

UNIVERSIDADE DO ESTADO DE MINAS GERAIS – UEMG
CAMPUS ITUIUTABA
CURSO DE SISTEMAS DE INFORMAÇÃO

SISTEMA WEB PARA A EMISSÃO DE CERTIFICADOS ACADÊMICOS

ALESSANDRO
FERNANDA

ITUIUTABA - MG
2019

**Alessandro de Medeiros Ferreira
Fernanda Christina Alves Pacheco**

SISTEMA WEB PARA A EMISSÃO DE CERTIFICADOS ACADÊMICOS

Trabalho de Conclusão de Curso de Graduação apresentado ao Curso de Graduação em Sistemas de Informação da Universidade do Estado de Minas Gerais – UEMG – Unidade Ituiutaba, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Prof. Msc. Igor Andrade Moraes

**ITUIUTABA - MG
2019**

RESUMO

Este trabalho de conclusão de curso, apresentado à Universidade do Estado de Minas Gerais – UEMG – unidade Ituiutaba tem como objetivo a criação de um sistema WEB para a automação do processo de emissão de certificados acadêmicos. Ele visa simplificar um processo que atualmente é feito de forma manual em várias instituições de ensino superior e que necessita de um servidor público apenas para a tarefa de confeccionar os certificados. Tendo em vista que as atividades que podem gerar uma certificação que servirá como atestado de comparecimento à determinado evento são muitas e a quantidade de alunos e/ou participantes por curso também é grande, a entrega dos certificados é demorada prejudicando o aluno e/ou participante.

Palavras-chave: Emissão de certificados, Sistema WEB, Aplicação WEB.

ABSTRACT

This course conclusion paper, presented to the Universidade do Estado de Minas Gerais – UEMG – unidade Ituiutaba aims to create a WEB system for the automation of the process of issuing academic certificates. It aims to simplify a process that is currently done manually in various higher education institutions and that requires a public servant just for the task of making the certificates. Given that the activities that can generate a certification that will serve as a certificate of attendance for a particular event are many and the number of students and / or participants per course is also large, the delivery of certificates is slow, harming the student and / or participant. .

Keywords: Certificate issuance, WEB system, WEB application.

Lista de Figuras

Figura 1 - Ciclo de vida de um software.....	15
Figura 2 - Exemplo de diagrama de caso de uso.....	18
Figura 3 - Exemplo de diagrama de sequência.....	18
Figura 4 - Exemplo diagrama de colaboração FONTE: (SAUVÉ, 2003).....	18
Figura 4: Exemplo diagrama de colaboração.....	19
Figura 5 - Esquema de um SGBD.....	20
Figura 6 - Exemplo de solicitação ao servidor web por um usuário web. FONTE: (PALMEIRA, 2012).....	23
Figura 7 - Exemplo de código em PHP.....	25
Figura 8 - Porcentagem de uso de servidores web no mundo.....	26
Figura 9 - Porcentagem de uso de linguagens server-side no mundo.....	26
Figura 10: Camadas de comunicação utilizando o protocolo HTTP.....	29
Figura 11 - Tela inicial do sistema Portal Eventos.....	32
Figura 12 - Tela de cadastro do usuário do sistema Portal Eventos.....	33
Figura 13 - Tela de login do sistema proposto por Librelato e Belusso.....	34
Figura 14 - Tela de cadastro de usuário do sistema proposto por Librelato e Belusso.....	35
Figura 15 - Tela de login do sistema SisAtos.....	37
Figura 16 - Tela de cadastro de eventos sistema SisAtos.....	37
Figura 17: Diagrama de caso de uso do sistema.....	45
Figura 18: Diagrama de caso de uso administrador.....	46
Figura 19: Diagrama de caso de uso professor.....	46
Figura 20: Diagrama de caso de uso usuário.....	47
Figura 21: Tela de login do sistema.....	48
Figura 22: Tela de cadastro de usuários.....	49
Figura 23: Tela de cadastro de eventos do sistema.....	51
Figura 24: Tela de listagem de eventos validados.....	53
Figura 25: Botão para que o usuário possa se cadastrar em um evento.....	55
Figura 26: Botão para que o usuário possa emitir seu certificado FONTE (Própria, 2019).....	56
Figura 27: Diagrama de classes do sistema.....	58
Figura 28: Modelagem de dados: diagrama entidade-relacionamento do sistema.....	59
Figura 29: Modelagem de dados: modelo lógico do sistema.....	60
Figura 30: Modelagem de dados - modelo físico do sistema.....	61
Figura 31: Modelagem de dados - modelo físico do sistema.....	62
Figura 32: Modelagem de dados - modelo físico do sistema.....	63
Figura 33: Modelagem de dados - modelo físico do sistema.....	63
Figura 27: Estrutura de diretórios do sistema a partir do diretório raiz.....	65
Figura 28: Estrutura do banco de dados do sistema.....	66
Figura 30: Tela de cadastro de usuário comum do sistema.....	67
Figura 29: Tela de login do sistema.....	67
Figura 30: Painel de controle do administrador - tela de boas-vindas do sistema.....	67
Figura 32: Painel de controle do administrador - tela de eventos com pendências do sistema.....	68
Figura 33: Painel de controle do administrador - tela de cadastro de evento principal do sistema.....	68
Figura 34: Painel de controle do administrador - tela de cadastro de evento individual do sistema.....	69

Figura 35: Painel de controle do administrador - tela de eventos validados do sistema.....	69
Figura 36: Painel de controle do administrador - tela de eventos que foram criados pelo usuário.....	70
Figura 37: Painel de controle do administrador - tela de eventos em que o usuário está inscrito do sistema.....	70
Figura 38: Painel de controle do administrador - tela de usuários cadastrados do sistema.....	71
Figura 39: Painel de controle do administrador - tela de cadastro de professores do sistema..	71
Figura 40: Painel de controle do administrador - tela de cadastro de usuários do sistema.....	72

Lista de Tabelas

Tabela 1: Comparativo de trabalhos correlatos.....	38
Tabela 2 - Requisito funcional - cadastro de usuário.....	39
Tabela 3 - Requisito funcional - realizar login.....	40
Tabela 4 - Requisito funcional - cadastrar evento.....	40
Tabela 5 - Requisito funcional - visualizar eventos.....	41
Tabela 6 - Requisito funcional - inscrição no evento.....	41
Tabela 7 - Requisito funcional - emitir certificado.....	42
Tabela 8 - Requisito funcional - realizar logoff.....	42
Tabela 9 - Requisito não funcional - menus intuitivos.....	42
Tabela 10 - Requisito não funcional - liberação de certificados.....	43
Tabela 11 - Requisito não funcional - acesso a toda a comunidade.....	43
Tabela 12 - Requisito não funcional - acesso fora do meio acadêmico.....	44
Tabela 13 - Requisito não funcional - autorização de eventos.....	44
Tabela 15 - Caso de uso efetuar login - fluxo ótimo.....	48
Tabela 16 - Caso de uso efetuar login - fluxo alternativo – usuário não possui cadastro.....	48
Tabela 17 - Caso de uso efetuar login - fluxo alternativo – usuário insere senha incorreta.....	49
Tabela 18 - Caso de uso cadastrar usuário - fluxo ótimo.....	50
Tabela 19 - Caso de uso cadastrar usuário - fluxo alternativo – usuário não preenche corretamente os dados solicitados.....	50
Tabela 20 - Caso de uso cadastrar usuário - fluxo alternativo – usuário não preenche completamente o formulário.....	51
Tabela 21 - Caso de uso cadastrar evento - fluxo ótimo.....	52
Tabela 22 - Caso de uso cadastrar evento - fluxo alternativo – usuário não preenche corretamente os dados solicitados no formulário.....	52
Tabela 23 - Caso de uso cadastrar evento - fluxo alternativo – usuário não preenche completamente o formulário.....	53
Tabela 24 - Caso de uso visualizar eventos - fluxo ótimo.....	54
Tabela 25 - Caso de uso visualizar eventos - fluxo alternativo – não existem eventos disponíveis para o usuário.....	54
Tabela 26 - Casdo de uso inscrição em eventos - fluxo ótimo.....	55
Tabela 27 - Caso de uso inscrição em evento - fluxo alternativo – número de vagas insuficiente.....	56
Tabela 28 - Caso de uso emitir certificado - fluxo ótimo.....	57
Tabela 29 - Caso de uso emitir certificado - fluxo alternativo – usuário não realizou cadastro no evento.....	57
Tabela 30 - Caso de uso validar usuário - fluxo ótimo.....	58

Lista de Siglas

CSS	Cascading Style Sheets
DER	Diagrama Entidade Relacionamento
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
MVC	<i>Model-View-Controller</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
UEMG	Universidade do Estado de Minas Gerais
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>

Sumário

CAPÍTULO 1.....	11
INTRODUÇÃO.....	11
1.1 - Introdução.....	11
1.2 - Objetivos.....	13
1.2.1 - Objetivo geral.....	13
1.2.2 - Objetivos específicos.....	13
CAPÍTULO 2.....	14
FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 Engenharia de Software.....	14
2.2 Engenharia de Requisitos.....	16
2.3 UML.....	17
2.4 Banco de Dados.....	19
2.5 Sistema Gerenciador de Banco de Dados.....	19
2.5.1 MySQL.....	20
2.6 Desenvolvimento WEB.....	21
2.6.1 HTML.....	23
2.6.2 CSS.....	23
2.6.3 JavaScript.....	24
2.6.4 JQuery.....	24
2.6.5 PHP.....	25
2.6.6 Composer.....	27
2.6.7 Bootstrap.....	27
2.6.8 IDE.....	27
2.6.9 Visual Studio Code.....	28
2.6.10 Violet UML.....	28
2.6.11 Protocolo HTTP.....	28
2.6.12 XAMPP.....	29
2.7 Conclusão.....	30
CAPÍTULO 3.....	31
O ESTADO DA ARTE.....	31
3.1 “Portal de Eventos” - Um sistema para gerenciamento de eventos acadêmicos e de inscrições.....	31
3.2 Sistema Web para Inscrição em Atividades Acadêmicas.....	33
3.3 Sistema de Atividades e Eventos.....	36
3.4 Quadro de Comparações.....	38
CAPÍTULO 4.....	39
DESENVOLVIMENTO.....	39
4.1 Levantamento dos requisitos.....	39
4.1.1 Requisitos Funcionais (RF).....	39
4.1.2 Requisitos Não Funcionais (RNF).....	42
4.2 Diagrama de caso de uso do sistema.....	44
4.2.1 Especificação dos casos de uso.....	47
4.3 Modelagem de dados.....	58
4.3.1 Diagrama de classes.....	58
4.3.2 Diagrama entidade-relacionamento.....	59

4.3.3 Modelo lógico da modelagem de dados.....	59
4.3.4 Modelo físico.....	61
CAPÍTULO 5.....	64
IMPLEMENTAÇÃO.....	64
CAPÍTULO 6.....	66
RESULTADOS.....	66
CAPÍTULO 7.....	73
CONCLUSÃO.....	73
7.1 Principais contribuições.....	73
7.2 Dificuldades.....	73
7.3 Trabalhos futuros.....	73
BIBLIOGRAFIA.....	74

CAPÍTULO 1

INTRODUÇÃO

1.1 - Introdução

A evolução das tecnologias de informação vem promovendo diversas mudanças na sociedade em geral. Entre elas está a disponibilização de uma quantidade cada vez maior de informações, resultado principalmente do aumento da capacidade de processamento e armazenamento. Santos (2016, p.11).

O surgimento e a popularização da internet foi um dos principais pontos para que este processo pudesse ocorrer e, com o advento da WEB 3.0, a forma como vários sistemas e dispositivos se comunicam foi simplificada. Segundo LEE (2007, *apud* LOTH *et al.* 2019, p. 41):

A Web Semântica é sobre a colocação de arquivos de dados na web. Não é apenas uma web de documentos, mas também de dados. A tecnologia de dados da Web Semântica terá muitas aplicações, todas interconectadas. Pela primeira vez haverá um formato comum de dados para todos os aplicativos, permitindo que os bancos de dados e as páginas da web troquem arquivos.

Toda essa tecnologia também se aplica às instituições de âmbito público, segundo Pereira *et al.* (2016, p.1)

“O grande volume de informações geradas, processadas e disseminadas pela evolução tecnológica resulta na necessidade de informatização de processos. Nas instituições públicas de ensino superior, barreiras burocráticas estão sendo rompidas e Sistemas de Informação (SI) têm sido implantados para facilitar e agilizar processos internos.”

Estes processos muitas vezes são onerosos e dependem que servidores públicos despendam horas de seu dia-a-dia para a execução das tarefas em um era em que, segundo Campelo e Pinto (2010, p.4 *apud* Pereira *et al.* 2016, p.19) “nunca se falou tanto em

automatizar e gerenciar os processos organizacionais como nos últimos anos. O desafio tem sido implantar sistemas que possam processar os dados e convertê-los em informação útil”.

Um processo interno muito comum em universidades é o processo de confecção de certificados de palestras, eventos, minicursos, etc. Estes certificados servem para validar a participação dos candidatos nestes tipos de evento. Porém um grande agravante neste processo é a quantidade de eventos realizados por ano somado à grande quantidade de participantes o que torna o processo de confecção dos certificados bastante penosa já que na maioria das vezes é um processo manual.

Essa problemática foi a causa da elaboração do presente trabalho que visa a criação de um sistema WEB para a informatização do processo de solicitação de certificados assim como a emissão automática dos mesmos através de uma plataforma web utilizando tecnologias já consolidadas no mercado quando o assunto é programação web .

1.2 - Objetivos

1.2.1 - Objetivo geral

Desenvolvimento de um sistema WEB para a emissão de certificados de eventos acadêmicos.

1.2.2 - Objetivos específicos

1. Automatizar o processo de emissão de certificados;
2. Auxiliar o usuário a visualizar todos os eventos oferecidos pela instituição;
3. Auxiliar a instituição na divulgação dos eventos;
4. Ajudar o professor a gerenciar os eventos;

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

No capítulo que segue serão apresentados as tecnologias e conceitos que serviram de embasamento teórico para a realização do presente trabalho de conclusão de curso, assim como suas características. Ao final do capítulo será apresentado a análise conclusiva do capítulo feita pelo autor do trabalho explicando a escolha de cada uma das ferramentas.

2.1 Engenharia de Software

A engenharia de software trata o desenvolvimento de software como um todo, desde, sua concepção, isto é, qual tipo de software será produzido, suas especificações, sua documentação, possibilidades de mudanças e manutenção ao longo de sua existência. Segundo Sommerville (2011) “A engenharia de software tem por objetivo apoiar o desenvolvimento profissional de software [...] Ela inclui técnicas que apoiam especificação, projeto e evolução de programas”.

Portanto, como já foi mencionado, a engenharia de software engloba os softwares de profissionais, ou seja, aqueles que podem ser vendidos a clientes. Sommerville (2011) diz que existem dois tipos de software:

- 1 - Produtos genéricos: Existem sistemas stand-alone, produzidos por uma organização de desenvolvimento e vendidos no mercado para qualquer cliente que esteja interessado em comprá-los. Exemplos desse tipo de produto incluem software para PCs, como ferramentas de banco de dados, processadores de texto, pacotes gráficos e gerenciamento de projetos. Também incluem as chamadas aplicações verticais projetadas para um propósito específico, como sistemas de informação de bibliotecas, sistemas de contabilidade ou sistemas de manutenção de registros odontológicos.
- 2 - Produtos sob encomenda: Estes são os sistemas encomendados por um cliente em particular. Uma empresa de software desenvolve o software especialmente para esse cliente. Exemplos desse tipo de software são sistemas de controle de dispositivos eletrônicos, sistemas escritos para apoiar um processo de negócio específico e sistemas de controle de tráfego aéreo.

Na engenharia de software, as aplicações são tratadas como produtos e como produto os softwares também possuem um ciclo de vida. Abaixo temos o ciclo de vida de um software:

Figura 1 - Ciclo de vida de um software

Ciclo de vida	Percepção da necessidade			
	Desenvolvimento	Concepção		
		Elaboração		
		Construção	Desenho inicial	
			Liberação	Desenho detalhado
				Codificação
				Testes de unidade
			Testes alfa	
		Transição		
	Operação			
Retirada				

FONTE: (FILHO, 2003)

É interessante notar que, para muitos, o papel da engenharia de software está ligado diretamente a codificação, Filho (2003) diz que “Para a maioria das pessoas, inclusive muitos profissionais da informática, esta parece ser a única tarefa de um programador, ou seja, um produtor de software”, o que já sabemos não ser a realidade.

Em toda sua plenitude, a engenharia de software deve atender a todo tipo de sistema, e cada tipo deles possuem suas especificidades. Como já foi dito, a engenharia de software utiliza-se de técnicas para apoiar o desenvolvimento destas aplicações, porém não existe uma fórmula mágica universal para amparar a todas elas, segundo Sommerville (2011) “Existem vários tipos de sistemas de software, desde os simples sistemas embutidos até os sistemas de informações complexos, de alcance mundial. Não faz sentido procurar notações, métodos ou técnicas universais para a engenharia de software, porque diferentes tipos de software exigem abordagens diferentes”. Isto é muito importante, pois sistemas desktop são diferente de sistemas WEB. O produto do presente trabalho será um sistema web, mas a diante será explicado como a engenharia de software contempla este tipo de sistema.

Aspectos financeiros e sociais também são levados em conta pela engenharia de software no momento da concepção dos sistemas. Ainda segundo Sommerville (2011):

Engenharia de software é uma abordagem sistemática para a produção de software; ela analisa questões práticas de custo, prazo e confiança, assim como as necessidades dos clientes e produtores do software. A forma como essa abordagem sistemática é realmente implementada varia dramaticamente de acordo com a organização que esteja desenvolvendo o software, o tipo de software e as pessoas envolvidas no processo de desenvolvimento. Não existem técnicas e métodos universais na engenharia de software adequados a todos os sistemas e todas as empresas. Em vez disso, um conjunto diverso de métodos e ferramentas de engenharia de software tem evoluído nos últimos 50 anos.

2.2 Engenharia de Requisitos

No processo de concepção de um sistema, seja ele de qual tipo for, é necessário entender as especificações do sistema, isto é, entender o que o sistema deve fazer, o que ele não pode fazer e todas as funcionalidades que ele deve oferecer. Segundo Sommerville (2011) “Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento”.

O documento de requisitos do sistema além de orientar a forma de como o sistema deverá ser construído, faz parte da documentação do sistema, segundo NARDI (2006) “Os requisitos elicitados e analisados devem ser documentados. Na atividade de documentação, um Documento de Especificação de Requisitos de Software (DERS) é normalmente gerado”.

Ainda segundo NARDI (2006)

“Qualquer que seja seu formato, o DERS é um dos artefatos mais importantes do processo de software, uma vez que é a base para praticamente todas as atividades de construção subsequentes. Assim, é imprescindível avaliar a qualidade de um DERS. Para tal, características de qualidade devem ser apontadas e métricas devem ser estabelecidas para avaliar a qualidade do DERS. “

Os requisitos são divididos em dois tipos: os requisitos funcionais e os requisitos não funcionais. Sommerville (2011) explica cada um deles da seguinte forma:

1 – Requisitos funcionais. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se

comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

2 – Requisitos não funcionais. São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

Deve-se estar atento também às mudanças nos requisitos, que são inevitáveis. É considerada boa prática de gerência de requisitos tentar antecipar as mudanças dos requisitos. Isto normalmente envolve classificar os requisitos para identificar os mais voláteis e analisar as possíveis mudanças que podem surgir (KOTONYA, G.; SOMMERVILLE, I., 1998 apud ESPINDOLA, R. S.; MAJDENBAUM, A.; AUDY, J L. N. 2004).

2.3 UML

Para o processo de produção de softwares também se faz necessário demonstrar como o sistema e suas funcionalidades e o meio externo a ele interagem entre si. A forma mais comum de se demonstrar essas interações é por meio dos diagramas da Linguagem de Modelagem Unificada (ou do inglês, *Unified Modeling Language* – UML). Macoratti (2019) define UML como “uma linguagem de modelagem designada para especificar, visualizar, construir e documentar um sistema”. Portanto os diagramas provenientes da UML também compõe a documentação do sistema.

Alguns diagramas da UML também fazem parte da especificação dos requisitos. Segundo Macoratti (2019), A análise de requisitos compõe-se dos seguintes diagramas: 1 – Diagrama de caso de uso; 2 – Diagrama de sequência e diagrama de colaboração. Ele explica cada um dos diagramas da seguinte forma:

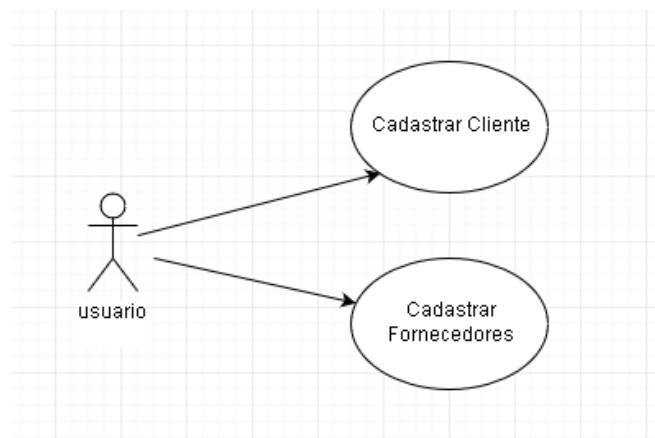
Diagrama de caso de uso: É um diagrama usado para se identificar como o sistema se comporta em várias situações que podem ocorrer durante sua operação. Descrevem o sistema, seu ambiente e a relação entre os dois

Diagrama de sequência: Um diagrama de sequência é um diagrama de objetos, ou seja, ele contém como primitiva principal um conjunto de objetos de diferentes classes. O objetivo dos diagramas de sequência é descrever as comunicações

necessárias entre objetos para a realização dos processos em um sistema computacional. Os diagramas de sequência têm este nome porque descrevem ao longo de uma linha de tempo a sequência de comunicações entre objetos.

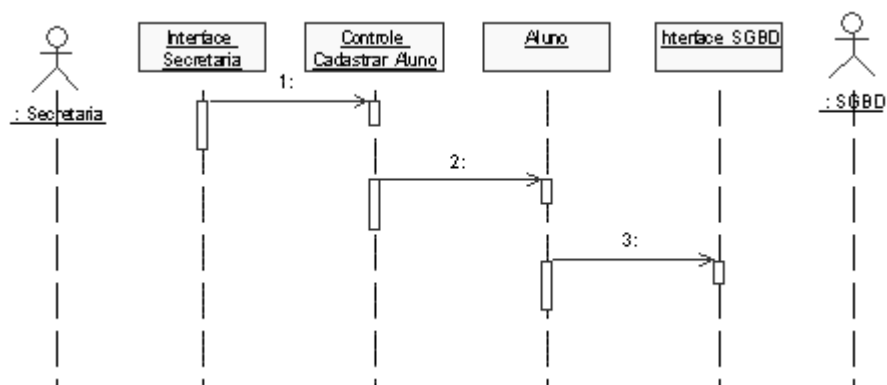
Diagrama de colaboração: Mostra a interação organizada ao lado de cada classe de objetos e a ligação entre elas. Como o diagrama de sequência, o diagrama de colaboração mostra as mensagens trocadas por um conjunto de objetos durante um cenário. É uma representação gráfica alternativa para mostrar um cenário.

Figura 2 - Exemplo de diagrama de caso de uso



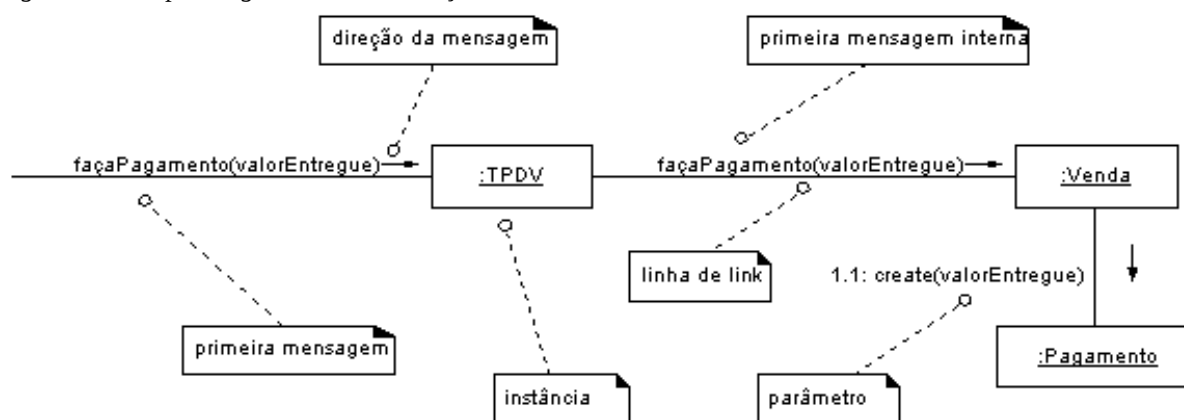
FONTE: (Própria, 2019)

Figura 3 - Exemplo de diagrama de sequência



FONTE: (MACORATTI, 2019)

Figura 4: Exemplo diagrama de colaboração



FONTE: (SAUVÉ, 2003)

2.4 Banco de Dados

“Os bancos de dados e os sistemas de bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia, a maioria de nós se depara com atividades que envolvem alguma interação com os bancos de dados.” (ELMASRI, 2005, p.2).

Segundo Korth (1994, apud REZENDE, 2006), um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”. Disso podemos concluir que sempre que tivermos informações que abrangem um mesmo escopo e que estiverem reunidas podemos dizer que temos um banco de dados.

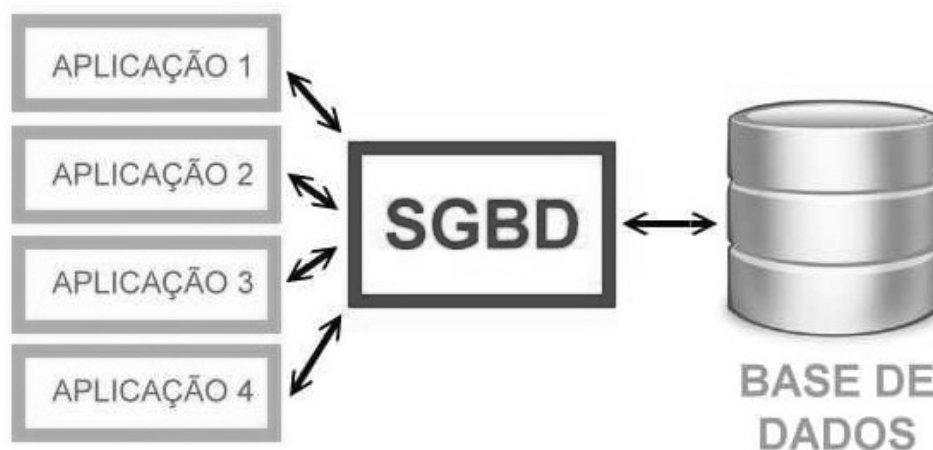
2.5 Sistema Gerenciador de Banco de Dados

Como já foi dito, banco de dados é tudo aquilo que contém informações relacionadas e agrupadas. Porém, como o presente trabalho trata da produção de um sistema, iremos discutir banco de dados no contexto da tecnologia.

Segundo Schimiguel (2014), SGBD é um software utilizado para gerir bases de dados, permitindo criá-las, modificá-las, eliminar bases de dados, inserir e excluir dados da mesma. Rezende (2006) Também diz: “Um sistema de gerenciamento de banco de dados (SGBD) é um software que possui recursos capazes de manipular as informações do banco de dados e

interagir com o usuário. Exemplos de SGBDs são: Oracle, SQL Server, DB2, PostgreSQL, MySQL, o próprio Access ou Paradox, entre outros.”

Figura 5 - Esquema de um SGBD



FONTE: (OLIBONI, 2016)

Para o presente trabalho foi escolhido o uso do SGBD MySQL, por se tratar de um SGBD que possui sua versão gratuita, e todas as características de um sistema robusto para criação de bases de dados.

2.5.1 MySQL

O SGBD MySQL foi o gerenciador escolhido para a construção da base de dados do presente projeto por se tratar de um Sistema poderoso, robusto, com todas as características necessárias para a construção de bases de dados e por ter *open-source*. O MySQL é desenvolvido, distribuído e mantido pela Oracle. Segundo MySQL (2019, tradução nossa) seu SGBD tem as seguintes características:

O MySQL é um sistema de gerenciamento de banco de dados (...) relacionais (...) é open source, e é muito rápido, confiável, escalável e fácil de usar. Ele funciona em sistemas cliente / servidor ou embarcados.

O MySQL Server possui um conjunto prático de recursos desenvolvidos em estreita cooperação com nossos usuários. É muito provável que o seu aplicativo ou idioma favorito suporte o servidor de banco de dados MySQL.

2.6 Desenvolvimento WEB

Com o avanço das WEB e das tecnologias que a compõe, aplicações WEB ganharam cada vez mais espaço no mercado. Segundo Ginige e Murugesan (2001, *apud* CONTE *et al.* 2005) “O rápido crescimento das aplicações Web, tanto em seu escopo quanto na extensão de seu uso, tem afetado todos os aspectos de nossas vidas .”

Por representar uma evolução do software convencional, algumas preocupações adicionais motivaram as pesquisas relacionadas à engenharia de aplicações Web (Web Engineering), mantendo o objetivo de aplicar princípios de engenharia para desenvolver aplicações Web de qualidade . Pressman (2000, *apud* CONTE *et al.* 2005).

O processo de desenvolvimento de sistemas web tem toda sua fundamentação na engenharia de software, como qualquer outro sistema, porém é importante lembrar que suas características são bem particulares e a engenharia de software comporta a essas especificidades. Sommerville (2011) fala sobre a evolução da web:

O desenvolvimento da Internet teve efeito profundo em nossas vidas. No início, a Internet era basicamente um armazenamento de informações acessível universalmente e tinha pouco efeito nos sistemas de software. Esses sistemas executavam em computadores locais e eram acessíveis apenas dentro da organização. Por volta do ano 2000, a Internet começou a evoluir, e mais e mais recursos passaram a ser adicionados aos navegadores. Isso significa que sistemas Web poderiam ser desenvolvidos e que, em vez de ter uma interface de usuário específica, poderiam ser acessados por um navegador. Isso levou ao desenvolvimento de uma enorme quantidade de novos produtos de software que ofereciam serviços inovadores e que eram acessados através da Internet.

A estrutura de uma aplicação web deve levar em conta fatores como a usabilidade do sistema e infraestrutura da web. Segundo Fraternalli e Paolini (1998, *apud* CONTE *et al.* 2005) o projeto de uma aplicação web deve considerar:

Estrutural (conceitual): define a organização das informações a serem tratadas pela aplicação e os seus relacionamentos;

Navegacional: representa como as informações serão acessadas através da aplicação, e;

Apresentação: descreve como as informações e o acesso a essas serão apresentados ao usuário da aplicação.

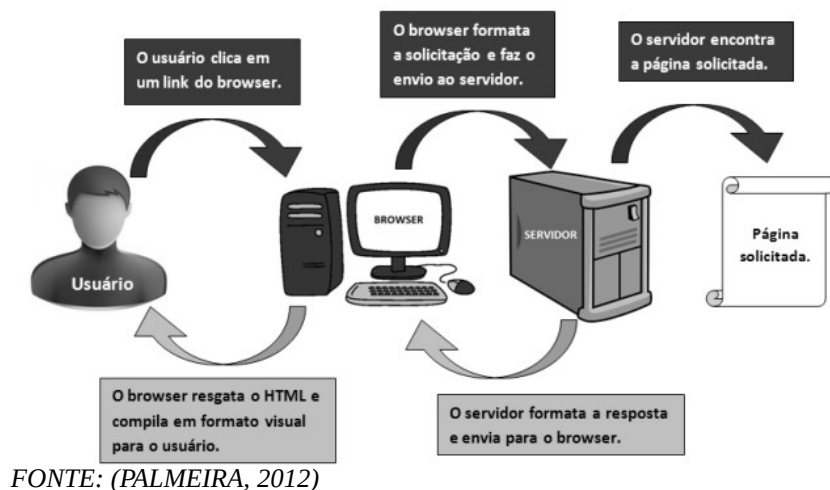
De acordo com Pastor (2004, *apud* CONTE *et al.* 2005), o objetivo principal da Engenharia Web é desenvolver aplicações corretas, nas quais suas estruturas, funcionalidades, aspectos navegacionais e de interação com o usuário estejam representados de forma apropriada.

Nos sistemas web, toda a aplicação fica alocada em um servidor web, os usuários podem ser tanto pessoas como outras aplicações. Aplicações acessam os sistemas através de API's, já pessoas utilizam-se de navegadores web como Google Chrome, Mozilla FireFox, Microsoft Edge, etc. Segundo Macêdo (2017) “Em essência, uma aplicação web é um software que é instalado em um servidor web e é projetado para responder a solicitações, processar informações, armazenar informações e dimensionar as respostas de acordo com a demanda e, em muitos casos, é distribuído em vários sistemas ou servidores”. O cliente (pessoa ou aplicação) faz uma requisição ao servidor web que recebe, processa e envia uma resposta.

Macêdo (2017) explica o modelo cliente/servidor da seguinte forma:

Um aplicativo de servidor é hospedado em um servidor web e é projetado para ser acessado remotamente por meio de um navegador da Web ou aplicativo habilitado para web. Normalmente, esse ambiente permite que vários aplicativos cliente acessem o servidor simultaneamente, para recuperar dados ou para exibir ou modificar dados. O cliente executa processamento mínimo de informações e normalmente é otimizado para apresentar as informações ao usuário. As informações são armazenadas no servidor, com algumas pequenas porções, como metadados residente no cliente.

Figura 6 - Exemplo de solicitação ao servidor web por um usuário web.



2.6.1 HTML

Criada por Tim Berners-Lee em 1990, o HTML foi um grande marco para que a internet se constituísse no que conhecemos hoje. De acordo com LUBBERS *et al.* (2011, tradução nossa). “Os anos 90 viram uma enorme quantidade de atividades em torno do HTML, com a versão 2.0, as versões 3.2 e 4.0 (no mesmo ano!) E, finalmente, em 1999, a versão 4.01. No curso de seu desenvolvimento, o World Wide Web Consortium (W3C) assumiu o controle da especificação”.

HTML (*HyperText Markup Language*, ou Linguagem de Marcação de HiperTexto) é o componente mais básico da web. Ela serve para definir o conteúdo e a estrutura básica de uma página web. *HyperText* (HiperTexto) refere-se aos *links* (ligações) que conectam uma página a outra, seja dentro de um mesmo website ou entre websites diferentes. Os links são um aspecto fundamental da web. O HTML usa *Markup* (Marcação) para mostrar textos, imagens e outros conteúdos para visualização em um navegador Internet. (MDN, 2019).

2.6.2 CSS

Para que as páginas ou aplicações web possam ser estilizadas quanto ao tipo de letra, posicionamento de elementos, cores, bordas de tabelas e outros elementos de estilo, utilizamos o CSS. Segundo MDN(2019) “CSS (*Cascading Style Sheets* ou Folhas de Estilo

em Cascata) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML [...] O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias”.

2.6.3 JavaScript

No process de construção de páginas web, muitas vezes se faz necessário construir mecanismos de interação mais complexos com o usuário, para isto usaremos o JavaScript. Segundo Flanagan (2006, tradução nossa) o JavaScript é:

Uma linguagem de programação leve e interpretada com recursos orientados a objetos. O núcleo de uso geral da linguagem foi incorporado ao Netscape, Internet Explorer e outros navegadores e embelezado para programação web com a adição de objetos que representam a janela do navegador e seu conteúdo. Esta versão do lado do cliente do JavaScript permite que o conteúdo executável seja incluído em páginas da Web - significa que uma página da Web não precisa mais ser HTML estático, mas pode incluir programas que interagem com o usuário, controlam o navegador e criam dinamicamente conteúdo HTML.

2.6.4 JQuery

Para facilitar a escrita de códigos em JavaScript, será utilizado neste trabalho a biblioteca JQuery que minimiza a quantidade de código escrito para acelerar o processo de desenvolvimento. Segundo Teixeira (2013) :

jQuery é uma biblioteca de funções em JavaScript que interage com o HTML, desenvolvida para simplificar os scripts interpretados no navegador do usuário (client-side) [...] é uma biblioteca de código aberto (*open source*) e possui licença dual, fazendo uso da Licença MIT ou da GNU General Public License versão 2. A sintaxe do jQuery foi desenvolvida para simplificar a navegação em documentos HTML, a seleção de elementos DOM, criar animações, manipular eventos, desenvolver aplicações AJAX e criação de plugins sobre ela. Permitindo aos desenvolvedores criarem camadas de abstração para interações de baixo nível de modo simplificado em aplicações web de grande complexidade.

JQuery, é uma forma de simplificar a escrita de código em JavaScript puro.

2.6.5 PHP

O PHP (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML. (PHP, 2019). Por meio dele, é possível conferir maior dinamicidade ao site, possibilitando o armazenamento de informações em bancos de dados, personalizar o conteúdo e, até mesmo, enviar e-mails e manipular arquivos. (SILVA, 2014).

Figura 7 - Exemplo de código em PHP

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>

    <?php
      echo "Olá, eu sou um script PHP!";
    ?>

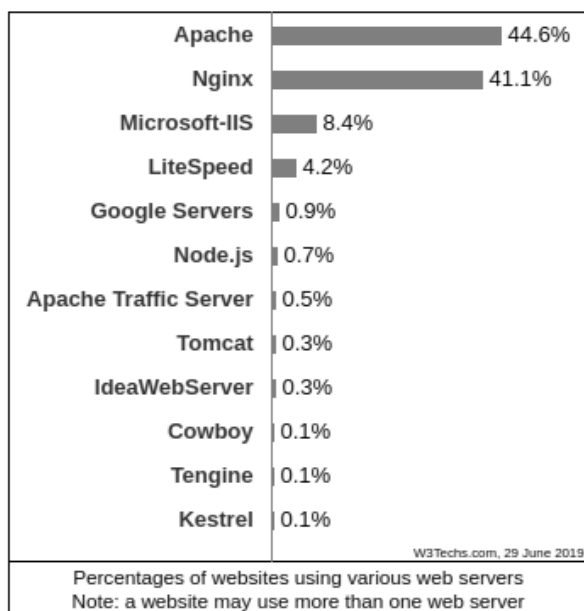
  </body>
</html>
```

FONTE: (PHP, 2019)

O PHP é uma linguagem interpretada não necessitando de um compilador, porém é necessário ter um mecanismo que consiga ler o código fonte e gerar a saída esperada pelo usuário ou então mostrar eventuais erros no arquivo que está sendo executado naquele momento.(PORTAL EDUCAÇÃO, 2019).

Então, com já vimos, o PHP é uma linguagem que necessita de um interpretador, existem vários servidores web que dão suporte ao PHP, o mais popular entre eles é o servidor Apache. Abaixo temos um gráfico de utilização do servidor apache defronte a outros servidores web:

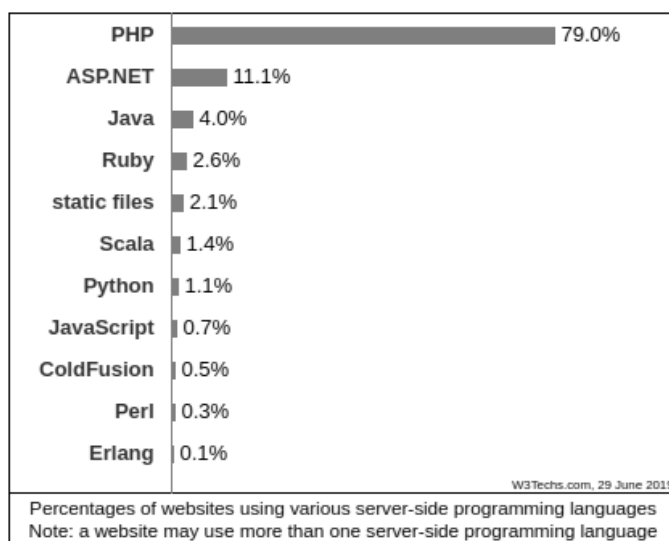
Figura 8 - Porcentagem de uso de servidores web no mundo.



FONTE: (W3TECHS, 2019)

Atualmente o PHP é a linguagem *server-side* mais presente nos *websites do mundo*. Abaixo temos um gráfico que demonstra a porcentagem de uso do PHP e outras linguagens *server-side* em *websites* no mundo:

Figura 9 - Porcentagem de uso de linguagens server-side no mundo



FONTE: (W3TECHS, 2019)

2.6.6 Composer

Gerenciadores de dependências tem a função de listar, adicionar, remover e atualizar dependências de terceiros em projetos. (VIEIRA, 2016).

O Composer, é um gerenciador de dependências para o PHP. Segundo o site do próprio composer Composer (2019, tradução nossa), o Composer é: “Uma ferramenta para gerenciamento de dependências em PHP. Ele permite que você declare as bibliotecas das quais seu projeto depende e as gerencia (instala / atualiza) para você”.

No presente projeto, Composer fará o gerenciamento das bibliotecas de terceiros que foram utilizadas assim como o *autoload* das classes por oferecer suporte nativo a esta funcionalidade.

2.6.7 Bootstrap

Bootstrap é um *framework* para auxiliar no desenvolvimento *fron-end* muito prático para se construir interfaces web. Ele foi escolhido para auxiliar a construções de alguns aspectos das páginas do sistema o qual o presente trabalho tem como objetivo. Segundo o próprio site o framework Bootstrap (2019, tradução nossa), Bootstrap é: “Um kit de ferramentas de código aberto para desenvolvimento com HTML, CSS e JS”.

2.6.8 IDE

Ambiente de Desenvolvimento Integrado (ou do inglês *Integrated Development Enviroment* – IDE) são ferramentas que visam auxiliar o programador a escrever códigos. Pires (2017) define uma IDE como “um programa que tem a função de reunir, em um único local, tudo que um desenvolvedor precisa para trabalhar”. Ele ainda cita as funcionalidades que uma IDE possui:

- 1 – um editor de textos;
- 2 – um compilador (quando necessário);
- 3 – verificadores de sintaxe;
- 4 – complementos e assistentes de código;
- 5 – prompt integrado; snippets de código;

- 6 – ferramentas de auxílio para conexão a banco de dados;
- 7 – automatizador de tarefas;
- 8 – editor de interface gráfica;

2.6.9 Visual Studio Code

O Visual Studio Code foi a IDE escolhida para o desenvolvimento do código do sistema a ser desenvolvido neste trabalho. Segundo a Própria documentação oficial da IDE, Visual Studio Code (2019, tradução nossa), define a IDE como:

O Visual Studio Code é um editor de código-fonte leve, mas poderoso, que é executado em sua área de trabalho e está disponível para Windows, macOS e Linux. Ele vem com suporte embutido para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens (como C ++, C #, Java, Python, PHP, Go) e tempos de execução (como .NET e Unity).

A IDE foi lançada em 2015 pela Microsoft.

2.6.10 Violet UML

O Violet UML foi o editor de diagramas UML escolhido para fazer os diagramas do projeto. Violet UML é um editor gratuito e multiplataforma. Segundo Jordão (2015):

O Violet UML Editor é um programa voltado a programadores que necessitem de colaboração entre uma equipe de desenvolvedores. O software utiliza a linguagem UML para a criação de diagramas. Tais diagramas são úteis para quem necessita de boa organização e um planejamento eficiente das funcionalidades e futuras implementações que ocorrerão nos aplicativos.

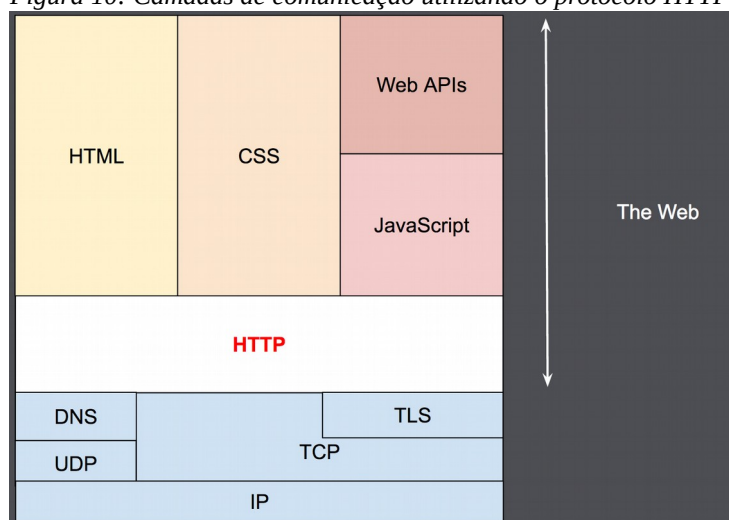
2.6.11 Protocolo HTTP

Para que a troca de informações possa ser realizada na web é necessário utilizar um protocolo de troca de dados. HTTP é um protocolo (*protocol*) que permite a obtenção de recursos, tais como documentos HTML. É a base de qualquer troca de dados na Web e um

protocolo cliente-servidor, o que significa que as requisições são iniciadas pelo destinatário, geralmente um navegador da Web. Mozilla (2019).

Ainda segundo Mozilla (2019) “Um documento completo é reconstruído a partir dos diferentes sub-documentos obtidos, como por exemplo texto, descrição do layout, imagens, vídeos, scripts e muito mais.”

Figura 10: Camadas de comunicação utilizando o protocolo HTTP



FONTE: <https://mdn.mozillademos.org/files/13673/HTTP%20%20layers.png>

2.6.12 XAMPP

A linguagem PHP é interpretada por um servidor HTTP chamado Apache. Para os testes realizados neste trabalho usaremos o ambiente de desenvolvimento XAMPP.

Segundo a Apache Friends (2019) “O XAMPP é completamente gratuito, de fácil de instalar a distribuição Apache, contendo MySQL, PHP e Perl. O pacote de código aberto do XAMPP foi criado para ser extremamente fácil de instalar e de usar.”

2.7 Conclusão

Pode-se perceber que todas as ferramentas aqui mencionadas são gratuitas. A escolha de cada uma delas foi feita segundo o critério de usabilidade, segurança e agilidade que cada uma proporciona para que este projeto pudesse ser concluído. A IDE Visual Studio Code é uma ferramenta para edição de código que possui um excelente suporte para o desenvolvimento com linguagens script como o PHP e JavaScript. Já o PHP foi escolhido por se tratar de uma linguagem sólida no mercado, quase todo servidor web proporciona suporte a esta linguagem, sua documentação é completa e possui uma comunidade de usuários enorme.

A ferramenta escolhida para criar o ambiente de desenvolvimento foi o XAMPP por ser gratuito e de fácil utilização. O gerenciador de dependências Composer nos permite gerenciar as bibliotecas de terceiros que serão utilizadas neste trabalho e proporciona um *autoload* de classes simples para evitar que tenhamos que adicionar cada classe manualmente. O SGBD MySQL oferece um poderoso manuseio das bases de dados proporcionando construir a base de dados do projeto com toda a segurança necessária.

Para o *front-end* temos as principais tecnologias: HTML5, CSS3 e JavaScript, que compõem toda a web. Será utilizado a biblioteca JQuery para que os códigos em JavaScript possam ser simplificados. O uso do *framework* Bootstrap também facilitará a construção da camada de interface com o usuário, proporcionando funções prontas utilizando-se das tecnologias *front-end*. Finalmente temos o Violet UML, um editor de diagramas que utiliza a linguagem UML, multi-plataforma e fácil de trabalhar.

CAPÍTULO 3

O ESTADO DA ARTE

Este capítulo abordará trabalhos já propostos ou parecidos com o mesmo tema deste projeto. Será mostrado suas características, suas diferenças e semelhanças com o projeto proposto por este trabalho de conclusão de curso.

3.1 “Portal de Eventos” - Um sistema para gerenciamento de eventos acadêmicos e de inscrições

Reis (2015) Apresenta um sistema capaz de realizar o gerenciamento de eventos em instituições acadêmicas. Nele o usuário consegue se cadastrar na plataforma e realizar sua inscrição nos eventos disponíveis.


O sistema permite a visualização de dados como: a quantidade de pessoas inscritas nos eventos, relação de atividades previstas e relação do número de vagas ocupadas. O sistema também possibilita a emissão de certificados de participação atestados e declarações de presença. A aplicação foi projetada para que se adequasse a quaisquer tipos de telas, sejam *smartphones*, *tablets* *NoteBooks* e *PC's*. Ele também possibilita a criação de *HotSites* para cada um dos eventos que forem cadastrados na plataforma, eliminando assim o gasto que a instituição teria ao contratar uma empresa apenas para criá-los.

Também é possível fazer a validação de presença dos participantes do evento em tempo real e *on-line*, e em caso de o evento ocorrer em uma área que não possua internet, o sistema possibilita a emissão de uma lista com as informações do participante, como nome e documento. É possível separar as atividades em grupos distintos como: palestras, oficinas, conferência, mesa-redonda, etc. A base de dados da aplicação foi feita utilizando-se o MySQL.

A aplicação permite o cadastro de 3 tipos de usuários: usuário, administrador e fiscal. Segundo Reis (2015) o usuário é a pessoa que tem interesse em participar dos eventos que são promovidos, o administrador é a pessoa que tem acesso a todas as funcionalidades que o sistema permite e o fiscal é a pessoa que fiscalizará e ter o controle dos usuários que participarão dos eventos.

Figura 11 - Tela inicial do sistema Portal Eventos

Área de Login

 Área do Usuário
Portal de Eventos

PORTAL DE EVENTOS - LOGIN

Preencha os dados abaixo:

E-mail:

Senha:

[Fazer Login](#) [Esqueceu sua senha?](#)

AINDA NÃO POSSUI CADASTRO?

Preencha os Campos abaixo para iniciar o seu cadastro:

Informe sua nacionalidade:
☒ Brasil (Informe seu CPF)
☐ Exterior (Informe seu Passaporte)

Informe seu Documento:

[Continuar Cadastro](#)

IMPORTANTE!

Você somente poderá assistir às Atividades do Evento apresentando o Documento informado no seu cadastro!

© 2015 - Portal de Eventos - INF - Instituto de Informática | UFRGS

FONTE: (REIS, 2015)

Figura 12 - Tela de cadastro do usuário do sistema Portal Eventos

Área de Login

Painel de Administração

Portal de Eventos

CADASTRO DE PARTICIPANTE

Dados do Participante:

Nome Completo:

Documento (Informe seu CPF):

Telefone (com DDD):

Data de Nascimento:

Endereço:

Número:

Complemento:

CEP:

Bairro:

Cidade:

Estado/UF:

Pais:

Informações de Acesso à Plataforma de Eventos:

Endereço de E-mail (Será utilizado para fazer login):

Senha:

Confirme a Senha:

FONTE: (REIS, 2015)

O trabalho realizado por Reis (2015) muito se assemelha ao presente trabalho pois ambos possuem cadastros de usuários com nível de acesso, validação de presença em tempo real, emissão de certificações a cerca dos eventos inscritos no sistema e a possibilidade de se gerenciar o conteúdo de cada evento.

3.2 Sistema Web para Inscrição em Atividades Acadêmicas

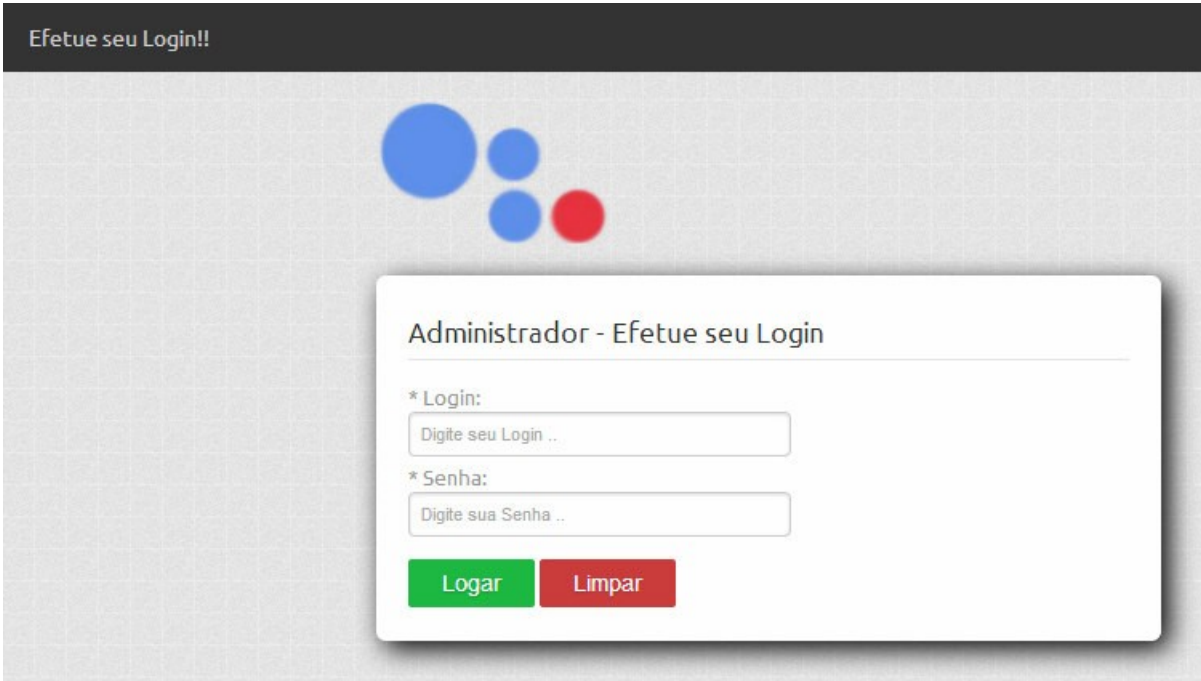
Librelato e Belusso (2013) apresentam em seu trabalho de conclusão de curso um sistema web para inscrição e gerenciamento de atividades acadêmicas. O sistema permite o gerenciamento das atividades cadastradas no mesmo, também permite que pessoas vinculadas

ou não vinculadas a instituição possam fazer o cadastro na plataforma e se inscreverem nas atividades ali listadas, já que as atividades de extensão são destinadas a toda a comunidade a qual a instituição está inserida, porém deixam claro que o sistema inicialmente foi construído e implementado para o uso interno da instituição de ensino ao qual o trabalho foi submetido.

Segundo Librelato e Belusso (2015) “para realização de inscrições em atividades de extensão ofertados pelo Departamento Acadêmico de Informática”, mas que poderia ser modificado para uso por outros departamentos ou instituições.

A aplicação conta com funcionalidades de registro de presença, emissão de certificados e controle de aproveitamento, permitindo que apenas os usuários com certa porcentagem de aproveitamento e presença possam emitir seus certificados. O sistema também apresenta distinção entre usuários aluno, professor e administrador.

Figura 13 - Tela de login do sistema proposto por Librelato e Belusso



FONTE: (LIBRELATO, P.H.; BELUSSO, W.V., 2013)

Figura 14 - Tela de cadastro de usuário do sistema proposto por Librelato e Belusso



FONTE: (LIBRELATO, P.H.; BELUSSO, W.V., 2013)

O trabalho proposto por Librelato e Belusso (2013) muito se assemelha à proposta do presente trabalho, pois trata-se de um sistema web para gerenciamento de eventos da instituição. Ambos possuem funcionalidades como gerencia de frequência dos alunos, emissão de certificados das atividades, permite que os próprios alunos cadastre-se nas atividades que preferirem e possuem login com níveis de acesso para cada tipo de usuário. Também o trabalho de Librelato e Belusso faz uso de tecnologias como PHP, arquitetura de projeto MVC e o SGBD MySQL, que também serão empregadas neste projeto.

3.3 Sistema de Atividades e Eventos

O sistema proposto por Christo e Milliati (2012) denominado SisAtos (Sistema de Atividades e Eventos) busca a automatização do processo de validação de certificados de atividades extracurriculares que compõe a carga horária obrigatória que os alunos devem apresentar junto à secretaria do curso. Segundo Christo e Milliati (2012) a validação destes documentos era feita por uma comissão formada por um professor presidente e demais integrantes que se reunia uma vez por semestre. O sistema possui funcionalidades como níveis de usuário, professores podem criar eventos na plataforma e validarem como atividades formativas. Christo e Milliati (2012) definem atividades formativas como “atividades extracurriculares que tem por objetivo ajudar na formação profissional do estudante, sendo de cunho obrigatório para o seu histórico escolar”.

No sistema o aluno também poderá enviar documentos referentes a atividades que ele já realizou para que a comissão avaliadora possa avaliá-los quando desejarem. O secretário do curso pode covalidar os documentos entregues com os documentos digitais aprovados pela comissão. Os eventos criados pelos professores dentro da plataforma serão automaticamente validados para os alunos.

Christo e Milliati (2012) aplicaram quatro níveis de acesso para o sistema que são:

- Aluno: possui como funcionalidades o cadastro de comprovantes, emissão de relatórios destas atividades, inscrição em eventos criados por professores e relatórios sobre atividades já validadas.
- Professor: poderá cadastrar novos eventos (e sub-eventos relacionados a estes) que serão validados como atividades formativas, o gerenciamento destes e no caso de professores integrantes da comissão avaliadora, a análise dos comprovantes cadastrados pelos alunos.
- Secretário: realiza o gerenciamento das informações dos Professores, Alunos, outros Secretários, atividades do curso e a composição das comissões. Porém, o seu papel principal é gerenciar a presença dos alunos nos eventos promovidos pelos professores e a convalidação dos comprovantes aprovados pela comissão.
- Administrador: possui acesso a praticamente todas as funcionalidades oferecidas pelo sistema, exceto o módulo de eventos que é gerenciado pela figura do professor e secretário. É o único perfil que pode cadastrar e gerenciar todas as informações pertinentes ao curso.

Figura 15 - Tela de login do sistema SisAtos



A screenshot of the login page for the SisAtos system. The header features the UFPR 100 logo on the left and right, with a central image of a classical building. The main content area contains a login form with two input fields: 'Usuário' and 'Senha'. Below these fields is a red arrow pointing to the 'Entrar' button. Under the 'Entrar' button, there are two links: 'Esqueceu seu usuário ou senha?' and 'Primeiro Acesso? Instruções aqui'.

FONTE: (CHRISTO, G.F.; MILLIATI, R.R.C., 2012)

Figura 16 - Tela de cadastro de eventos sistema SisAtos



A screenshot of the event registration page for the SisAtos system. The header features the UFPR 100 logo on the left and right, with a central image of a classical building. Below the header is a navigation bar with links: 'Home', 'Meus Dados', 'Aluno - Painel', and 'Logout'. Below the navigation bar is a breadcrumb trail: 'Você está aqui: Aluno - Inscrição em Evento'. The main content area contains a registration form with two input fields: 'Matricula:' (containing 'GRR20080000') and 'Evento:' (containing 'Evento de Software Livre'). Below these fields are two buttons: 'Detalhes do Evento' and 'Realizar Inscrição'. A red arrow points to the 'Detalhes do Evento' button, and another red arrow points to the 'Realizar Inscrição' button.

FONTE: (CHRISTO, G.F.; MILLIATI, R.R.C., 2012)

O trabalho proposto por Christo e Milliati (2012), se assemelha um pouco ao presente projeto no queisto de cadastro de usuários com nível de acesso, cadastro de eventos e a possibilidade de o usuário poder escolher em qual evento quer se cadastrar e por utilizar o SGBD MySQL.

3.4 Quadro de Comparações

A tabela 1 compara as funcionalidades do sistema proposto pelo presente trabalho quando confrontado com os três trabalhos apresentados nas seções 2.1, 2.2 e 2.3.

Tabela 1: Comparativo de trabalhos correlatos

	Reis	Librelato e Belusso	Christo e Milliati	Alessandro e Fernanda
Cadastro de usuários	X	X	X	X
Níveis de acesso de usuários	X	X	X	X
Cadastro de eventos	X	X	X	X
Emissão de certificados	X	X	X	X
Gestão de eventos cadastrados	X	X	X	X
Controle de frequência	X	X		X
Open Source				X

FONTE (Própria, 2019)

CAPÍTULO 4

DESENVOLVIMENTO

Neste capítulo será demonstrado como foi o processo de desenvolvimento do sistema. As técnicas utilizadas para a parte de desenvolvimento dos componentes da UML tiveram como base a obra de Cardoso (2003).

4.1 Levantamento dos requisitos

Neste tópico, abordaremos os requisitos funcionais (RF) e os requisitos não funcionais (RNF). Os requisitos foram levantados através de reunião com o pessoal responsável pela emissão de certificados na secretaria da UEMG unidade ITUIUTABA.

4.1.1 Requisitos Funcionais (RF)

RF001 – Cadastro de usuários

O sistema deverá permitir que o usuário se cadastre no sistema. Esta atividade se inicia em seu ciclo ótimo quando o usuário acessa a página inicial do sistema e lá encontra a opção de realizar o seu cadastro, e se encerra quando o usuário termina seu cadastro e recebe a mensagem de que foi cadastrado com sucesso.

Tabela 2 - Requisito funcional - cadastro de usuário

Descrição do risco	Risco	Prioridade
A implementação é conhecida e o prazo de entrega é razoável	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RF002 – Realizar login

O sistema deverá permitir que o usuário cadastrado efetue login para ter acesso as funcionalidades da plataforma. Esta atividade se inicia em seu fluxo ótimo quando o usuário

acessa a página inicial do sistema e este lhe apresenta a tela de login com nome de usuário e senha e se encerra quando o usuário tem acesso às funcionalidades do sistema.

Tabela 3 - Requisito funcional - realizar login

Descrição do risco	Risco	Prioridade
A implementação é conhecida e o prazo de entrega é razoável.	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RF003 – Cadastrar evento

O sistema deverá permitir que usuários do tipo professor e administrador cadastrem eventos na plataforma. Esta atividade se inicia em seu fluxo ótimo quando um destes dois tipos de usuário está logado e acessa a funcionalidade de **cadastrar novo evento** e se encerra quando o evento é cadastrado.

Tabela 4 - Requisito funcional - cadastrar evento

Descrição do risco	Risco	Prioridade
A implementação é um pouco conhecida e o prazo de entrega é razoável	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RF004 – Visualizar eventos

O sistema deverá permitir que o usuário visualize os eventos já cadastrados na plataforma. Esta atividade se inicia em seu fluxo ótimo quando o usuário acessa a página de eventos e se encerra quando o sistema retorna uma lista com todos os eventos disponíveis.

Tabela 5 - Requisito funcional - visualizar eventos

Descrição do risco	Risco	Prioridade
A implementação é conhecida e o prazo de entrega é razoável.	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RF006 – Inscrição no evento

O sistema deverá permitir que o usuário faça sua inscrição no evento. Esta atividade se inicia em seu fluxo ótimo quando o usuário acessa a lista de eventos disponíveis e seleciona a opção **fazer inscrição** que estará vinculada a cada evento e se encerra quando o usuário recebe uma mensagem do sistema informando que foi inscrito com sucesso.

Tabela 6 - Requisito funcional - inscrição no evento

Descrição do Risco	Risco	Prioridade
A implementação não é conhecida porém existe pessoal capacitado e o prazo de entrega é razoável.	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RF007 – Emitir certificado

O sistema deverá permitir a emissão de certificados referentes ao eventos que o usuário participou. Esta atividade se inicia em seu fluxo ótimo quando o usuário, após participar do evento e ter sua presença confirmada na plataforma, entra na lista de eventos, clica no evento que participou e nesta tela seleciona a opção **gerar certificado** e se encerra quando o sistema abre uma nova aba no navegador com o certificado pronto para ser impresso ou salvo.

Tabela 7 - Requisito funcional - emitir certificado

Descrição do Risco	Risco	Prioridade
A implementação não é conhecida e a função é crítica para o sistema porém existe pessoal capacitado. O prazo de entrega é pouco razoável	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RF008 – Realizar logoff

O sistema deverá permitir que o usuário encerre sua sessão no sistema. Esta atividade se inicia em seu fluxo ótimo quando o usuário seleciona a opção de logoff e se encerra quando o sistema encerra a sessão do usuário e retorna a tela de login.

Tabela 8 - Requisito funcional - realizar logoff

Descrição do Risco	Risco	Prioridade
A implementação é conhecida e o prazo de entrega é razoável.	Médio	Média

FONTE: (Própria, 2019)

4.1.2 Requisitos Não Funcionais (RNF)

RNF001 – Menus intuitivos

O sistema deverá possuir menus autoexplicativos para facilitar que os usuários realizem as atividades que necessitarem.

Tabela 9 - Requisito não funcional - menus intuitivos

Descrição do Risco	Risco	Prioridade
Evitar que os usuários tenham dificuldades para encontrarem as opções que desejam.	Médio	Média

FONTE: (Própria, 2019)

RNF002 – Liberação de certificados

O sistema não deverá permitir que certificados sejam emitidos para quaisquer usuários sem que toda a documentação referente ao evento esteja devidamente preenchida na plataforma.

Tabela 10 - Requisito não funcional - liberação de certificados

Descrição do Risco	Risco	Prioridade
Exigência primordial do cliente para que não faltem as informações necessárias para que o certificado tenha validade	Altíssimo	Altíssima

FONTE: (Própria, 2019)

RNF003 – Acesso a toda comunidade

O sistema deverá permitir acesso tanto para usuários vinculados à instituição quanto para aqueles que convivem no meio que ela está inserida.

Tabela 11 - Requisito não funcional - acesso a toda a comunidade

Descrição do Risco	Risco	Prioridade
O sistema não poderá barrar a inscrição de pessoas não vinculadas a Universidade.	Alto	Alta

FONTE: (Própria, 2019)

RNF004 – Acesso fora do meio acadêmico

O sistema deverá permitir o acesso às suas funcionalidades mesmo estando fora da rede na Universidade.

Tabela 12 - Requisito não funcional - acesso fora do meio acadêmico

Descrição do Risco	Risco	Prioridade
O sistema não poderá barrar o acesso fora da rede da Universidade.	Alto	Alta

FONTE: (Própria, 2019)

RNF005 – Autorização de eventos

O sistema não deverá permitir o cadastro de evento sem que o formulário específico tenha sido preenchido por completo.

Tabela 13 - Requisito não funcional - autorização de eventos

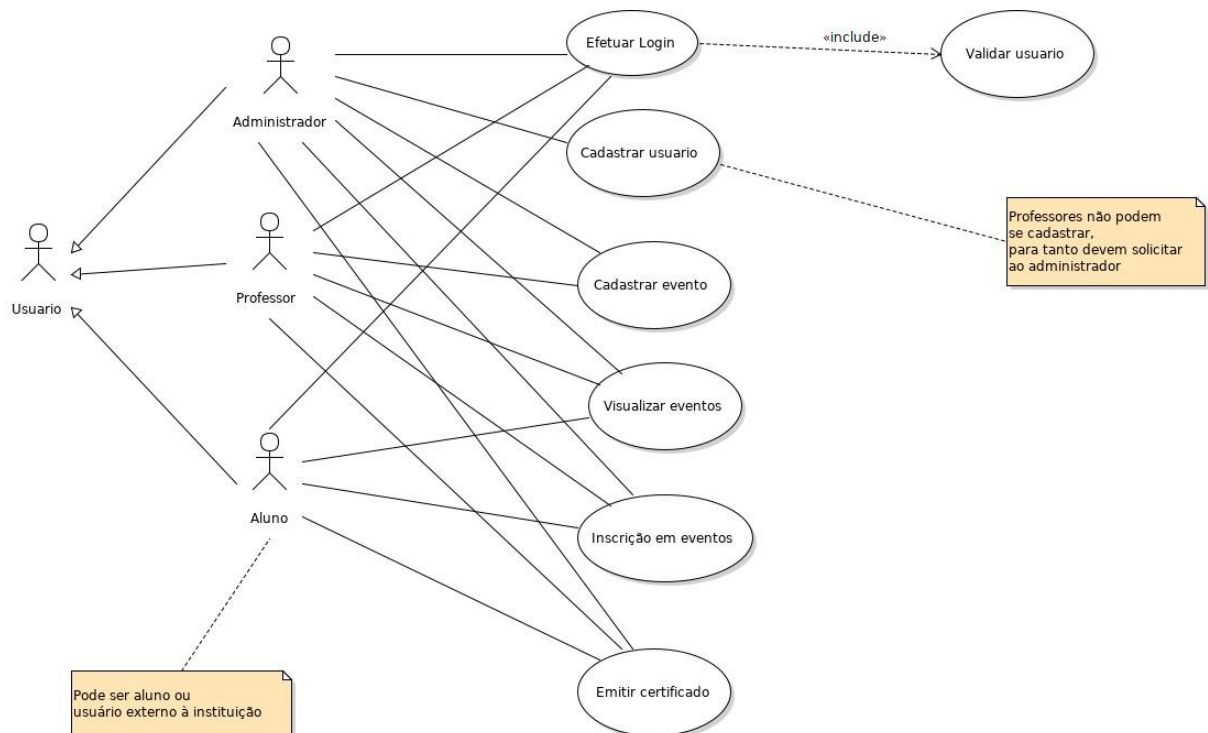
Descrição do Risco	Risco	Prioridade
O sistema não pode permitir a criação de eventos sem o devido preenchimento do formulário de criação de eventos.	Altíssimo	Altíssima

FONTE: (Própria, 2019)

4.2 Diagrama de caso de uso do sistema

A figura 17 mostra o diagrama de caso de uso que foi projetado para o sistema:

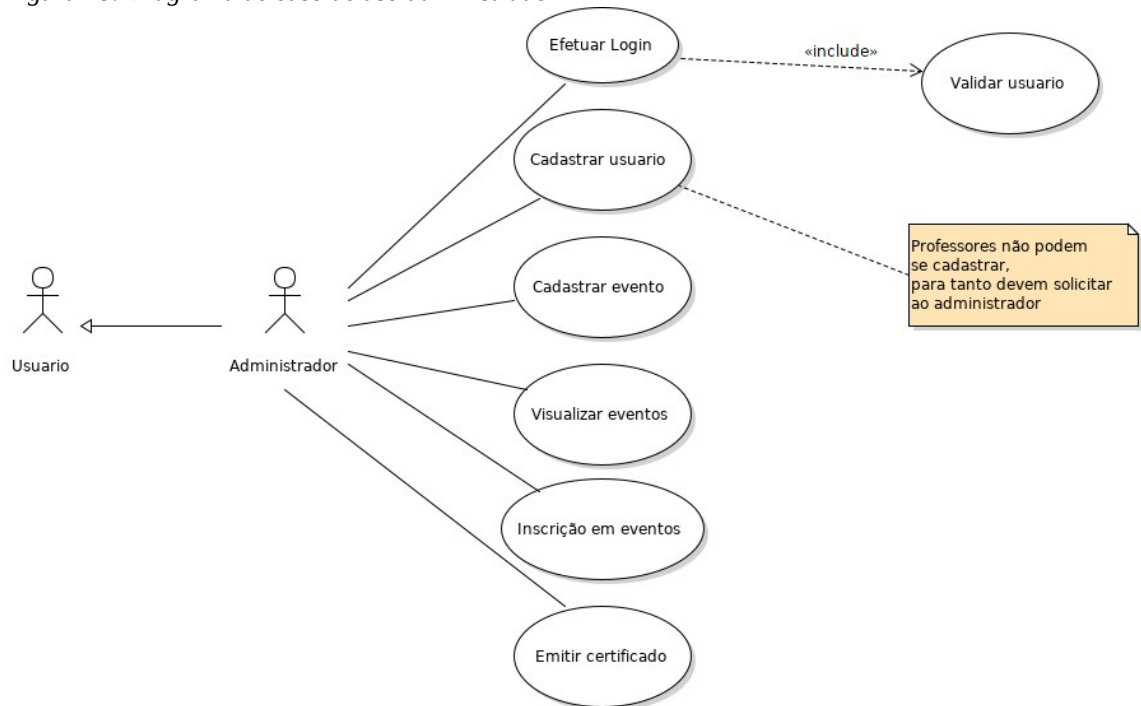
Figura 17: Diagrama de caso de uso do sistema



FONTE: (Própria, 2019)

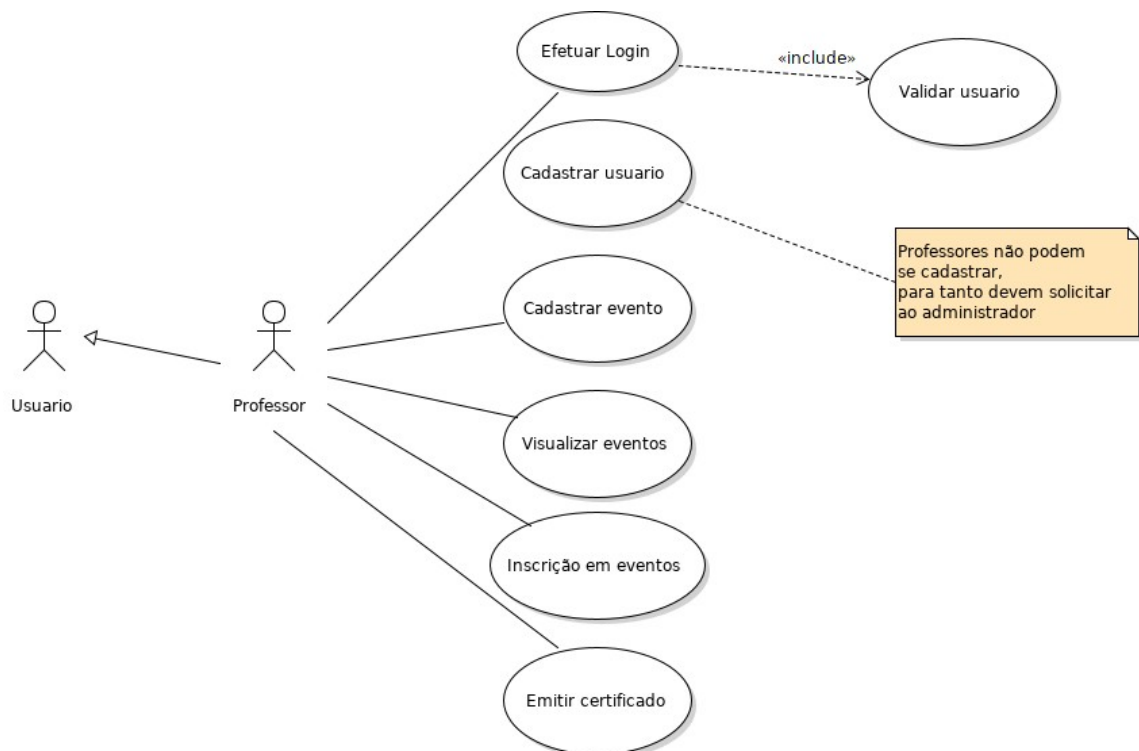
As figuras abaixo demonstram cada um dos atores do caso de uso da figura 17 para uma melhor compreensão.

Figura 18: Diagrama de caso de uso administrador



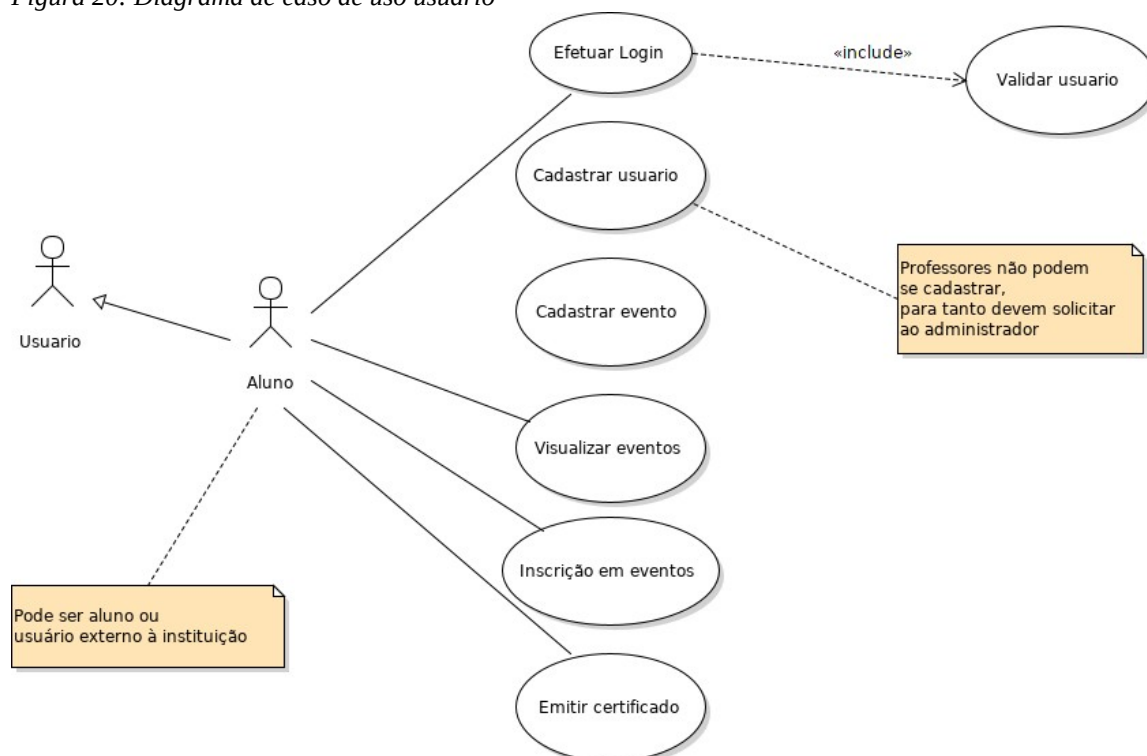
FONTE: (Própria, 2019)

Figura 19: Diagrama de caso de uso professor



FONTE: (Própria, 2019)

Figura 20: Diagrama de caso de uso usuario



FONTE: (Própria, 2019)

4.2.1 Especificação dos casos de uso

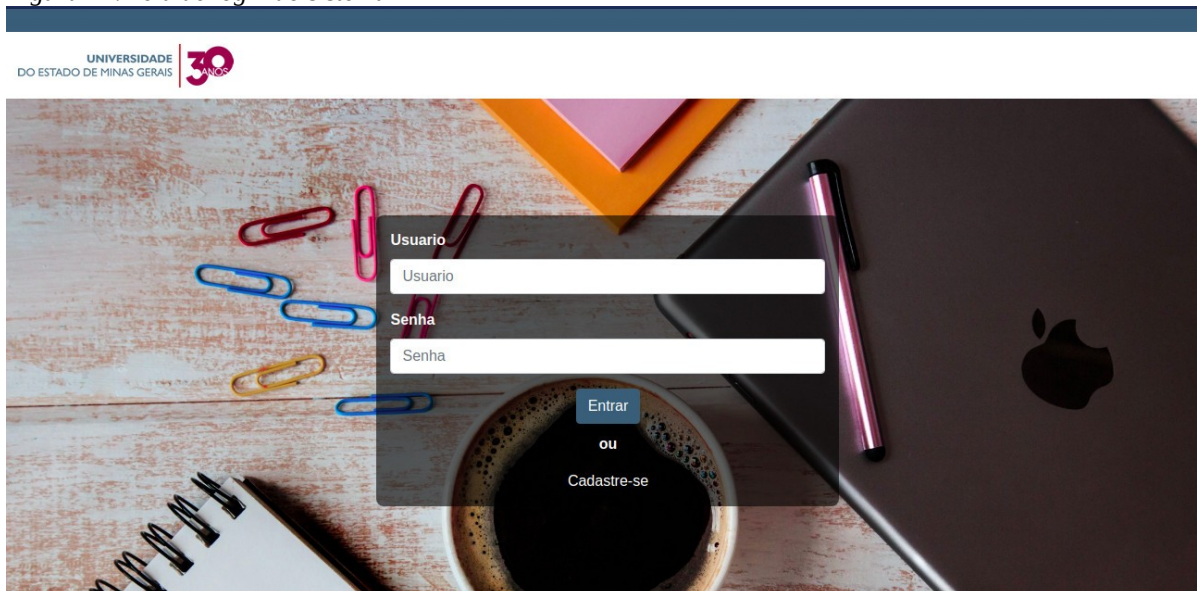
Caso de uso: Efetuar login

Descrição: Este caso de uso se inicia quando o usuário insere seu usuário e sua senha na página inicial do sistema e termina, em seu fluxo ótimo, quando o usuário tem acesso ao sistema.

Pré condição: Usuário deve possuir cadastro no sistema.

Início do caso de uso: O caso de uso inicia-se no momento em que o usuário clica no botão **Login**

Figura 21: Tela de login do sistema



FONTE (Própria, 2019)

Fluxo ótimo

Tabela 15 - Caso de uso efetuar login - fluxo ótimo

Ação Recebida	Ação Realizada
1.Usuário insere os dados e clica no botão "Login".	2.O sistema recebe os dados. 3.Valida os dados recebidos. 4.Envia mensagem de sucesso. 5.permite acesso ao usuário.

FONTE: (Própria, 2019)

Fluxos Alternativos

– Usuário não possui cadastro

Tabela 16 - Caso de uso efetuar login - fluxo alternativo – usuário não possui cadastro

Ação Recebida	Ação Realizada
1.Usuário insere os dados e clica no botão "Login".	2.O sistema recebe os dados. 3.Valida os dados recebidos. 4.Envia mensagem de usuário inexistente. 5.Direciona o usuário à página inicial.

FONTE: (Própria, 2019)

– Usuário insere a senha incorreta

Tabela 17 - Caso de uso efetuar login - fluxo alternativo – usuário insere senha incorreta

Ação Recebida	Ação Realizada
1.Usuário insere os dados e clica no botão “Login”.	2.O sistema recebe os dados. 3.Valida os dados recebidos. 4.Envia mensagem de senha ou usuário incorretos. 5.Direciona o usuário à página inicial.

FONTE: (Própria, 2019)

Caso de uso: Cadastrar usuário

Descrição: Este caso de uso se inicia quando o usuário acessa a página inicial do sistema e clica no botão **cadastre-se**, e se encerra, em seu fluxo ótimo, quando o usuário clica no botão **Cadastrar** e recebe uma notificação de que o cadastro foi realizado com sucesso e é enviado para a tela de login.

Pré condição: Nenhuma.

Início do caso de uso: O caso de uso inicia-se no momento em que o usuário clica no botão **Cadastre-se**.

Figura 22: Tela de cadastro de usuários

UNIVERSIDADE DO ESTADO DE MINAS GERAIS 30

Nome

Usuário

Sexo ☒ Masculino ☐ Feminino

Senha

Redigite a senha

CPF

E-mail

Endereço

Número

Bairro

Cidade

UF

Cep

Telefone

FONTE (Própria, 2019)

Fluxo ótimo

Tabela 18 - Caso de uso cadastrar usuário - fluxo ótimo

Ação Recebida	Ação Realizada
1.Usuário acessa a página inicial do sistema e clica no botão “Cadastre-se”.	2.O sistema envia o usuário para a tela de preenchimento de formulário de cadastro.
3.O usuário preenche as informações que são solicitadas na tela.	
4.O usuário clica no botão “Cadastrar”.	5.O sistema valida o formulário preenchido. 6.Retorna uma mensagem de sucesso ao usuário. 7.Envia o usuário para a tela de login.

FONTE: (Própria, 2019)

Fluxos Alternativos

– Usuário não preenche corretamente os dados solicitados no formulário

Tabela 19 - Caso de uso cadastrar usuário - fluxo alternativo – usuário não preenche corretamente os dados solicitados

Ação Recebida	Ação Realizada
1.Usuário acessa a página inicial do sistema e clica no botão “Cadastre-se”.	
3.O usuário preenche as informações que são solicitadas na tela.	2.O sistema envia o usuário para a tela de preenchimento de formulário de cadastro.
4.O usuário clica no botão “Cadastrar”.	5.O sistema valida o formulário preenchido. 6.Verifica que existem campos incorretos. 7.Informa o usuário o campo que está incorreto.

FONTE: (Própria, 2019)

– Usuário não preenche completamente o formulário

Tabela 20 - Caso de uso cadastrar usuário - fluxo alternativo – usuário não preenche completamente o formulário

Ação Recebida	Ação Realizada
<p>1.Usuário acessa a página inicial do sistema e clica no botão “Cadastre-se”.</p> <p>3.O usuário preenche as informações que são solicitadas na tela.</p> <p>4.O usuário clica no botão “Cadastrar”</p>	<p>2.O sistema envia o usuário para a tela de preenchimento de formulário de cadastro.</p> <p>5.O sistema valida o formulário preenchido.</p> <p>6.Verifica que existem campos não preenchidos.</p> <p>7.Informa o usuário o campo que está faltando.</p>

FONTE: (Própria, 2019)

Caso de uso: Cadastrar evento

Descrição: Este caso de uso se inicia quando o usuário já cadastrado e logado no sistema clica no botão **Eventos**, e se encerra, em seu fluxo ótimo, quando o sistema retorna uma mensagem de cadastro de evento com sucesso e lista o evento cadastrado na lista de eventos disponíveis.

Pré condição: Estar logado no sistema e ter acesso a ele como professor ou administrador.

Início do caso de uso: O caso de uso inicia-se no momento em que o usuário clica no botão

Cadastrar novo evento.

Figura 23: Tela de cadastro de eventos do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Eventos Pendências Evento principal **Cadastrar novo evento** Listar eventos validados Meus eventos Minhas inscrições

Cadastros Sair

Descrição Carga Horária

Início Fim Código Evento

Caso o evento pertença a um outro evento maior ou conjunto de evento preencha o campo **Código Evento** com o código fornecido pela aba **Evento Principal**

Tipo: ☐ Extensão ☐ Pesquisa

Informado ao colegiado do curso: ☐ Sim ☐ Não

Instituição de fomento

Outras Ocorrências:

Outras Ocorrências

Curso: Oficina/Minicurso: Tipo apresentação:

Caso não haja ocorrência para as opções **Instituição de fomento**, **Curso**, **Oficina/Minicurso** e **Tipo apresentação** deixar em branco!

Cadastrar

FONTE (Própria, 2019)

Fluxo ótimo

Tabela 21 - Caso de uso cadastrar evento - fluxo ótimo

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Cadastrar novo evento”	2.O sistema envia o usuário para a tela de preenchimento de formulário de cadastro de eventos.
3.O usuário preenche as informações que são solicitadas na tela.	5.O sistema valida o formulário preenchido.
4.Clica no botão “Cadastrar”.	6.Retorna uma mensagem de evento criado com sucesso.

FONTE: (Própria, 2019)

Fluxos Alternativos

– Usuário não preenche corretamente os dados solicitados no formulário

Tabela 22 - Caso de uso cadastrar evento - fluxo alternativo – usuário não preenche corretamente os dados solicitados no formulário

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Cadastrar novo evento”	2.O sistema envia o usuário para a tela de preenchimento de formulário de cadastro de eventos.
3.O usuário preenche as informações que são solicitadas na tela.	5.O sistema valida o formulário preenchido.
4.Clica no botão “Cadastrar”.	6.Verifica que existem campos incorretos.
	7.Informa o usuário o campo que está incorreto.

FONTE: (Própria, 2019)

– Usuário não preenche completamente o formulário

Tabela 23 - Caso de uso cadastrar evento - fluxo alternativo – usuário não preenche completamente o formulário

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Cadastrar novo evento”	2.O sistema envia o usuário para a tela de preenchimento de formulário de cadastro.
3.O usuário preenche as informações que são solicitadas na tela.	5.O sistema valida o formulário preenchido.
4.Clica no botão “Cadastrar”.	6.Verifica que existem campos não preenchidos.
	7.Informa o usuário o campo que está faltando.

FONTE: (Própria, 2019)

Caso de uso: Visualizar eventos

Descrição: Este caso de uso se inicia quando o usuário loga no sistema e clica no botão **Listar eventos validados** e se encerra, em seu fluxo ótimo, quando o sistema mostra a relação de eventos disponíveis ao usuário.

Pré condição: O usuário deve estar logado no sistema.

Início do caso de uso: O caso de uso inicia-se no momento em que o usuário clica no botão **Visualizar eventos**.

Figura 24: Tela de listagem de eventos validados

UNIVERSIDADE DO ESTADO DE MINAS GERAIS								
Eventos								
Cadastros								
Sair								
Evento principal								
Descrição								
Carga horária								
Data início								
Data fim								
Responsável								
Validado								
Permite emissão de certificado								
-								
SEMANA SISTEMAS DE INFORMAÇÃO								
09:00:00								
25/11/2019								
26/11/2019								
Alessandro de Medeiros Ferreira								
Sim								
Sim								

FONTE (Própria, 2019)

Fluxo ótimo

Tabela 24 - Caso de uso visualizar eventos - fluxo ótimo

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Listar eventos validados”.	2.O sistema direciona o usuário para a tela específica. 3.Busca os eventos disponíveis. 4.Carrega a lista de eventos disponíveis. 5.Mostra a relação de eventos disponíveis ao usuário.

FONTE: (Própria, 2019)

Fluxos Alternativos

– Não existem eventos disponíveis para o usuário

Tabela 25 - Caso de uso visualizar eventos - fluxo alternativo – não existem eventos disponíveis para o usuário

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Listar eventos validados”.	2.O sistema direciona o usuário para a tela específica. 3.Busca os eventos disponíveis. 4.Verifica que não existem. 5.Mostra tabela de eventos vazia

FONTE: (Própria, 2019)

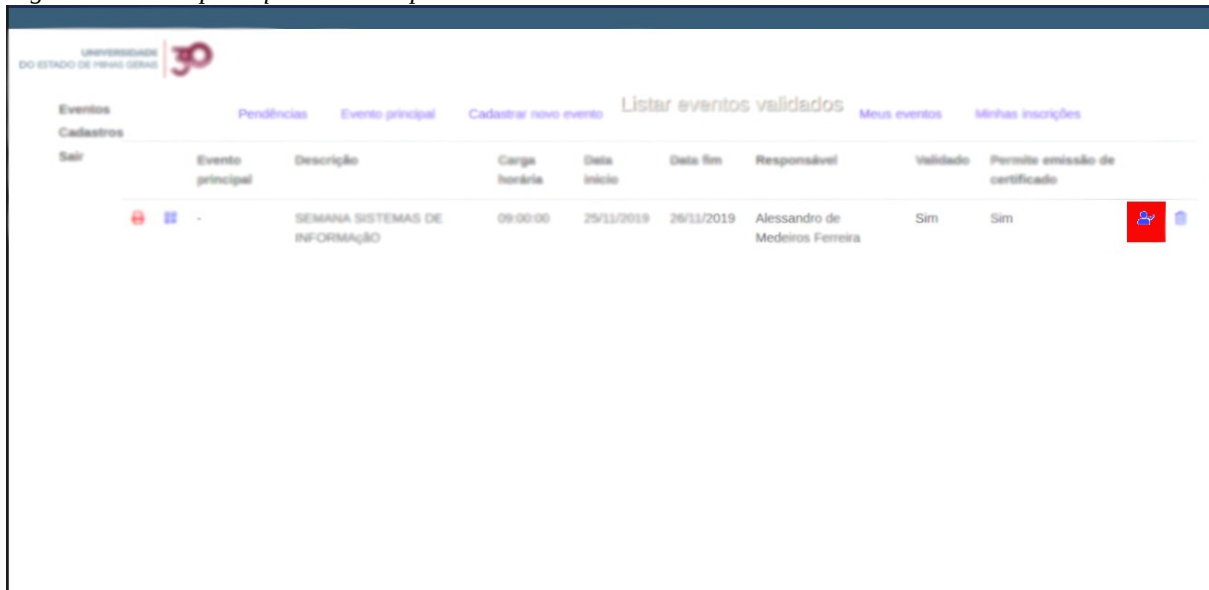
Caso de uso: Inscrição em evento

Descrição: Este caso de uso se inicia quando o usuário loga no sistema e clica no botão **Listar eventos validados** e logo em seguida clica no botão **Inscriver-se** que estará vinculado a cada um dos eventos disponíveis e se encerra, em seu fluxo ótimo quando o sistema retorna uma mensagem de “Inscrição realizada com sucesso”.

Pré condição: O usuário deve estar logado no sistema.

Início do caso de uso: O caso de uso inicia-se no momento em que o usuário clica no botão **Inscriver-se**.

Figura 25: Botão para que o usuário possa se cadastrar em um evento



FONTE (Própria, 2019)

Fluxo ótimo

Tabela 26 - Casdo de uso inscrição em eventos - fluxo ótimo

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Inscrever-se”.	2.O sistema recebe a solicitação. 3.Verifica se existe vaga disponível. 4.Cadastra o usuário. 5.Retorno mensagem de “Inscrição realizada com sucesso”.

FONTE: (Própria, 2019)

Fluxos Alternativos

– Número de vagas insuficiente

Tabela 27 - Caso de uso inscrição em evento - fluxo alternativo – número de vagas insuficiente

Ação Recebida	Ação Realizada
1.O usuário clica no botão “Inscrever-se”.	2.O sistema recebe a solicitação. 3.Verifica se existe vaga disponível. 4.Verifica que não existem vagas disponíveis. 5.Retorna mensagem de “Vagas indisponíveis para este evento”.

FONTE: (Própria, 2019)

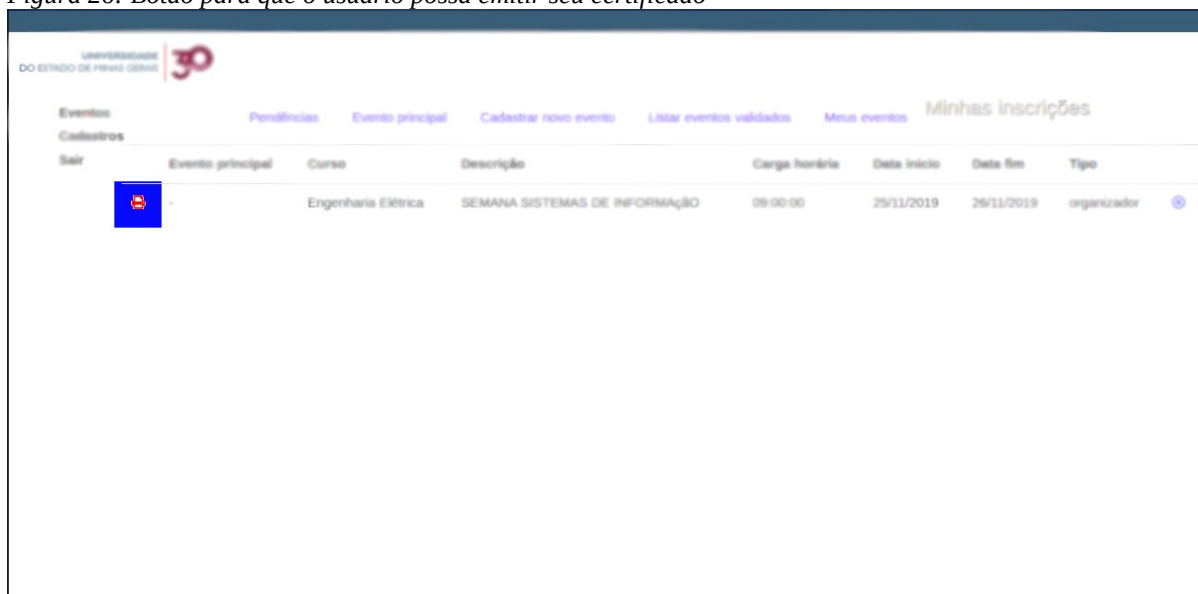
Caso de uso: Emitir certificado

Descrição: Este caso de uso se inicia quando o usuário loga no sistema e clica no botão **Minhas inscrições** e logo em seguida clica no botão **Emitir certificado** e se encerra, em seu fluxo ótimo quando o sistema abre uma nova guia no navegador com o certificado do usuário em PDF.

Pré condição: O usuário deve estar logado no sistema, inscrito no evento.

Início do caso de uso: O caso de uso inicia-se no momento em que o usuário clica no botão **Gerar certificado**.

Figura 26: Botão para que o usuário possa emitir seu certificado



FONTE (Própria, 2019)

Fluxo ótimo

Tabela 28 - Caso de uso emitir certificado - fluxo ótimo

Ação Recebida	Ação Realizada
1.Usuário clica no botão “Emitir certificado”	2.O sistema recebe a solicitação. 3.Verifica se o usuário está inscrito no evento. 4.Direciona o usuário a uma nova aba do navegador. 5.Mostra o certificado em PDF.

FONTE: (Própria, 2019)

Fluxos Alternativos

– Usuário não realizou o cadastro no evento.

Tabela 29 - Caso de uso emitir certificado - fluxo alternativo – usuário não realizou cadastro no evento

Ação Recebida	Ação Realizada
1.Usuário clica no botão “Emitir certificado”	2.O sistema recebe a solicitação. 3.Verifica se o usuário está inscrito no evento. 4.Recebe a informação de que o usuário não realizou a inscrição. 5.Retorna o usuário para a tela de visualização de eventos disponíveis.

FONTE: (Própria, 2019)

Caso de uso: Validar usuário

Descrição: Este caso de uso se inicia quando o usuário loga no sistema, uma rotina interna valida os dados preenchidos pelo usuário e se encerra, em seu fluxo ótimo com o retorno da solicitação.

Pré condição: Nenhuma, é uma rotina interna do sistema.

Início do caso de uso: assim que o botão **Entrar** é acionado.

Fluxo ótimo

Tabela 30 - Caso de uso validar usuário - fluxo ótimo

Ação Recebida	Ação Realizada
	1.O sistema recebe a solicitação. 2.Verifica na base de dados se os dados inseridos pelo usuário são válidos. 3.Envia mensagem de retorno true ou false.

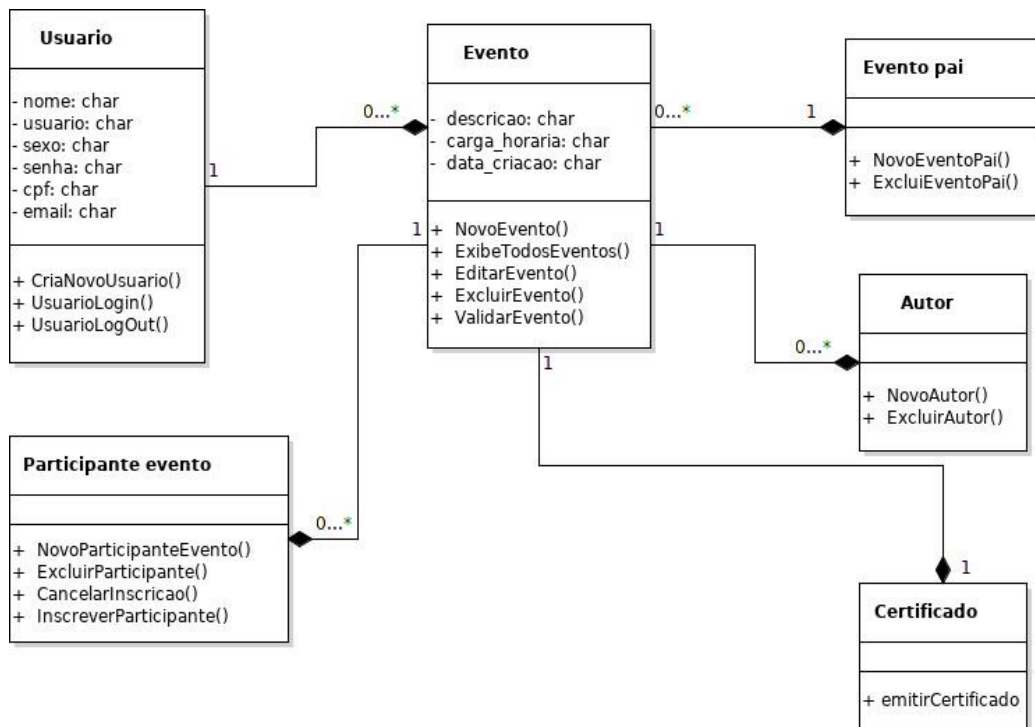
FONTE: (Própria, 2019)

4.3 Modelagem de dados

4.3.1 Diagrama de classes

A figura 21 mostra o diagrama de classes que serviu como base para dar início à modelagem dos dados.

Figura 27: Diagrama de classes do sistema

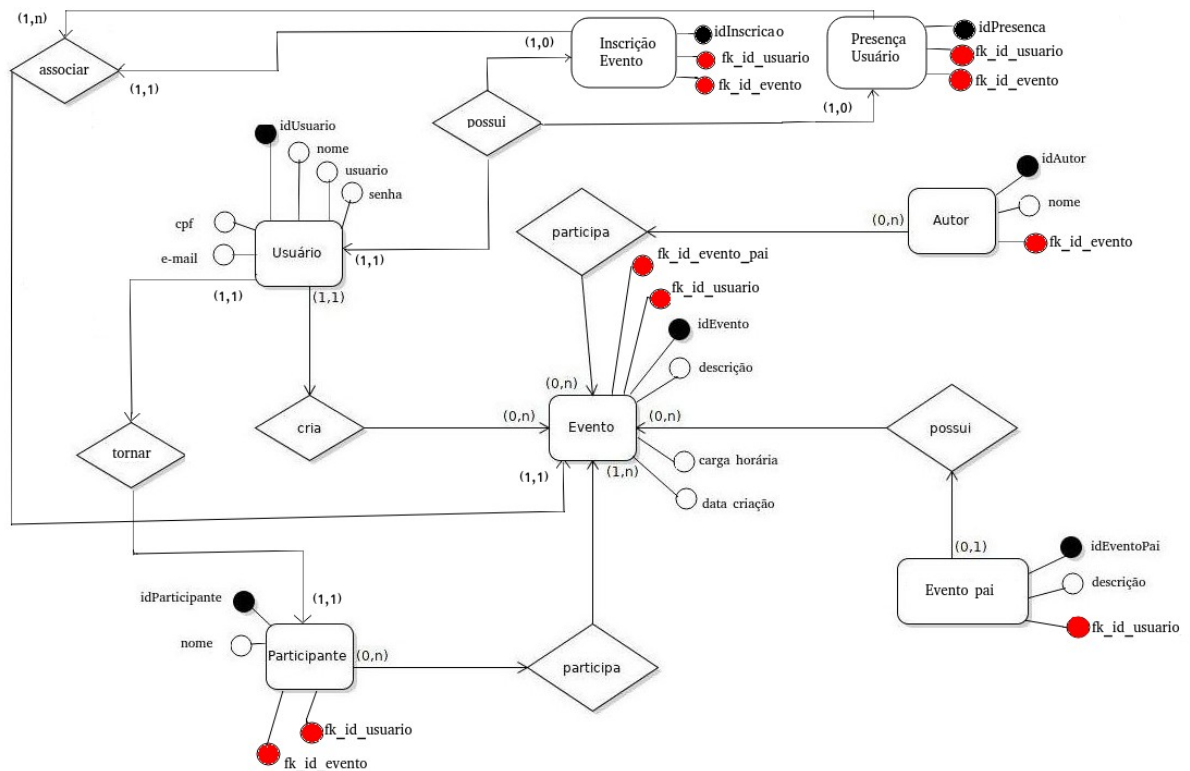


FONTE (Própria, 2019)

4.3.2 Diagrama entidade-relacionamento

A figura 22 mostra o diagrama entidade-relacionamento do sistema. O modelo conceitual é Aquele em que os objetos, suas características e relacionamentos têm a representação fiel ao ambiente observado, independentemente de quaisquer limitações imposta por tecnologias, técnicas de implementação ou dispositivos físicos. Google Sites Banco de Dados (2019).

Figura 28: Modelagem de dados: diagrama entidade-relacionamento do sistema.

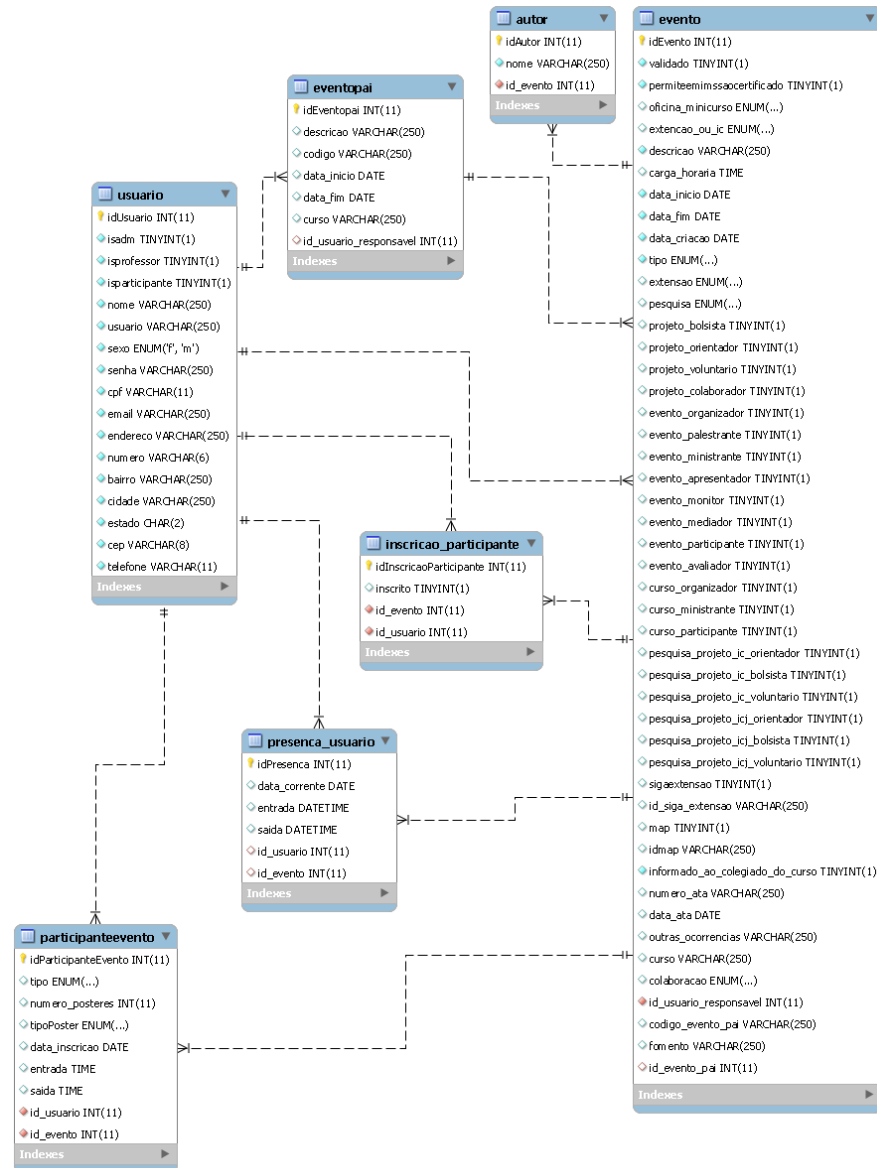


FONTE (Própria, 2019)

4.3.3 Modelo lógico da modelagem de dados

A figura 23 mostra o modelo lógico. O modelo lógico é Aquele em que os objetos, suas características e relacionamentos têm a representação de acordo com as regras de implementação e limitações impostas por algum tipo de tecnologia. Google Sites Banco de Dados(2019).

Figura 29: Modelagem de dados: modelo lógico do sistema.



FONTE (Própria, 2019).

4.3.4 Modelo físico

A figura 24, 25 e 26 mostra o modelo físico. É a representação dos objetos feita sob o foco do nível físico de implementação das ocorrências, ou instâncias das entidades e seus relacionamentos. Google Sites Banco de Dados (2019).

Figura 30: Modelagem de dados - modelo físico do sistema

```

1 create database tcc;
2
3 use tcc;
4
5 create table usuario (
6     idUsuario int(11) primary key auto_increment,
7     isadm boolean not null,
8     isprofessor boolean not null,
9     isparticipante boolean not null,
10    nome varchar(250) not null,
11    usuario varchar(250) not null unique,
12    sexo enum('f','m') not null,
13    senha varchar(250) not null,
14    cpf varchar(11) not null unique,
15    email varchar(250) not null unique,
16    endereco varchar(250) not null,
17    numero varchar(6) not null,
18    bairro varchar(250) not null,
19    cidade varchar(250) not null,
20    estado char(2) not null,
21    cep varchar(8) not null,
22    telefone varchar(11) not null
23 );
24
25
26 create table evento(
27     idEvento int primary key auto_increment,
28     validado boolean not null,
29     permiteemissao de certificado boolean not null,
30     oficina_minicurso enum ('','oficina','minicurso'),
31     extensao_ou_ic enum ('','extensionista','iniciacao Cientifica'),
32     descricao varchar(250) not null,
33     carga_horaria time,
34     data_inicio date not null,
35     data_fim date not null,
36     data_criacao date not null,
37     tipo enum('extensao','pesquisa') not null,

```

FONTE: (Própria, 2019)

Figura 31: Modelagem de dados - modelo físico do sistema

```

38  extensao enum('projeto', 'evento', 'curso', ''),
39  pesquisa enum('iniciacao cientifica', 'iniciacao cientifica junior', ''),
40  projeto_bolsista boolean,
41  projeto_orientador boolean,
42  projeto_voluntario boolean,
43  projeto_colaborador boolean,
44  evento_organizador boolean,
45  evento_palestrante boolean,
46  evento_ministrante boolean,
47  evento_apresentador boolean,
48  evento_monitor boolean,
49  evento_mediador boolean,
50  evento_participante boolean,
51  evento_avaliador boolean,
52  curso_organizador boolean,
53  curso_ministrante boolean,
54  curso_participante boolean,
55  pesquisa_projeto_ic_orientador boolean,
56  pesquisa_projeto_ic_bolsista boolean,
57  pesquisa_projeto_ic_voluntario boolean,
58  pesquisa_projeto_icj_orientador boolean,
59  pesquisa_projeto_icj_bolsista boolean,
60  pesquisa_projeto_icj_voluntario boolean,
61  sigaextensao boolean,
62  id_siga_extensao varchar(250),
63  map boolean,
64  idmap varchar(250),
65  informado_ao_colegiado_do_curso boolean not null,
66  numero_ata varchar(250),
67  data_ata date,
68  outras_ocorrencias varchar(250),
69  curso varchar(250),
70  colaboracao enum('coordenacao', 'curso', ''),
71  id_usuario_responsavel int not null, /*Id do usuario que criou o evento*/
72  codigo_evento_pai varchar(250),
73  fomento varchar(250),
74  id_evento_pai int /*id do evento pai*/

```

FONTE: (Própria, 2019)

Figura 32: Modelagem de dados - modelo físico do sistema

```

77 create table eventopai(
78     idEventopai int primary key auto_increment,
79     descricao varchar(250),
80     codigo varchar(250) unique,
81     data_inicio date,
82     data_fim date,
83     curso varchar(250),
84     id_usuario_responsavel int
85 );
86
87 /*ATENÇÃO!!!! Monitores e palestrantes terão que ter seu cadastro feito como um usuario seja de qual natureza for para evitar cadastros
desnecessários e uma tabela somente para cadastrar esses usuarios*/
88 create table participanteevento(
89     idParticipanteEvento int primary key auto_increment,
90     tipo enum('orientador','bolsista','voluntario','colaborador','organizador','monitor','palestrante','mediador',
91     'debatedor','ministrante','ouvinte','apresentador','avaliador',''),
92     numero_posteres int,
93     tipoPoster enum('extensao','iniciacaoCientifica',''),
94     data_inscricao date,
95     entrada time,
96     saida time,
97     id_usuario int not null,
98     id_evento int not null
99 );
100 create table autor(
101     idAutor int primary key auto_increment,
102     nome varchar(250) not null,
103     id_evento int not null
104 );
105
106 create table inscricao_participante(
107     idInscricaoParticipante int primary key auto_increment,
108     inscrito boolean,
109     id_evento int not null,
110     id_usuario int not null
111 );

```

FONTE: (Própria, 2019)

Figura 33: Modelagem de dados - modelo físico do sistema

```

113 create table presenca_usuario(
114     idPresenca int primary key auto_increment,
115     data_corrente DATE,
116     entrada DATETIME,
117     saida DATETIME,
118     id_usuario int,
119     id_evento int
120 );
121
122 /*FOREIGN KEYS*/
123 alter table eventopai add constraint fk_id_usuario_responsavel_eventopai foreign key(id_usuario_responsavel) references usuario(idUsuario) on
delete cascade;
124
125 alter table evento add constraint fk_id_usuario_responsavel_evento foreign key(id_usuario_responsavel) references usuario(idUsuario) on delete
cascade;
126
127 alter table evento add constraint fk_id_evento_pai foreign key(id_evento_pai) references eventopai(idEventopai) on delete cascade;
128 alter table participanteevento add constraint fk_id_usuario_participanteevento foreign key(id_usuario) references usuario(idUsuario) on delete
cascade;
129
130 alter table participanteevento add constraint fk_id_evento_participanteevento foreign key(id_evento) references evento(idEvento) on delete
cascade;
131
132 alter table autor add constraint fk_id_evento_autor foreign key(id_evento) references evento(idEvento) on delete cascade;
133
134 alter table inscricao_participante add constraint fk_id_evento_inscricao_participante foreign key(id_evento) references evento(idEvento) on
delete cascade;
135
136 alter table inscricao_participante add constraint fk_id_usuario_inscricao_participante foreign key(id_usuario) references usuario(idUsuario)
on delete cascade;
137
138 alter table presenca_usuario add constraint fk_id_usuario_presenca_usuario foreign key(id_usuario) references usuario(idUsuario) on delete
cascade;
139
140 alter table presenca_usuario add constraint fk_id_evento_presenca_usuario foreign key(id_evento) references evento(idEvento) on delete
cascade;
141

```

FONTE: (Própria, 2019)

CAPÍTULO 5

IMPLEMENTAÇÃO

A implementação do sistema constituiu-se de 3 etapas: codificação do banco de dados, codificação da parte de controle e regras de negócio (*back-end*) e codificação da interface de interação com o usuário (*front-end*). Como já mencionado antes no capítulo 2 o SGBD escolhido foi o MySQL, portanto todo o código desta etapa foi escrito em linguagem SQL padrão deste SGBD. Toda a parte de controle e regras de negócio foi escrita em PHP. Três arquivos são responsáveis por fazerem o controle de acesso do usuário, sendo eles: `paineldecontrole.php`, `painelprofessor.php` e `painelusuario.php`, cada um deles representa um dos três tipos de usuários do sistema: administrador, professor e usuário respectivamente.

A validação de acesso é feita por meio de `SESSIONS` (variável super global do PHP) com as senhas dos usuários sendo armazenadas no banco de dados e criptografadas com o método nativo do PHP chamado `password_hash`. O painel de controle de cada usuário sempre verifica a existência de uma sessão com os dados do usuário, se houver esta sessão e os dados coincidirem com o que está no banco de dados o sistema permite o acesso.

As ações de cada usuário dentro do sistema são passadas por meio de URLs com variáveis de controle que fazem uma chamada para o próprio arquivo de controle de cada tipo de usuário que, dependendo do valor que a variável assumir, permitirá acesso ao usuário a determinada funcionalidade.

A camada de interface com o usuário utiliza o Bootstrap com o seu sistema de grids para o posicionamento dos itens na tela e suas classes para os botões e formulários de preenchimento e também foi utilizado JavaScript para a estilização de efeitos e formulários dinâmicos, como no caso do formulário de cadastro de evento.

Os ícones utilizados no sistema ficaram por conta da biblioteca Font Awesome uma biblioteca que possui vários ícones gratuitos. A IDE utilizada foi o Visual Studio Code pois é gratuito e proporciona um ótimo editor de código para linguagens como PHP e JavaScript.

Não foi utilizado uma arquitetura padrão para o projeto pois foi de comum acordo que para a natureza e tamanho do presente projeto um sistema de divisão de pastas simples seria de melhor entendimento e organização, ficando o sistema de pastas como descrito na figura 27.

O sistema utiliza duas bibliotecas de terceiros que são: FPDF para a geração dos PDF's dos certificados e a biblioteca CHILLERLAN para a geração de um QRCode para cada evento, e para fazer o gerenciamento das versões destas bibliotecas assim como também fazer o *auto-load* das classes nos vários arquivos foi utilizado o gerenciador de dependências COMPOSER que foi descrito no capítulo 2 deste trabalho.

Foi utilizado o XMAPP versão para Linux como servidor local para os teste do sistema. Todo o código do sistema está hospedado no GitHub e está aberto sob licença MIT.

Figura 27: Estrutura de diretórios do sistema a partir do diretório raiz



FONTE (Própria, 2019)

CAPÍTULO 6

RESULTADOS

Ao final do trabalho foi obtido um sistema WEB com interface amigável e totalmente funcional pronto para ser implantado no ambiente de produção. A figura 28 mostra como o banco de dados ficou estruturado no SGBD MySQL:

Figura 28: Estrutura do banco de dados do sistema

Tabela	Ações	Registos	Tipo	Agrupamento (Collation)	Tamanho	Suspensão
autor	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	utf8mb4_general_ci	32 KB	-
evento	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	utf8mb4_general_ci	32 KB	-
eventopai	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	1	InnoDB	utf8mb4_general_ci	48 KB	-
participanteevento	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	0	InnoDB	utf8mb4_general_ci	48 KB	-
usuario	Procurar Estrutura Pesquisar Inserir Limpar Eliminar	2	InnoDB	utf8mb4_general_ci	64 KB	-
5 tabelas	Soma	3	InnoDB	utf8mb4_general_ci	224 KB	0 Bytes

Nome: Número de colunas:

Executar

FONTE (própria, 2019)

As figuras 30 a 40 mostram as telas que o sistema possui:

Figura 30: Tela de cadastro de usuário comum do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Nome

Senha

CPF

Endereço

Cidade

Telefone

Usuário

Redigite a senha

E-mail

Número

UF

Sexo ☒ Masculino ☐ Feminino

Bairro

Cep

FONTE (Própria, 2019)

Figura 30: Painel de controle do administrador - tela de boas-vindas do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Eventos

Cadastros

Sair

Bem vindo **Alessandro de Medeiros Ferreira** este é seu painel de controle!

Use o menu a sua esquerda para fazer a administração dos eventos cadastrados nesta plataforma!

FONTE (Própria, 2019)

Figura 32: Painel de controle do administrador - tela de eventos com pendências do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Pendências

Evento principal | Cadastrar novo evento | Listar eventos validados | Meus eventos | Minhas inscrições

Eventos

Cadastros

Sair

Evento principal	Curso	Descrição	Carga horária	Data início	Data fim	Responsável	Validado	Permite emissão de certificado
------------------	-------	-----------	---------------	-------------	----------	-------------	----------	--------------------------------

FONTE (Própria, 2019)

Figura 33: Painel de controle do administrador - tela de cadastro de evento principal do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Evento principal

Pendências | Cadastrar novo evento | Listar eventos validados | Meus eventos | Minhas inscrições

Eventos

Cadastros

Sair

Descrição do evento

Data início dd/mm/aaaa Data fim dd/mm/aaaa

Curso

Cadastrar

Descrição	Data Inicio	Data Fim	Codigo	Responsável
-----------	-------------	----------	--------	-------------

localhost/sistema-emissao-certificado/painelcontrole.php?a...

FONTE (Própria, 2019)

Figura 34: Painel de controle do administrador - tela de cadastro de evento individual do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Eventos [Pendências](#) [Evento principal](#) [Cadastrar novo evento](#) [Listar eventos validados](#) [Meus eventos](#) [Minhas inscrições](#)

Cadastros

Sair

Descrição Carga Horária

Início Fim Código Evento

Caso o evento pertença a um outro evento maior ou conjunto de evento preencha o campo **Código Evento** com o código fornecido pela aba **Evento Principal**

Tipo: ☒ Extensão ☐ Pesquisa

Informado ao colegiado do curso: ☒ Sim ☐ Não

Instituição de fomento

Outras Ocorrências:

Outras Ocorrências

Curso: Oficina/Minicurso: Tipo apresentação:

Caso não haja ocorrência para as opções **Instituição de fomento**, **Curso**, **Oficina/Minicurso** e **Tipo apresentação** deixar em branco!

[Cadastrar](#)

FONTE (Própria, 2019)

Figura 35: Painel de controle do administrador - tela de eventos validados do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Eventos [Pendências](#) [Evento principal](#) [Cadastrar novo evento](#) [Listar eventos validados](#) [Meus eventos](#) [Minhas inscrições](#)

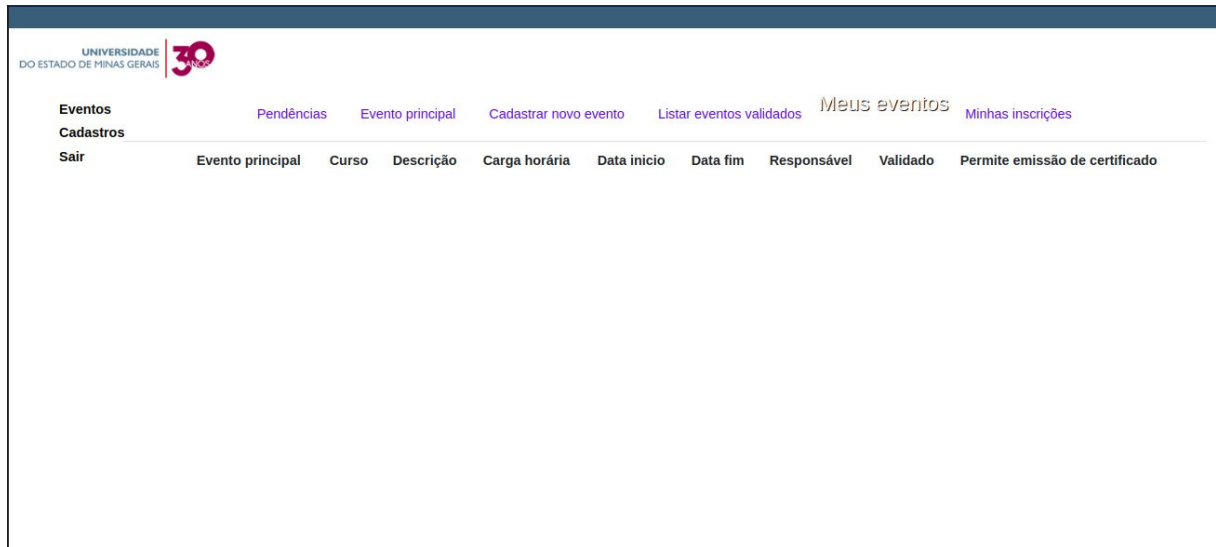
Cadastros

Sair

Evento principal	Descrição	Carga horária	Data início	Data fim	Responsável	Validado	Permite emissão de certificado
------------------	-----------	---------------	-------------	----------	-------------	----------	--------------------------------

FONTE (Própria, 2019)

Figura 36: Painel de controle do administrador - tela de eventos que foram criados pelo usuário



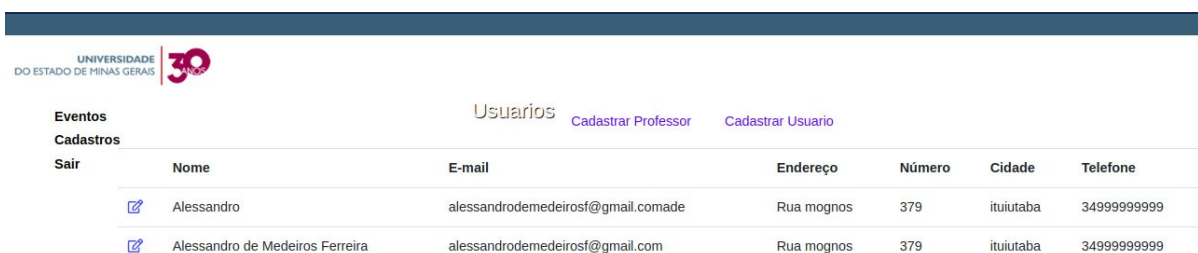
FONTE (Própria, 2019)

Figura 37: Painel de controle do administrador - tela de eventos em que o usuário está inscrito do sistema



FONTE (Própria, 2019)

Figura 38: Painel de controle do administrador - tela de usuários cadastrados do sistema



Sair	Nome	E-mail	Endereço	Número	Cidade	Telefone
	Alessandro	alessandrodemedeirosf@gmail.comade	Rua mognos	379	ituiutaba	34999999999
	Alessandro de Medeiros Ferreira	alessandrodemedeirosf@gmail.com	Rua mognos	379	ituiutaba	34999999999

FONTE (Própria, 2019)

Figura 39: Painel de controle do administrador - tela de cadastro de professores do sistema



Nome
 Usuario

Sexo ☒ Masculino ☐ Feminino

Senha
 Redigite a senha

CPF
 E-mail

Endereço
 Número

Bairro
 Cidade

UF
 Cep

Telefone

FONTE (Própria, 2019)

Figura 40: Painel de controle do administrador - tela de cadastro de usuários do sistema

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Eventos
Cadastros
Sair

Usuarios Cadastrar Professor Cadastrar Usuario

Nome Usuario

Sexo ☒ Masculino ☐ Feminino

Senha Redigite a senha

CPF E-mail

Endereço Número

Bairro Cidade

UF Cep

Telefone

FONTE (Própria, 2019)

CAPÍTULO 7

CONCLUSÃO

Os objetivos propostos no presente trabalho foram alcançados com êxito. O sistema está pronto para ser implantado em um ambiente de produção para que assim possa ser usado por qualquer pessoa do setor administrativo das universidades que se interessarem pelo projeto. Atualmente o sistema esta configurado com temas e cursos da Universidade do Estado de Minas Gerais – UEMG – unidade Ituiutaba/MG pois foi o local da concepção, testes e apresentação do projeto como um trabalho de conclusão de curso para a obtenção do grau de Bacharel dos alunos envolvidos no mesmo. O sistema também possui uma rotina que cria um QRCode para cada evento criado na plataforma, o intuito deste QRCode era a implantação de uma aplicação mobile para o controle de frequência, porém a mesma não pode ser desenvolvida por conta do prazo de entrega do sistema principal.

7.1 Principais contribuições

Este projeto foi desenvolvido para automatizar o processo de emissão de certificados acadêmicos concluindo com êxito o seu propósito.

7.2 Dificuldades

Houveram algumas dificuldades na concepção de como seria estruturado o banco de dados e em como o sistema trataria as requisições de ações feitas a ele, isto demandou bastante tempo de estudo dos participantes do projeto.

7.3 Trabalhos futuros

Para trabalhos futuros, foi idealizado a criação de uma Aplicação Mobile para a validação e controle de presença dos participantes dos eventos que fará este controle através da leitura de um QRCode.

BIBLIOGRAFIA

Apache Friends. **O que é o XAMPP?** Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em: 21 de novembro de 2019.

BOOTSTRAP. **Bootstrap**. Bootstrap, 2019, Disponível em: <<https://getbootstrap.com/>>. Acesso em: 29 de Junho de 2019.

BUDIMAN, Edy et al. Eloquent object relational mapping models for biodiversity information system. In: **2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)**. IEEE, 2017. p. 1-5.

CARDOSO, Caique. **UML na Prática: do problema ao sistema**. 1ª ed. Rio de Janeiro: Ciência Moderna Ltda, 2003.

CHRISTO, Gustavo Fernando; MILLIATI, Ray Ridlav Câmara. **Sistema de Atividades e Eventos**. Universidade Federal do Paraná. Curitiba, 2012.

CONTE, Tayana; MENDES, Emília; TRAVASSOS, Guilherme Horta. **Processos de desenvolvimento para aplicações web: Uma revisão sistemática**. In: Proceedings of the 11th Brazilian Symposium on Multimedia and Web (WebMedia 2005). 2005. p. 107-116.

DE ESPINDOLA, Rodrigo Santos; MAJDENBAUM, Azriel; AUDY, Jorge Luis Nicolas. **Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software**. In: **WER**. 2004. p. 226-238.

ELMASRI, Ramez et al. **Sistemas de banco de dados**. 2005.

FILHO, W. de P. P. **Engenharia de software**. [S.l.]: LTC, 2003. v. 2.

FLANAGAN, David. **JavaScript: the definitive guide**. 4 ed. O'Reilly Media, Inc., 2006.

Google Sites. **3.2 Níveis de Modelagem**. Disponível em: <https://sites.google.com/site/fkbancodedados1/modelodados/niveismodelagem>. Acesso em : 21 de novembro de 2019.

JAQUES, Rafael. **O que é um Framework?** Para que serve? Phpfit, 2016. Disponível em: <http://www.phpfit.com.br/artigos/o-que-e-um-framework.phpfit>. Acesso em: 23 de Junho de 2019.

JORDÃO, Fábio . **Violet UML Editor**, Baixaki, 2015. Disponível em: <https://www.baixaki.com.br/download/violet-uml-editor.htm>. Acesso em: 29 de Junho de 2019.

LIBRELATO, Pedro Henrique; BELUSSO, Wellison Victor. **Sistema Web Para Inscrição em Atividades Acadêmicas**. Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

LOTH, A. F; PRETO, L. S; OLIVEIRA, R. A. M; ZSCHORNACK, T. **As Tendências e Desafios da WEB 3.0 à Luz da Gestão do Conhecimento**. RISUS - Journal on Innovation and Sustainability, São Paulo, Vol. 10, Nº 1, Pgs 37 a 47, 2019.

LUBBERS, Peter et al. **Pro HTML5 programming**. New York, NY, USA.: Apress, 2011.

MACÊDO, Diego. **Entendendo as Aplicações Web**. Diego Macedo, 2017. Disponível em: <<https://www.diegomacedo.com.br/entendendo-as-aplicacoes-web/>>. Acesso em: 29 de Junho de 2019.

MACORATTI, José Carlos. **Padrões de Projeto: O modelo MVC – Model View Controller**. Macoratti.net, 2002. Disponível em: <http://www.macoratti.net/vbn_mvc.htm>. Acesso em: 29 de Junho de 2019.

MACORATTI, José Carlos. **UML: Conceitos Básicos II**. Macoratti.net, 2019. Disponível em: <http://www.macoratti.net/vb_uml2.htm>. Acesso em: 27 de Junho de 2019.

MCCOOL, Shawn. **Laravel Starter**. Packt Publishing Ltd, 2012.

MDN Web docs. **CSS**. Mozilla, 2019. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 25 de Junho de 2019.

MDN Web docs. **HTML**. Mozilla, 2019. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 25 de Junho de 2019.

MDN Web Docs. **Uma visão geral do HTTP**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>>. Acesso em: 21 de novembro de 2019.

MEDEIROS, Higor. **Introdução ao padrão MVC**. DevMedia, 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>. Acesso em: 29 de Junho de 2019.

MYSQL. **What is MySQL?** MySQL, 2019. Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>>. Acesso em: 29 de Junho de 2019.

NARDI, Julio Cesar; DE ALMEIDA FALBO, Ricardo. **Uma Ontologia de Requisitos de Software**. In: CIBSE. 2006. p. 111-124.

OLIBONI, Daniel. **O que é um SGBD?** Oficina da Net, 2016. Disponível em: <<https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd>>. Acesso em: 29 de Junho de 2019.

PALMEIRA, Thiago Vinícius Varallo. **Como funcionam as aplicações web**. DevMedia, 2012. Disponível em: <<https://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888>>. Acesso em: 29 de Junho de 2019.

PHP. **O que é o PHP?** Php.net. Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 22 de Junho de 2019.

Portal Educação. **Introdução ao PHP:** linguagem interpretada. Portal da Educação, 2019. Disponível em: <<https://www.portaleducacao.com.br/conteudo/artigos/informatica/introducao-ao-php-linguagem-interpretada/31642>>. Acesso em: 23 de Junho de 2019.

RAMOS, Allan. **O que é MVC?** Tableless, 2015. Disponível em: <<https://tableless.com.br/mvc-afinal-e-o-que/>>. Acesso em: 29 de Junho de 2019.

REIS, Daniel Bruno de Castro. **“Portal Eventos” – Um sistema para gerenciamento de eventos acadêmicos**. Instituto de Informática – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2015.

REZENDE, Ricardo. **Conceitos Fundamentais de Banco de Dados**. DevMedia, 2006. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 29 de Junho de 2019.

ROBERTO, João. **O que é Laravel?** Porque usá-lo? Medium, 2017. Disponível em: <<https://medium.com/joaorobertopb/o-que-%C3%A9-laravel-porque-us%C3%A1-lo-955c95d2453d>>. Acesso em: 24 de Junho de 2019.

SANTOS, Jéssica Ferreira. **Avaliação de critérios de seleção de software especializado para automação de unidades de informação: estudo comparativo entre os softwares Sophia, Pergamum, Alexandria, Aleph e Biblivre**. 2016.

SAUVÉ, Jacques. **Diagramas de Colaboração**. Universidade Federal de Campina Grande, 2003. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/apoo/html/proj1/proj4.htm>>. Acesso em: 27 de Junho de 2019.

SCHIMIGUEL, Juliano. **Gerenciamento de Banco de Dados: Análise comparativa de SGBD's**. DevMedia, 2014. Disponível em: <<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788>>. Acesso em: 29 de Junho de 2019.

SILVA, Júlia Marques Carvalho. **PHP na Prática: 200 exercícios resolvidos**. 1 ed. Rio de Janeiro: Elsevier, 2014.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª ed. São Paulo: Pearson, 2011.

TEIXEIRA, José Ricardo. **JQuery Tutorial**. DevMedia, 2013. Disponível em: <<https://www.devmedia.com.br/jquery-tutorial/27299>>. Acesso em: 29 de Junho de 2019.

VIEIRA, Rafael. **Gerenciadores de dependências.** Disponível em: <<https://medium.com/@nvieirarafael/gerenciadores-de-depend%C3%AAscias-bed99b01e94a>>. Acesso em: 21 de novembro de 2019.

Visual Studio Code. **Getting Started.** Microsoft, 2019. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 29 de Junho de 2019.

W3TECHS. **Usage of server-side programming languages for websites.** W3Techs, 2019. Disponível em: <https://w3techs.com/technologies/overview/programming_language/all>. Acesso em: 29 de Junho de 2019.

W3TECHS. **Usage of web servers.** W3Techs, 2019. Disponível em: <https://w3techs.com/technologies/overview/web_server/all>. Acesso em: 29 de Junho de 2019.