias atributos como guardar o código e ainda fazer o MER no mesmo, superando assim em termos de atributos o Wampserver.

7. SQL (Structured Query Language)

A linguagem **SQL** é usada para comunicar com bases de dados. É uma linguagem padrão para base de dados, declarativa, ao contrário das linguagens habituais que são do tipo procedimental. O SQL é uma linguagem de programação que serve criar, actualizar e responder a questões apartir da informação armazenada num Sistema Gerenciador de Base de Dados (SGBD). SQL é a Sigla para Structured Query Language. (Linguagem estruturada para consulta.).

8. PHP

PHP é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e actuaentes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.

9. Bootstrap

Bootstrap é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web usando HTML, CSS, JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do utilizador em um site amigável e responsivo.

10. Laravel

Laravel é um framework PHP livre e open-source criado por Taylor B. Otwell para o desenvolvimento de sistemas web que utilizam o padrão MVC (model, view, controller). Algumas características proeminentes do Laravel são sua sintaxe simples e concisa, um sistema modular com gerenciador de dependências dedicado, várias formas de acesso a banco de dados relacionais e vários utilitários indispensáveis no auxílio ao desenvolvimento e manutenção de sistemas

11. Composer

O Composer é um gerenciador de pacotes no nível do aplicativo para a linguagem de programação PHP que fornece um formato padrão para gerenciar dependências do software PHP e bibliotecas necessárias.

12. Node.js + NPM

Node.js é um software de código aberto, multiplataforma, que executa códigos JavaScript no backend/servidor e frontend/interface (dependendo apenas das bibliotecas e dos frameworks usados), baseado no V8 interpretador de JavaScript em C++ do Google, mantido pela fundação Node.js em parceria com a Linux Foundation.

Mas apenas instalei o Node.js para poder usar um dos seus componentes, o **npm**.

npm é um gerenciador de pacotes para a linguagem de programação JavaScript. npm, Inc. é uma subsidiária do GitHub, que fornece hospedagem para desenvolvimento de software e controle de versão com o uso do Git. npm é o gerenciador de pacotes padrão para o ambiente de tempo de execução JavaScript Node.js.

13. jQuery

jQuery é uma biblioteca de funções JavaScript que interage com o HTML, desenvolvida para simplificar os scripts interpretados no navegador do cliente (client-side).

14. GitHub

GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo. GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos (issues, pull request).

15. GitBash

O **Git Bash** é o aplicativo para ambientes do Microsoft Windows que oferece a camada de emulação para a experiência de linha de comando **Git. Bash** é acrônimo para "Bourne Again Shell". Shells são aplicativos terminais usados como interface em sistemas operacionais por meio de comandos gravados.

16. Gmail

O Gmail é um serviço gratuito de webmail criado pela Google em 2004. Por padrão, o Gmail está programado para usar o HTTPS, um protocolo seguro que fornece comunicações autenticadas e criptografadas. Actualmente novos cadastros estão abertos ao público e podem ser feitos directamente no site.

17. PDF

Portable Document Format (PDF) é utilizado para exibir e/ou compartilhar documentos, independentemente de software, hardware ou sistema operacional.

5. RECOMENDAÇÕES

Recomenda-se os seguintes requisitos mínimos para o bom funcionamento da aplicação:

Cliente

- a) Um computador ou dispositivo móvel (de preferência digital);
- b) Um navegador compatível tal como chrome, Opera Mini, Microsoft Edge ou FireFox;
- c) Conta no Gmail activa;
- d) Leitor de PDF ou browser que possibilita esta ação.

Servidor

- a) Sistema Operacional: Microsoft Windows 10;
- b) Memória: 10GB;
- c) HD: 200Gb;
- d) Processador: 5GHz;
- e) Servidor de Base de dados MySQL;
- f) Antivírus – recomenda-se Kaspersky PURE 2.0 Total Security 2019.

6. DIFICULDADES ENCONTRADAS

- a) Implementar envio de emails;
- b) Implementar recebimento de notificações;
- c) Acesso à internet: sempre que ficava sem internet, não podia testá-la por completo;

- d) Falta de dinheiro de táxi: o capital era todo reencaminhado para o saldo de internet, uma vez que não tinha acesso à mesma e sem ela não haveria grandes avanços;
- e) Definir o layout principal. Foi necessário muito análise, opiniões e críticas das pessoas para construir um layout convincente;
- f) Falta de fontes de informação confiáveis e importantes para o desenvolvimento da plataforma.

CONCLUSÃO

Ao chegar chego a conclusão que este trabalho ajudou-me a desenvolver as minhas habilidades e conhecer novas tecnologias, também mostrando a pressão de como é o mundo profissional.

E se esse projecto fôr levado adiante, a carga de trabalho dos professores a nível nacional diminuirá e este projecto ainda tem a possibilidade de melhorar, citando a adição da versão paga, onde terão mais funcionalidades (como adição da api do whatsapp, que possibilitará o acesso as funcionalidades da plataforma pelo mesmo).

REFERÊNCIAS BIBLIOGRÁFICAS

ALBANI DE CARVALHO, ; FROSSARD TEIXEIRA, G. **Programação Orientada à Objectos**. São Paulo: Clatina - ES, v. Único, 20124

<https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>

<https://www.devmedia.com.br/php-declaracao-e-atribuicao-de-arrays-em-php/38621>

<https://pt.wikipedia.org/wiki/XAMPP>

<https://pt.wikipedia.org/wiki/JQuery>

<https://pt.wikipedia.org/wiki/Node.js>

<https://pt.wikipedia.org/wiki/Composer>

Monteiro, Paulo (2004) - “Proposta Desenvolvimento: PEP - Desenvolvimento da template v1.18”

APACHE. What is the Apache HTTP Server Project? 2016.

Http://httpd.apache.org/ABOUT_APACHE.html.

HOOKE, D. The Seven Principles of Software Development. 1996.

<Http://c2.com/cgi/wiki?SevenPrinciplesOfSoftwareDevelopment>

LAUDON, K.; LAUDON, J. Perspectivas em sistemas de informação e tecnologia da informação. In: Sistemas de Informação Gerenciais. 9. ed. New York: Pearson Prentice Hall, 2010.

OPENLDAP. OpenLDAP Software 2.4 Administrator’s Guide. 2011.

<Http://www.openldap.org/doc/admin24/OpenLDAP-Admin-Guide.pdf>. Acesso em: 6 jul. 2016.

SCHMIDT, D. C.; GOKHALE, A.; NATARAJAN, B. Leveraging application frameworks. Queue, ACM, v. 2, p. 66–75, 2004.

SMITH, A.; JONES, B. On the complexity of computing. In: SMITH-JONES, A. B. (Ed.). Advances in Computer Science. [S.l.]: Publishing Press, 1999. p. 555–566.

SOMMERVILLE, I. Software Engineering. 9. ed. [S.l.]: Pearson, 2011

HTML5SQL. HTML5SQL. 2017. Documentação do HTML5SQL.

SOMMERVILLE, I. Engenharia de Software-9a Edição. [S.l.]: Pearson Education, 2011. W3. HTML5. 2010. Documentação do HTML5.

W3. Javascript. 2017. Documentação do Javascript. W3SCHOOLS. CSS3. 2017. Documentação sobre CSS3

<https://medium.com/@nayeemdev/how-to-create-multi-authentication-with-multiple-table-in-laravel-dbc43e229f9f>

<https://realprogrammer.in/how-to-use-multiple-authentication-guards-in-laravel-8-app/>

<https://laravel.com/docs/8.x/queries>

<https://laravel.com/docs/8.x/filesystem#file-uploads>

<https://coreui.io/laravel/#compare?affChecked=1>

<https://github.com/laravel/framework/issues/18342>

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/datetime-local>

https://www.w3schools.com/tags/att_input_type_datetime-local.asp

<https://stackoverflow.com/questions/17784037/how-to-display-pdf-file-in-html>

<https://stackoverflow.com/questions/44559142/how-to-display-uploaded-image-with-php-laravel>

<https://laravel.com/docs/8.x/filesystem#file-visibility>

<https://www.ssldragon.com/blog/install-an-ssl-certificate-on-xampp/>

<https://www.maujor.com/tutorial/propriedade-css-para-estilizacao-de-background.php>

https://www.php.net/manual/pt_BR/index.php

<https://tecnoblog.net/236041/guia-normas-abnt-trabalho-academico-tcc/>

<https://www.todamateria.com.br/normas-abnt-trabalhos/>

<https://medium.com/@shrijan00003/the-preferred-way-of-using-bootstrap-with-laravel-7-8e51ba845574>

<https://www.portalsaofrancisco.com.br/portugues/trabalho-escolar>

<https://pt.wikipedia.org/wiki/PHP>

https://pt.wikipedia.org/wiki/Linguagem_interpretada

<https://www.coursera.org/learn/ciencia-computacao-python-conceitos>

https://tutorial.djangogirls.org/pt/intro_to_command_line/

APÊNDICE

APÊNDICE A – FORMULÁRIOS E PÁGINAS

Aisha | Schoolworks

Login Cadastro

Login

E-Mail

Senha

Login

[Esqueceu a sua senha?](#)

Figura 8: Formulário de Login

Aisha | Schoolworks

Login Cadastro

Cadastro

Nome:

Nº do BI:

E-Mail:

Palavra-passe:

Confirmar Palavra-passe:

Cadastrar como:

Professor

▼

Cadastrar

Figura 9: Formulário de Cadastro

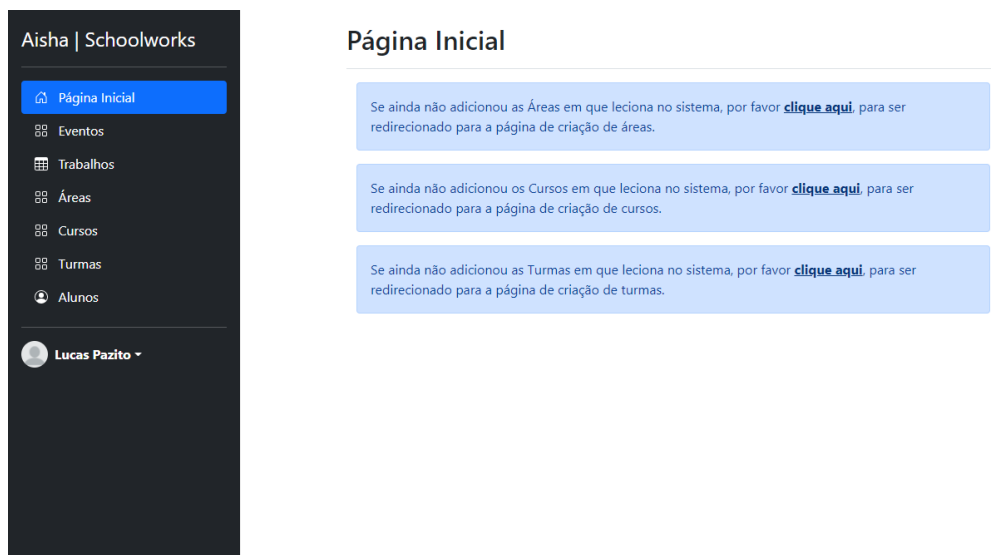


Figura 10: Página Inicial do Professor

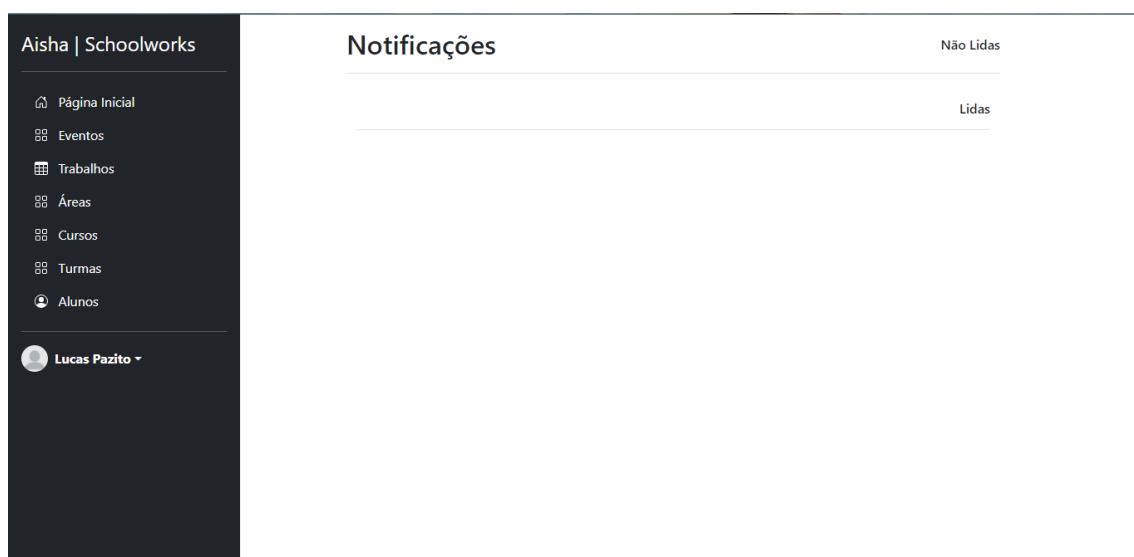


Figura 11: Área de Notificações do Professor

Aisha | Schoolworks

Página Inicial

Eventos

Trabalhos

Áreas

Cursos

Turmas

Alunos

Lucas Pazito

Perfil

Minhas informações

Nome

Lucas Pazito

Email

pazito@gera.com

Nº do Bilhete de Identidade

008876988LA876

Salvar Alterações

Palavra-passe

Palavra-passe Nova

Confirmação da Palavra-passe Nova

Salvar Palavra-passe

Foto de Perfil

Nome: Lucas Pazito

Email: pazito@gera.com

Nº do BI: 008876988LA876

Trocar Foto de Perfil

Adicionais

Eliminar a minha conta

Figura 12: Perfil do Professor

Aisha | Schoolworks

Página Inicial

Eventos

Trabalhos

Áreas

Cursos

Turmas

Alunos

Lucas Pazito

Eventos

Criar um Evento para Área(s)

Lista de Eventos

Adicionar Evento

Nome do Evento

Selecione o estado do Evento.

Activo

Selecione em que formato os trabalhos escolares podem ser enviados neste evento.

Ficheiros PDF (.pdf)

Ficheiros Word (.doc, .docx)

Ficheiros PowerPoint (.ppt)

Ficheiros Excel (.xlsx)

Digite uma descrição para o Evento.

Descrição do Evento

Selecione a Área em que o Evento estará disponibilizado.

Mantenha o dedo pressionado na tecla 'Ctrl' ou 'Command' antes de seleccionar ou desseleccionar.

Adicionar

Figura 13: Formulário de Criação de Evento para Áreas

61

Aisha | Schoolworks

Página Inicial

Eventos

Trabalhos

Áreas

Cursos

Turmas

Alunos

Lucas Pazito

Áreas

Adicionar uma Área

Lista de Áreas

Adicionar uma Área

Adicionar Área

Nome da Área

Adicionar

Figura 14: Formulário para Cadastro de Áreas