



**INSTITUTO FEDERAL
SANTA CATARINA**

**CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA
TECNOLOGIA DA INFORMAÇÃO**

TALISSA REGINA OCCHI BUCKELL

**Protótipo de uma aplicação para a
Web: Semana Nacional de Ciência e
Tecnologia do IFSC Câmpus
Florianópolis**

**Florianópolis - SC
2017**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA
CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA
INFORMAÇÃO**

TALISSA REGINA OCCHI BUCKELL

**PROTÓTIPO DE UMA APLICAÇÃO PARA A WEB: SEMANA NACIONAL DE
CIÊNCIA E TECNOLOGIA DO IFSC CÂMPUS FLORIANÓPOLIS**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Gestão da Tecnologia da Informação.

Professor Orientador:
Prof. Herval Daminelli, Esp.

**FLORIANÓPOLIS – SC
DEZEMBRO DE 2017**

FICHA DE IDENTIFICAÇÃO

Buckell, Talissa Regina Occhi
Protótipo de uma aplicação para a web: Semana Nacional de
Ciência e Tecnologia do IFSC Câmpus Florianópolis/ Talissa
Regina Occhi Buckell ; orientação de Herval Daminelli -
Florianópolis, SC, 2017.
79 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de
Educação, Ciência e Tecnologia de Santa Catarina, Câmpus
Florianópolis. CST em Gestão da TI. Departamento Acadêmico de
Saúde e Serviços.
Inclui Referências.

1. Desenvolvimento 2. Aplicação web 3. Gestão de eventos I.
Daminelli, Herval II. Instituto Federal de Educação, Ciência
e Tecnologia de Santa Catarina. Departamento Acadêmico de
Saúde e Serviços. III. Título.

**PROTÓTIPO DE UMA APLICAÇÃO PARA A WEB: SEMANA NACIONAL DE
CIÊNCIA E TECNOLOGIA DO IFSC CÂMPUS FLORIANÓPOLIS**

TALISSA REGINA OCCHI BUCKELL

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 06 de dezembro de 2017.



Prof. Julio Cesar da Costa Ribas, Dr.
Coordenador do CST em Gestão da Tecnologia da Informação
Instituto Federal de Santa Catarina

Banca Examinadora:



Prof. Herval Daminelli, Esp.
Orientador
Instituto Federal de Santa Catarina



Prof. Elenir Aparecida Crestani Lisot, Me.
Instituto Federal de Santa Catarina



Prof. Adriano Heis, Me.
Instituto Federal de Santa Catarina

Dedico a realizaço deste trabalho à minha famlia, meus amigos e a todos que me apoiaram durante esta trajetria.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pela força, sabedoria, paciência e aos meus pais, Jose e Jussara, e minha irmã Thais, pelo amor, compreensão e apoio durante toda a vida; ao meu namorado Rafael, pelo amor, por todo o incentivo para que eu voltasse a estudar e pelo apoio durante toda essa jornada; a todos os demais familiares e amigos pelo carinho e apoio e aos colegas e professores pelo incentivo e compreensão, em especial ao meu orientador, prof. Herval, por todo o carinho, dedicação e conhecimento, que contribuíram para que este projeto fosse concluído.

*“São as nossas escolhas, mais do que as nossas capacidades,
que mostram quem realmente somos.”*

J. K. Rowling, escritora

RESUMO

Este trabalho apresenta o desenvolvimento do protótipo de uma aplicação para a *web* para a Semana Nacional de Ciência e Tecnologia do Instituto Federal de Santa Catarina, Câmpus Florianópolis, desenvolvido utilizando-se HTML, CSS, PHP e JavaScript, com banco de dados MySQL. O protótipo tem como objetivo auxiliar a gestão das informações dos eventos realizados, automatizando processos antes executados de forma manual, facilitando a comunicação e evitando retrabalhos. Constitui-se, também, como ferramenta para a gestão da informação e possibilita a tomada de decisões importantes em relação ao evento.

Palavras-chave: Desenvolvimento. Aplicação *web*. Gestão de eventos.

ABSTRACT

This work presents the development of a prototype of an application for a web for a Semana Nacional de Ciência e Tecnologia of the Instituto Federal de Santa Catarina, Camps Florianópolis, developed using HTML, CSS, PHP and JavaScript, with MySQL database. The prototype aims to assist in the management of the information of the events performed, automating processes previously performed manually, facilitating communication and avoiding rework. It is also a tool for information management and enables important decisions to be taken in relation to the event.

Key-words: *Development. Web application. Event management.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Tipos de requisitos não funcionais	24
Figura 2 - Diagramas da UML	26
Figura 3 - Exemplo de diagrama de casos de uso	28
Figura 4 - Exemplo de classe	29
Figura 5 - Modelo de arquitetura em camadas para aplicações cliente-servidor	37
Figura 6 - Comunicação cliente-servidor na internet	38
Figura 7 - Diagrama de casos de uso da aplicação	43
Figura 8 - Diagrama de classes.....	51

LISTA DE TABELAS

Tabela 1 - O padrão cliente-servidor	36
Tabela 2 - Documentação do caso de uso "Acessar área administrativa"	44
Tabela 3 - Documentação do caso de uso "Gerenciar cadastros internos"	45
Tabela 4 - Documentação do caso de uso "Gerenciar atividades"	46
Tabela 5 - Documentação do caso de uso "Gerar lista de inscritos"	47
Tabela 6 - Documentação do caso de uso "Gerar certificados dos participantes"	48
Tabela 7 - Documentação do caso de uso "Realizar inscrição"	49
Tabela 8 - Documentação do caso de uso "Acompanhar inscrição"	50

LISTA DE ABREVIATURAS E SIGLAS

CSS	<i>Cascading Style Sheets</i> (Folhas de Estilo em Cascata)
IFSC	Instituto Federal de Ciência e Tecnologia de Santa Catarina
HTML	<i>HyperText Markup Language</i> (Linguagem de Marcação de Hipertexto)
HTTP	<i>HyperText Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
MCTIC	Ministério da Ciência, Tecnologia, Inovações e Comunicações
PHP	PHP: <i>HyperText Preprocessor</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SNCT	Semana Nacional de Ciência e Tecnologia
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
URL	<i>Uniform Resource Locator</i> (Localizador Uniforme de Recursos)
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Justificativa	14
1.2 Definição do problema	15
1.3 Objetivo Geral	16
1.4 Objetivos Específicos	16
2 METODOLOGIA.....	18
2.1 Classificação da pesquisa	18
2.2 Métodos aplicados	19
2.3 Organização do documento	19
2.4 Definição das variáveis da pesquisa	20
2.5 Delimitação da pesquisa	20
3 REVISÃO DA LITERATURA	20
3.1 Engenharia de <i>software</i>	21
3.1.1 PROCESSOS DE <i>SOFTWARE</i>	21
3.2 Engenharia de requisitos.....	22
3.2.1 LEVANTAMENTO E ANÁLISE DE REQUISITOS	22
3.2.1.1 Requisitos funcionais	23
3.2.1.2 Requisitos não funcionais.....	23
3.2.2 UML	24
3.2.2.1 Itens	25
3.2.2.2 Relacionamentos.....	25
3.2.2.3 Diagramas	25
3.2.2.3.1 Diagrama de casos de uso	27
3.2.2.3.2 Diagrama de classes	28
3.3 Aplicação <i>web</i>	29
3.4 Tecnologias	30
3.4.1 LINGUAGENS DE PROGRAMAÇÃO.....	30
3.4.2 HTML.....	31
3.4.3 CSS	31
3.4.4 JAVASCRIPT.....	32
3.4.5 PHP	32
3.4.6 BANCOS DE DADOS.....	33

3.4.7 ARQUITETURA CLIENTE-SERVIDOR	35
3.4.8 FRAMEWORKS.....	39
4 DESENVOLVIMENTO	40
4.1 Análise do sistema	40
4.2 Levantamento de requisitos	40
4.2.1 REQUISITOS FUNCIONAIS.....	40
4.2.2 REQUISITOS NÃO FUNCIONAIS	41
4.3 Estudo comparativo com aplicações já existentes	41
4.4 Tecnologias utilizadas	42
4.4.1 DIAGRAMA DE CASOS DE USO.....	43
4.4.2 DIAGRAMA DE CLASSES	50
4.4.3 MODELAGEM DE DADOS.....	51
4.4.4 DESENVOLVIMENTO DO PROTÓTIPO.....	52
4.4.5 INTERFACE GRÁFICA.....	52
5 CONCLUSÃO.....	53
REFERÊNCIAS.....	55
GLOSSÁRIO	58
APÊNDICE A – MODELAGEM DE DADOS.....	59
APÊNDICE B – CODIFICAÇÃO DA TELA INDEX.PHP.....	60
APÊNDICE C – CODIFICAÇÃO DA TELA CADASTRO_INSCRITOS.PHP	64
APÊNDICE D – CODIFICAÇÃO DA TELA CADASTRO_ATIVIDADES.PHP	71
APÊNDICE E – INTERFACE GRÁFICA DA TELA INDEX.PHP	76
APÊNDICE F – INTERFACE GRÁFICA DA TELA CADASTRO_INSCRITOS.PHP	77
APÊNDICE G – INTERFACE GRÁFICA DA TELA CADASTRO_ATIVIDADES.PHP	78
APÊNDICE H – DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO ...	79

1 INTRODUÇÃO

O ser humano possui a necessidade de se comunicar e é através da comunicação que se torna possível externar pensamentos e demonstrar competências, sentimentos e experiências.

Em um mundo onde a possibilidade de se comunicar é cada vez mais fácil e rápida, torna-se importante utilizar ferramentas úteis na estratégia de comunicação de uma organização. Uma dessas ferramentas utilizadas é o evento. Simões (1995, p.170) afirma que evento é “um acontecimento criado com a finalidade específica de alterar a história da relação organização-público, em face das necessidades observadas”. Um evento permite que diferentes públicos interajam, incitando o interesse pela participação da vida em comunidade.

Através do Decreto de 9 de junho de 2004, foi criada a Semana Nacional de Ciência e Tecnologia (SNCT), que é realizada sempre no mês de Outubro, com a coordenação do Ministério da Ciência, Tecnologia, Inovações e Comunicações (MCTIC), contando com a colaboração de diversas entidades. De acordo com o *site*¹ oficial do evento, a SNCT tem “o objetivo de aproximar a Ciência e Tecnologia da população, promovendo eventos que congregam centenas de instituições a fim de realizarem atividades de divulgação científica em todo o país.” Ainda, segundo o *site* oficial, o evento tem a ideia de criar uma linguagem que seja de fácil entendimento por toda a população, estimulando a curiosidade e motivando todos a analisar a Ciência na sociedade e aprofundar seus conhecimentos sobre o tema.

A 14ª SNCT, realizada de 23 a 29 de outubro de 2017, propôs o tema “A Matemática está em tudo”, baseando-se no fato de que a Olimpíada Internacional de Matemática e o Congresso Internacional de Matemáticos serão realizados pela primeira vez no Brasil entre 2017 e 2018.

A SNCT é realizada em diversas unidades do Instituto Federal de Ciência e Tecnologia de Santa Catarina (IFSC). No câmpus Florianópolis, as atividades são realizadas dentro das dependências do próprio estabelecimento, organizadas pelo departamento de Diretoria de Pesquisa e Extensão do câmpus Florianópolis. Nesse

¹ <http://snct.mctic.gov.br/portal>

evento, são propostas diversas atividades durante dois dias, ligadas ou não ao tema sugerido, para a comunidade interna e externa do IFSC Florianópolis.

Pelas experiências dos eventos anteriores, notou-se a necessidade de controlar as informações produzidas. Com base nisso, foi sugerida a criação de um protótipo de aplicação para a *web*, destinado a controlar, gerir e automatizar o fluxo das informações envolvidas na SNCT, como, por exemplo, os dados de cada participante e os dados das atividades que seriam ofertadas. Com essa aplicação, é possível solucionar os problemas encontrados, implementando-se a automação para o gerenciamento das informações e aplicando-se estratégias de uso da TI para garantir o melhor desempenho dos eventos e satisfação dos usuários envolvidos.

1.1 Justificativa

Com o decorrer da história, a humanidade desenvolveu técnicas de se comunicar de forma rápida e eficiente, possibilitando diferentes formas de registrar as informações produzidas, auxiliando na construção do conhecimento de quem utiliza essa informação.

Diante de um mundo informacional inserido no paradigma das Tecnologias de Informação e Comunicação (TICs), as organizações são protagonistas de um ambiente que se caracteriza por uma grande produção informacional, e que precisa ser organizada para seu acesso e uso. Assim, a informação precisa ser gerenciada, com vistas à construção do conhecimento organizacional e, como consequência, a possibilidade de a organização responder mais rapidamente às demandas de um mercado em constante transformação.

(VALENTIM, 2010, p. 144)

É necessário destacar a diferença entre dado, informação e conhecimento. Dado é o elemento de forma bruta; informação é o dado capaz de produzir conhecimento; e conhecimento é o entendimento produzido por uma informação. Segundo Alter (1992), dados são formatados, filtrados e manipulados para gerar informação. A transformação desses dados em informação é baseada em conhecimento sobre como formatar, filtrar e manipular dados para serem favoráveis em uma determinada ocasião.

Assim, para haver um gerenciamento eficiente e eficaz das informações na organização, faz-se necessário estabelecer procedimentos de maneira estruturada, que forneça aos gestores condições de desempenho,

conforme o processo de gestão definido pela instituição. O mecanismo encontrado para fornecer este suporte gerencial é o sistema de informação. (BEUREN; MARTINS, 2001, p. 1)

Com base nessas definições, identificou-se a necessidade de utilizar a informação de forma digital, para que possa ser gerenciada por um sistema e seja transformada em conhecimento, utilizando-a de forma estratégica.

Apesar dessas limitações, esses bancos de dados e informação, se adequadamente estruturados, fornecem informação relevante para quem o está acessando e, portanto, auxiliam na construção do conhecimento por parte do usuário. Nesse sentido, defende-se a importância de a informação digital ser gerenciada eficientemente pelos sistemas de informação corporativos, com o objetivo de propiciar o acesso e a recuperação de dados e informações consistentes, bem como que sua utilização seja transformada em conhecimento e em vantagem competitiva. (VALENTIM, 2010, p. 144)

A proposta deste projeto é criar um protótipo de aplicação para a *web* que possa gerenciar algumas das informações vinculadas à SNTC, permitindo que os interessados façam inscrições antecipadamente, para que se possa controlar a quantidade de participantes em cada atividade e proporcionar um local adequado para cada uma delas, otimizando os processos, eliminando os procedimentos manuais e evitando a ocorrência de erros. Com a aplicação também é possível gerar relatórios, bem como declarações de frequência e certificados de forma automatizada.

1.2 Definição do problema

O departamento de Diretoria de Pesquisa e Extensão do IFSC, Câmpus Florianópolis identificou diversos problemas durante a organização do evento nos anos anteriores. Com base nessas experiências, sem uma aplicação para gerenciar as informações relacionadas à data, foram encontradas muitas dificuldades, mais especificamente, relacionadas à:

- identificação antecipada dos participantes de cada atividade, não sendo possível dimensionar o porte do evento, impedindo que se reservem ambientes com tamanho suficiente para comportar os presentes;
- confirmação da presença dos participantes no evento. A forma encontrada pelo departamento consistiu-se em uma lista manuscrita com os dados dos presentes, o que gerou, em diversas ocasiões, uma aglomeração de pessoas diante dos locais das atividades, aguardando

o preenchimento de seus dados na lista de presença. Em outras ocasiões, para evitar a aglomeração, a lista foi passada de participante em participante durante a atividade, para que os mesmos preenchessem seus dados;

- emissão de declarações de frequência e certificados para todos os participantes. Com relação à certificação, a dificuldade foi quanto ao processo manual de emissão de cada certificado, sendo necessária a alteração de cada informação (dados do participante e dados da atividade), depois de confirmada a presença do participante. Com base no último evento realizado, foram gerados manualmente mais de 2.500 certificados, levando cerca de 4 meses para o procedimento.

Baseando-se nos problemas levantados, surgem as seguintes questões: como é possível melhorar os processos relacionados à SNCT? Como os dados gerados podem ser utilizados de maneira mais eficiente na gestão do evento?

1.3 Objetivo Geral

Criar um protótipo de aplicação para *web* destinado a automatizar algumas das atividades relacionadas à SNCT do IFSC – Câmpus Florianópolis.

1.4 Objetivos Específicos

Os objetivos específicos do presente trabalho são:

- compreender os processos da SNCT;
- reconhecer a importância das diversas tecnologias necessárias para a criação e o desenvolvimento de um protótipo para a *web*;
- entender a comunicação de dados entre uma aplicação na linguagem PHP e um banco de dados MySQL;
- utilizar a arquitetura cliente-servidor, que viabiliza uma aplicação para a *web*;
- descrever alguns métodos e técnicas que facilitem a criação de uma aplicação para a *web*;

- estabelecer os requisitos necessários para que a aplicação atue como ferramenta de gestão para o auxílio à tomada de decisão.

2 METODOLOGIA

Cervo e Bervian (2004) apud. Kahlmeyer-Mertens et al.¹ descreve que metodologia científica “é o estudo dos métodos de conhecer, de buscar o conhecimento. É uma forma de pensar para se chegar à natureza de um determinado problema, seja para explicá-lo ou estudá-lo.” O autor ainda afirma que “A pesquisa constitui seu principal instrumento ou meio de acesso”.

Para Descartes (1999) apud. Kahlmeyer-Mertens et al.², “método é a ordem dos elementos de um processo para atingir um fim”.

Nas palavras de Barros e Lehfeld (1986) apud. Kahlmeyer-Mertens et al.³, “a metodologia não procura soluções, mas escolhe as maneiras de encontrá-las”.

A pesquisa faz parte do processo educativo. O pesquisador só tem oportunidade de fazê-la à medida que compreende e domina uma série de técnicas e de conhecimento. Sua edificação e seu aprimoramento são conquistas que o pesquisador obtém ao longo de seus estudos, da realização de investigações e da elaboração de trabalhos acadêmicos.
(KAHLMAYER-MERTENS et al.)⁴

2.1 Classificação da pesquisa

De acordo com Selltiz et al. (1967) apud. Gil (2002, p. 41), a pesquisa deste projeto é definida como exploratória quanto ao seu objetivo, pois envolve levantamento bibliográfico, entrevistas com pessoas que tiveram experiências práticas com o problema e análise de exemplos que estimulam a compreensão.

Segundo Prodanov e Freitas⁵, sob o ponto de vista de sua natureza, a pesquisa pode ser classificada como aplicada, pois “objetiva gerar conhecimentos para a aplicação prática dirigidos à solução de problemas específicos. Envolve verdades e interesses locais.”

Para Prodanov e Freitas⁶, do ponto de vista da forma de abordagem do problema, a pesquisa pode ser considerada qualitativa, porque “[...] esta não requer o uso de métodos e técnicas estatísticas. O ambiente natural é a fonte direta para a coleta de dados e o pesquisador é o instrumento-chave.”

¹ KAHLMAYER-MERTENS et al., 2007, p. 15

² Id.

³ Ibid., p. 16

⁴ Ibid., p. 24

⁵ PRODANOV; FREITAS, 2013 p. 51

⁶ Ibid., p. 70

Sob o ponto de vista do procedimento técnico utilizado, a pesquisa, de acordo com Yin (2001) apud. Gil (2002, p.54), pode ser considerada como estudo de caso, porque é o “[...] delineamento mais adequado para investigação de um fenômeno contemporâneo dentro de seu contexto real, onde os limites entre o fenômeno e o contexto não são claramente percebidos.”

2.2 Métodos aplicados

Foi realizado um estudo comparativo das ferramentas já existentes, relacionadas à gestão de eventos acadêmicos e em geral. Além desse estudo, este trabalho foi baseado na pesquisa bibliográfica, mais notadamente no que se refere aos métodos e técnicas de criação de uma aplicação *web*.

O levantamento dos dados foi realizado através de entrevistas com a equipe do Departamento de Pesquisa e Extensão do IFSC, de acordo com a disponibilidade dos entrevistados.

O desenvolvimento do protótipo da aplicação, bem como os diagramas da UML e modelagem de dados, foi efetivado através de ferramentas gratuitas disponíveis na internet.

2.3 Organização do documento

Ao longo do capítulo 3, são apresentados os conceitos teóricos considerados como base no decorrer do trabalho. São abordadas questões como: engenharia de software e seus processos, engenharia de requisitos, UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada, em português) e a descrição das tecnologias utilizadas no desenvolvimento do protótipo, como linguagens de programação e marcação, bancos de dados, arquitetura cliente-servidor e *frameworks*.

O capítulo 4 descreve os procedimentos realizados no desenvolvimento do protótipo da aplicação para a *web*, tais como: análise do sistema, estudo comparativo com as ferramentas já existentes, levantamento de requisitos, diagramas da UML, modelagem de dados e o desenvolvimento do código do protótipo.

2.4 Definição das variáveis da pesquisa

Para o desenvolvimento do protótipo da aplicação é necessário analisar os requisitos do sistema, sejam requisitos funcionais ou não funcionais. Também é necessário que o protótipo utilize-se de elementos como desempenho, usabilidade, confiabilidade, segurança e disponibilidade, fundamentais para o desenvolvimento de uma aplicação para a web.

A codificação do protótipo procura respeitar os padrões estabelecidos pela W3C (*World Wide Web Consortium*).

2.5 Delimitação da pesquisa

A pesquisa consiste no desenvolvimento de um protótipo de aplicação para a web, que será utilizado durante a SNCT pelo Departamento de Pesquisa e Extensão do IFSC, Câmpus Florianópolis, bem como pelos interessados em participar do evento.

As ferramentas utilizadas para o desenvolvimento do projeto foram definidas respeitando os conhecimentos do desenvolvedor e estão descritas nos respectivos itens do capítulo 4.

O desenvolvimento do protótipo está em fase inicial e não foi concluído, não sendo possível, ainda, a sua plena utilização prática pelo Departamento de Pesquisa e Extensão do IFSC, Câmpus Florianópolis.

3 REVISÃO DA LITERATURA

Serão tratados os conceitos e definições de forma fundamentada, relacionados ao tema deste projeto: levantamento de requisitos, modelagem de banco de dados, utilização de linguagem PHP com banco de dados MySQL e arquitetura cliente-servidor, bem como estudos que permitam criar uma visão daquilo que já existe atualmente, em termos de SNTC em outras instituições e que possa ser reaproveitado no presente estudo. No capítulo serão apresentados os conceitos de engenharia de software, engenharia de requisitos, aplicações web e as tecnologias pesquisadas.

[...] o principal produto de uma equipe de desenvolvimento não são documentos bonitos, reuniões sofisticadas, ótimos slogans ou linhas de código-fonte merecedoras do prêmio Pulitzer. O principal produto é um bom software capaz de satisfazer às necessidades de seus usuários e respectivos negócios. Tudo o mais é secundário.

(BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar; 2000, p. 3)

3.1 Engenharia de *software*

De acordo com Sommerville¹, *softwares* são “programas de computador e documentação associada. Produtos de *software* podem ser desenvolvidos para um cliente específico ou para o mercado em geral.” O autor também afirma que a engenharia de *software* tem o propósito de auxiliar no desenvolvimento de *softwares* profissionais, pois utiliza técnicas que apoiam especificação, projeto e evolução de programas, normalmente não utilizadas para desenvolvimento de *softwares* pessoais.

A engenharia de *software* é importante pois cada vez mais todos dependem de programas avançados. Por isso, a necessidade de se produzir *softwares* confiáveis, econômicos e ágeis, utilizando métodos e técnicas da engenharia de *software*, para não haver retrabalho no desenvolvimento.

3.1.1 PROCESSOS DE SOFTWARE

Existem diversos processos de *software*, mas, de acordo com Sommerville², são quatro as atividades fundamentais que todos os processos devem incluir:

- *especificação*: definição das funções e restrições do *software*;
- *projeto e implementação*: o *software* precisa respeitar as especificações;
- *validação*: o *software* precisa ser validado para garantir as demandas do cliente;
- *evolução*: o *software* precisa evoluir para atender as mudanças necessárias dos clientes.

¹ SOMMERVILLE, 2011, p. 3-4

² Ibid., p. 5

3.2 Engenharia de requisitos

Para um *software*, os requisitos são as descrições do que este deve fazer, quais serviços disponibiliza e as restrições de seu funcionamento. O método de encontrar, avaliar e registrar esses serviços e restrições é conhecido como engenharia de requisitos.

Para Sommerville¹, podem ser definidos dois tipos de requisitos para um *software*: requisitos de usuário e requisitos de sistema. O primeiro descreve de uma forma mais informal e geral quais os serviços e restrições o *software* fornecerá aos usuários, enquanto o segundo descreve de forma detalhada e específica as funções, serviços e restrições do *software*. É também chamado de especificação funcional. Essa separação se torna útil para informar diferentes tipos de leitores dos diferentes tipos de detalhamento dos requisitos, que podem utilizá-los de diversas formas.

A seguir, são descritos os conceitos de levantamento e análise de requisitos e UML.

3.2.1 LEVANTAMENTO E ANÁLISE DE REQUISITOS

Nessa etapa, obtêm-se informações com os *stakeholders* (usuários finais e pessoas da organização que serão afetadas pelo *software*) sobre o que ou não o *software* deverá fazer.

As atividades relacionadas ao processo de análise de requisitos são: descoberta, classificação e organização, priorização e negociação, especificação.

Para a análise de requisitos, geralmente são realizadas entrevistas com os *stakeholders*, que, segundo Sommerville², podem ser do tipo fechada, em que são definidas as perguntas da entrevista, e do tipo aberta, onde não são perguntas predefinidas, mas é feita uma análise com os *stakeholders* para uma melhor compreensão de suas necessidades.

É nessa etapa também que são definidos os requisitos funcionais e não funcionais.

¹ SOMMERVILLE, 2011, p. 58

² Ibid., p. 72

Para Guedes (2011, p. 23), após o levantamento dos requisitos vem a fase da análise desses requisitos, para avaliar se todas as necessidades dos usuários foram entendidas corretamente e determinar se algo precisa ser melhorado.

3.2.1.1 *Requisitos funcionais*

Os requisitos funcionais descrevem o que este deverá fazer, ou seja, as funcionalidades do *software*, de forma geral ou específica, de acordo com o tipo de *software* a ser desenvolvido.

É importante que, ao especificar os requisitos, todos os serviços e funções solicitados pelo cliente sejam definidos e essas definições sejam coerentes. Dessa forma, o *software* poderá ser entregue mais rápido, sem necessidade de rever o desenvolvimento por causa de requisitos mal definidos ou inconsistentes (SOMMERVILLE)¹.

3.2.1.2 *Requisitos não funcionais*

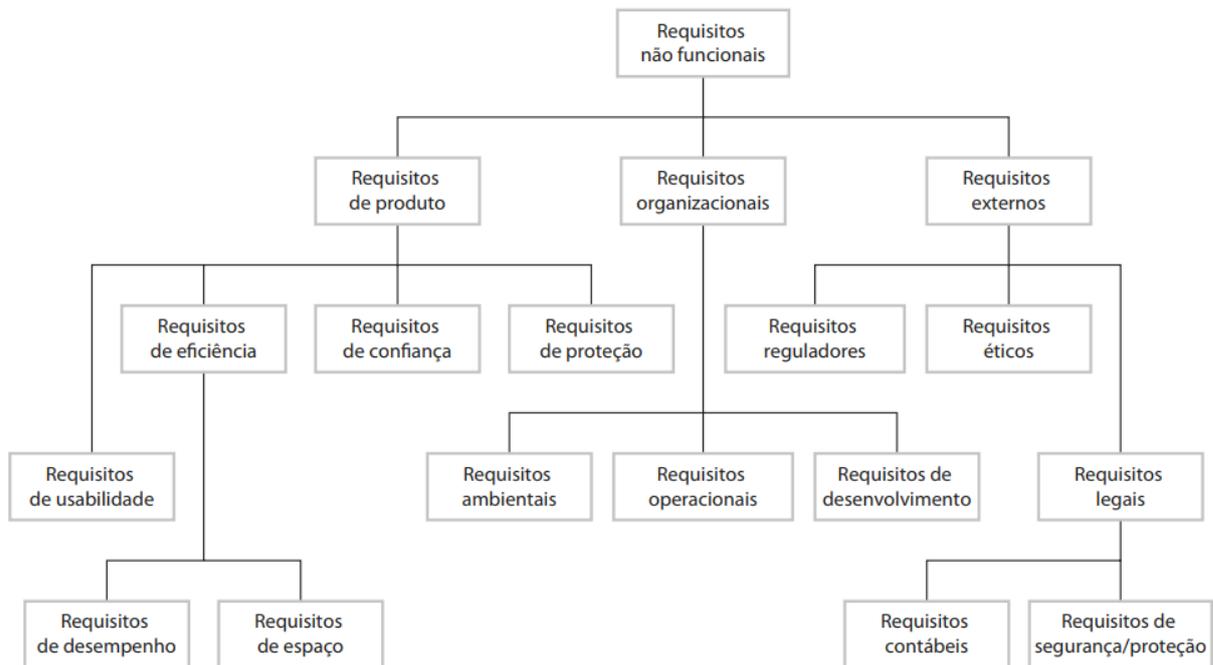
Requisitos não funcionais, nas palavras de Sommerville², “são requisitos que não estão propriamente associados aos serviços e funções específicos disponibilizados pelo *software* aos usuários”. O autor diz ainda que “eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área”. Para Guedes (2011, p. 22), “podem existir diversos tipos de requisitos não-funcionais, como de usabilidade, desempenho, confiabilidade, segurança ou interface.”, como mostra a Figura 1.

Requisitos não funcionais tendem a ser mais decisivos que requisitos funcionais, pois os usuários podem encontrar uma forma de evitar alguma funcionalidade do *software* que não atenda às suas necessidades. Já com um requisito não funcional, pode significar a inutilização do *software*.

¹ Ibid., p. 73-74

² SOMMERVILLE, 2011, p. 60

Figura 1 - Tipos de requisitos não funcionais



Fonte: Sommerville (2011, p. 61)

3.2.2 UML

A UML (*Unified Modeling Language*) é uma linguagem designada a visualizar, especificar, construir e documentar, permitindo o entendimento de um *software*.

Existem diversos projetos de *software* disponíveis no mercado, alguns bem-sucedidos e outros fracassados, e quando são bem-sucedidos, a semelhança se dá por diversos aspectos, e a utilização da modelagem é um deles (BOOCH; RUMBAUGH; JACOBSON; 2000, p. 5-6). Os autores ainda afirmam que “um modelo é uma simplificação da realidade” e que “construímos modelos para compreender melhor o sistema que estamos desenvolvendo.”

Para Guedes (2011, p. 21), “um modelo de *software* captura uma visão de um sistema físico, é uma abstração do sistema com um certo propósito, como descrever aspectos estruturais ou comportamentais do *software*.” Ela afirma ainda que “esse propósito determina o que deve ser incluído no modelo e o que é considerado irrelevante.”

Para Sampaio¹, o modelo contido na UML é composto de itens, relacionamentos e diagramas.

3.2.2.1 Itens

De acordo com Sampaio², estão divididos em:

- *estruturais*: classes, casos de uso, componentes;
- *comportamentais*: interações, máquinas de estado;
- *grupos de elementos*: pacotes, *frameworks* e subsistemas;
- *anotacionais*: notas.

3.2.2.2 Relacionamentos

Segundo Sampaio³, os relacionamentos podem ser entre classes, casos de uso, entre outros e podem ser classificados em dependência, associação, generalização e realização.

3.2.2.3 Diagramas

Para Booch, Rumbaugh e Jacobson (2000, p. 25-27), “um diagrama é a apresentação gráfica de um conjunto de elementos [...] desenhados para permitir a visualização de um sistema sob diferentes perspectivas” e que podem ser classificados nestes principais tipos, também demonstrados na Figura 2:

- *diagrama de classes*: exibe um conjunto de classes, interfaces, colaborações e seus relacionamentos;
- *diagrama de objetos*: exibe um conjunto de objetos e seus relacionamentos;
- *diagrama de casos de uso*: exibe um conjunto de caso de uso e atores e seus relacionamentos;
- *diagrama de interação*: exibe um conjunto de objetos e seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles;

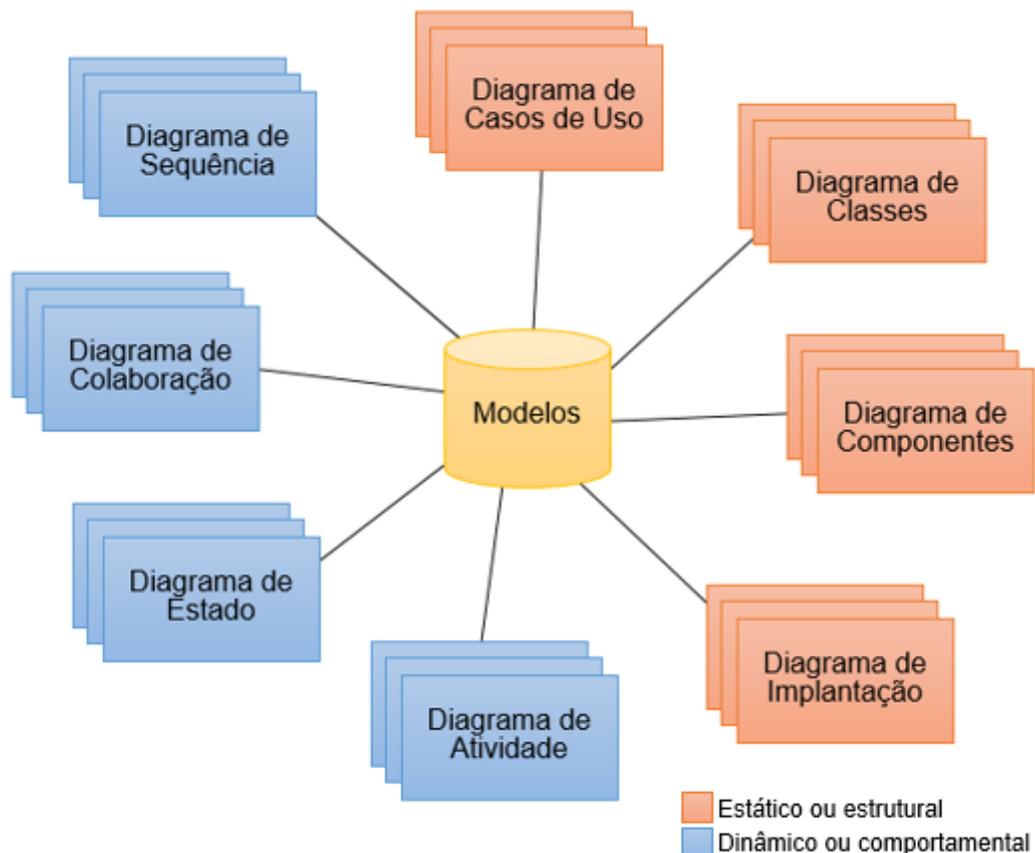
¹ SAMPAIO, 2007

² Id.

³ Id.

- ♦ *diagrama de seqüência*: diagrama de interação com destaque para a ordenação temporal das mensagens trocadas;
- ♦ *diagrama de colaborações*: diagrama de interação com destaque para a organização estrutural dos objetos que enviam e recebem mensagens.
- *diagrama de gráfico de estados*: exibe uma máquina de estados, composta de estados, transições, eventos e atividades;
- *diagrama de atividade*: exibe um diagrama de gráfico de estados, que apresenta o fluxo de uma atividade para outra
- *diagrama de componente*: exibe as organizações e as dependências;
- *diagrama de implantação*: exibe a configuração dos nós de processamento em tempo de execução e seus componentes;

Figura 2 - Diagramas da UML



Fonte: Sampaio (2007)

3.2.2.3.1 Diagrama de casos de uso

De acordo com Guedes¹, o diagrama de caso de uso é utilizado no início da modelagem de um *software*, nas etapas de levantamento e análise de requisitos, apesar de ser consultado e possivelmente modificado durante todo o processo de engenharia do software e sirva como base para outros diagramas.

Guedes² ainda afirma que o diagrama de casos de uso auxilia na identificação e compreensão dos requisitos, possibilitando especificar, visualizar, e documentar as características, funções e serviços do *software* que o usuário deseja. O diagrama também tenta identificar quais os tipos de usuário que irão utilizar o *software*, quais papéis esses usuários assumirão e quais funções um usuário específico poderá requisitar.

O diagrama apresenta dois tipos de itens: atores e casos de uso, que se relacionam entre si, conforme mostra o exemplo na Figura 3.

Os atores representam os papéis que os usuários exercerão quando utilizarem os serviços e funções do *software*. O ator também poderá ser um *hardware* especial ou outro *software*, ou seja, qualquer elemento externo que se relacione com o *software*.

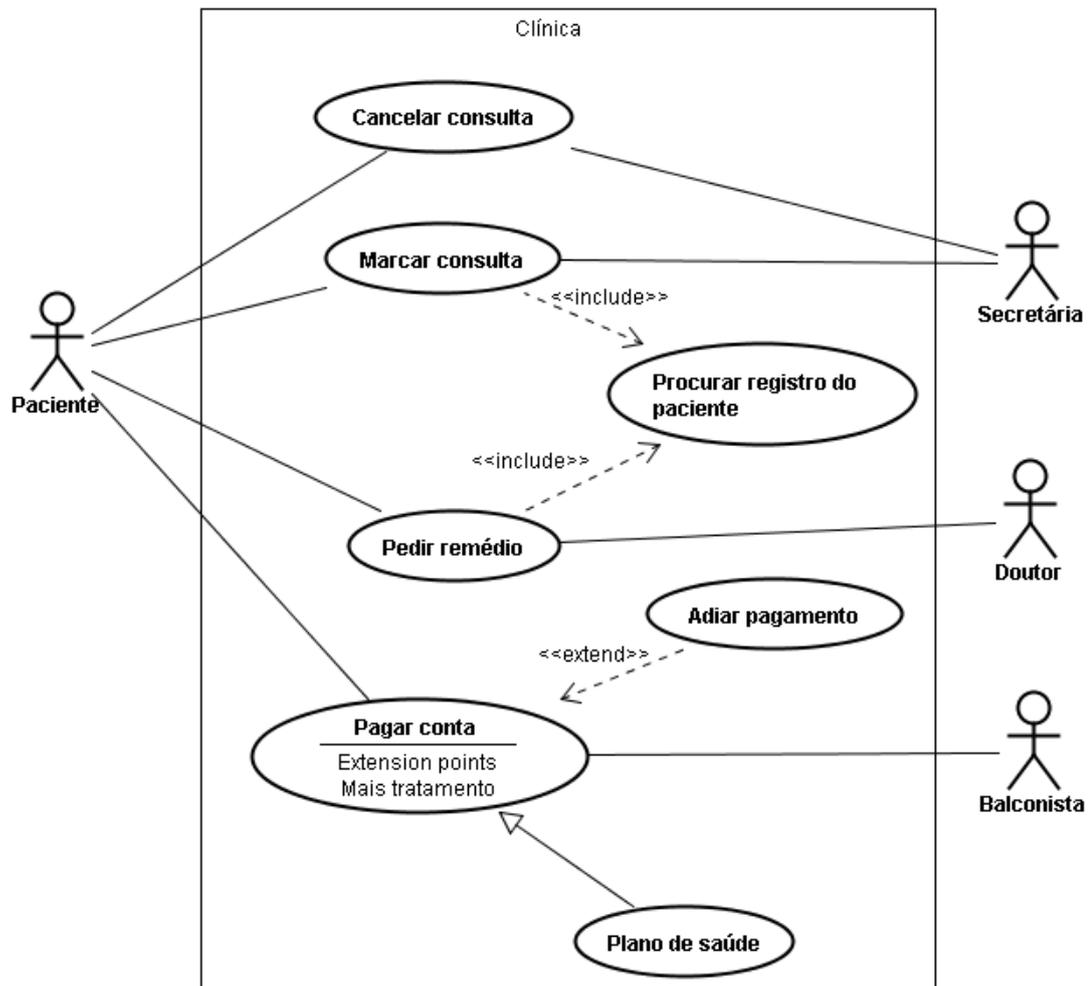
Para Booch, Rumbaugh e Jacobson (2000, p. 217), os casos de usos especificam o comportamento desejado, mas não como este será executado.

Nas palavras de Sampaio (2007), os relacionamentos “ajudam a descrever casos de uso”, e podem ser entre um ator e um caso de uso, entre atores ou entre casos de uso.

¹ GUEDES, 2011, p. 52

² Id.

Figura 3 - Exemplo de diagrama de casos de uso



Fonte: Sampaio (2007)

3.2.2.3.2 Diagrama de classes

De acordo com Booch, Rumbaugh e Jacobson (2000, p. 114), diagramas de classes são uma representação gráfica da visão estética do projeto de um *software*.

Para Guedes¹, esse diagrama é o mais importante e o mais utilizado, pois permite visualizar as classes com seus atributos e métodos, demonstrando como as classes se relacionam, complementam e transmitem informações entre elas.

Ainda segundo Guedes², classes podem ter atributos (que armazenam os dados dos objetos da classe) e métodos (funções que uma instância da classe pode

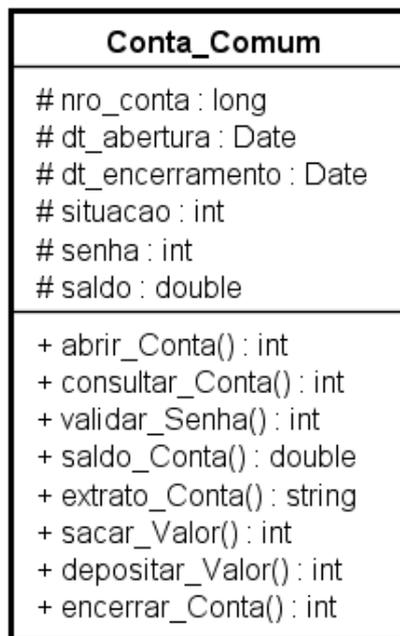
¹ GUEDES, 2011, p. 101

² Ibid., p. 101-102

executar), e o diagrama de classes não define as etapas que os métodos deverão percorrer quando forem chamados.

A classe pode ser representada por um retângulo dividido em até 3 partes, sendo que a primeira contém o nome da classe, a segunda os atributos com seus tipos de dados e a terceira com os métodos, como demonstrado no exemplo da Figura 4.

Figura 4 - Exemplo de classe



Fonte: Guedes (2011, p. 102)

3.3 Aplicação web

Uma das dúvidas que surgem até mesmo entre programadores é o conceito de *site* e aplicação *web*.

De acordo com Messenlehner e Coleman (2014, p. 24), “um *site* é um conjunto de uma ou mais páginas *web* contendo informações e que é acessado por meio de um navegador *web*”, como, por exemplo, *sites* de empresas, que exibem a história da empresa, formas de contato e descrição de seus produtos.

Uma aplicação *web* pode ser definida com um sistema que pode realizar muitas funcionalidades, como cadastros, relatórios e automatização de processos.

Segundo Beighley e Morrison (2010, p. 105), “uma aplicação *web* é um *site* dinâmico elaborado para atender a um determinado propósito dos seus usuários.”

Para Messenlehner e Coleman (2014, p. 25-27), aplicações *web* possuem algumas características semelhantes:

- *elementos interativos: links*, rolagens, formulários, barras de ferramenta, elementos para arrastar e soltar, editores de texto rico (*rich text*) e *sliders* (controles deslizantes);
- *tarefas no lugar de conteúdo*: auxiliam o usuário a realizar tarefas específicas, como exemplo os usuários do Google Maps recebem orientações sobre direções a seguir;
- *logins*: permitem salvar informações sobre seus usuários;
- *capacidade de se adequar aos dispositivos*: podem se adequar de acordo com o dispositivo que a acessar, se ajustando a diferentes tamanhos de tela, resoluções e capacidades;
- *trabalhar offline*: permitem que se trabalhe sem a interatividade com a internet;
- *mashups*: podem se associar a uma ou mais aplicações *web*, como exemplo, pode se extrair a localização do Twitter e postar no Google Maps.

3.4 Tecnologias

A seguir são descritas as tecnologias pesquisadas para o presente projeto.

3.4.1 LINGUAGENS DE PROGRAMAÇÃO

Segundo Sebesta (2011, p. 23-24), linguagens de programação com diversos objetivos têm sido desenvolvidas para atender as mais diferentes áreas, desde usinas nucleares a jogos de celular. O autor ainda afirma que as primeiras linguagens de programação de alto nível foram projetadas para atender as necessidades de aplicações científicas, que necessitam principalmente de eficiência.

A *World Wide Web* é mantida por uma eclética coleção de linguagens, que vão desde linguagens de marcação, como XHTML, que não é de programação, até linguagens de programação de propósito geral, como Java. Dada a necessidade universal por conteúdo dinâmico na Web, alguma capacidade de computação geralmente é incluída na tecnologia de apresentação de conteúdo. Essa funcionalidade pode ser fornecida por código de programação embarcado em um documento XHTML. Tal código

é normalmente escrito com uma linguagem de scripting, como JavaScript ou PHP. Existem também algumas linguagens similares às de marcação que têm sido estendidas para incluir construções que controlam o processamento de documentos [...].
(SEBESTA, 2011, p. 25)

3.4.2 HTML

Segundo o site W3C Brasil, em 1989 Tim Berners-Lee inventou a *World Wide Web* e em 1990 escreveu a primeira versão do HTML (*HyperText Markup Language* – Liguagem de Marcação de Hipertexto, em português), “a linguagem de formatação de documentos com a capacidade de *links* de hipertexto e que tornou-se o formato básico para publicação na *web*.” Ele também é fundador da organização *World Wide Web Consortium*, mais conhecida como W3C, onde organizações filiadas, equipe em tempo integral e o público interessando trabalham juntos para desenvolver padrões para a *web*.

De acordo com Silva (2014), “podemos resumir hipertexto como todo o conteúdo inserido em um documento para a *web* e que tem como principal característica a possibilidade de se interligar a outros documentos da *web*”.

Conforme o site W3Schools, o HTML descreve as estruturas das páginas da *web* usando marcações, representadas, normalmente, por um par de *tags*, denominadas *tag* de abertura e de fechamento, que são processadas por *browsers* (navegadores) para exibir o conteúdo da página.

A versão atual do HTML é a 5 e, nas palavras de Crowther et al. (2014, p. 25), “é a primeira versão a adotar a *web* como uma plataforma para o desenvolvimento de aplicativos *web*.”

É possível exibir o código HTML de qualquer navegador, clicando com o botão direito sobre a página *web* e em “Exibir código fonte da página”, no caso do Google Chrome.

3.4.3 CSS

Segundo o site W3C, CSS (*Cascading Style Sheet* – Folhas de Estilo em Cascata, no português) é um método simples para adicionar estilo às páginas da *web*, como, por exemplo, fontes, cores, espaçamento, entre outros.

De acordo com Silva (2014, p. 215), “define-se como regra de estilo como a menor porção de código capaz de causar efeito de estilização no conteúdo de um elemento da marcação.” Uma regra de estilo possui 3 partes:

- *seletor*: é capaz de definir com precisão onde na marcação será aplicada a regra de estilo;
- *propriedade*: define o que será estilizado;
- *valor*: define o quanto ou como será estilizado, finalizada com um sinal de ponto-e-vírgula.

A propriedade e o valor devem ser colocados entre sinais de chaves e um sinal de dois-pontos separa a propriedade de seu valor.

3.4.4 JAVASCRIPT

Segundo Flanagan¹, a linguagem foi criada na empresa Netscape, que foi usada para descrever a implementação da linguagem pelo navegador Netscape (hoje Mozilla Firefox).

De acordo com Flanagan², “JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais.”. É usada pela maioria dos sites modernos e interpretada por todos os navegadores mais atualizados.

JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas.
(FLANAGAN)³

3.4.5 PHP

De acordo com o site PHP, o PHP (PHP: *Hypertext Preprocessor*) “é uma linguagem de programação de ampla utilização, interpretada, [...] para desenvolvimento para a *web* e pode ser mesclada dentro do código HTML”.

¹ FLANAGAN, 2013, p. 2

² Ibid., p. 1

³ Id.

O PHP é delimitado pelas *tags* de início `<?php` e fechamento `?>`, permitindo que se entre ou saia do modo PHP.

O código PHP é executado no servidor, que gera o HTML e envia o código ao navegador. Também pode ser utilizado na maioria dos sistemas operacionais e servidores *web*. É possível se utilizar programação estruturada, programação orientada a objetos ou uma mistura das duas.

Segundo o site PHP, “uma das características mais fortes e significativas do PHP é seu suporte a uma ampla variedade de bancos de dados”.

A maioria dos programas PHP dedica uma grande parcela de tempo à manipulação de texto porque se atém à geração de HTML e ao trabalho com as informações de um banco de dados. O código HTML é texto formatado de maneira especial, e as informações do banco de dados, como um nome de usuário, a descrição de um produto, ou um endereço, também são texto. O fato de ser possível manipular texto facilmente significa que você pode construir páginas web dinâmicas sem esforço.
(SKLAR, 2016, p. 40)

Para funcionamento do PHP é necessário instalar um servidor *web*, como o Apache, instalar o PHP e, possivelmente, um banco de dados, como o MySQL.

3.4.6 BANCOS DE DADOS

Nas palavras de Date¹, “um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros [...], ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados.” Em outras palavras, “é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar” (DATE)².

[Banco de dados é] um conjunto de dados inter-relacionados, organizados de forma a permitir que sistemas de aplicação armazenem novos dados, encontrem dados armazenados, alterem seu conteúdo e excluam dados indesejáveis por meio de métodos precisos de manipulação e localização.
(FEITOSA, 2013, p. 15)

Segundo o site DevMedia, um sistema de gerenciamento de banco de dados (SGBD) “é um *software* que possui recursos capazes de manipular as

¹ DATE, 2003, p. 3

² Ibid., p. 6

informações do banco de dados e interagir com o usuário”, que possui quatro componentes básicos: dados, *hardware*, *software* e usuários.

Para Ramakrishnan e Gehrke¹, dentre as vantagens de se usar um SGBD estão:

- *independência de dados*: o SGBD abstrai detalhes de representação e armazenamento de dados dos sistemas;
- *acesso eficiente aos dados*: o SGBD utiliza técnicas sofisticadas para armazenar e recuperar dados de forma eficiente;
- *integridade e segurança dos dados*: o SGBD pode forçar restrições de integridade entre os dados informados, além de controles de acesso dos dados aos usuários;
- *administração de dados*: o SGBD organiza a representação dos dados para diminuir redundâncias e garantir uma eficiência recuperação;
- *acesso concorrente e recuperação de falhas*: o SGBD permite que os dados sejam acessados por diversos usuários mesmo parecendo que seja somente um usuário por vez, além de proteger contra os efeitos das falhas no sistema;
- *tempo reduzido de desenvolvimento de aplicativos*: o SGBD permite que diversas tarefas importantes sejam tratadas por ele, não precisando depurar e testar no aplicativo.

O modelo relacional é muito simples e elegante: um banco de dados é uma coleção de uma ou mais relações, em que cada relação é uma tabela com linhas e colunas. Essa representação tabular simples permite que até usuários iniciantes entendam o conteúdo de um banco de dados e possibilita o uso de linguagens de alto nível simples para consultar os dados. As principais vantagens do modelo relacional em relação aos modelos de dados mais antigos são sua representação de dados simples e a facilidade com que mesmo consultas complexas podem ser expressas. (RAMAKRISHNAN; GEHRKE)²

Segundo Date (2003, p. 93), o modelo relacional possui três aspectos principais de dados: estrutura, manipulação e integridade de dados.

¹ RAMAKRISHNAN; GEHRKE, 2011, p. 7

² Ibid., p. 49

De acordo com Ramakrishnan e Gehrke (2011, p. 50), o modelo relacional representa dados através da relação, sendo que uma relação consiste em um esquema e uma instância de relação, onde a instância é um conjunto de tuplas ou registros, podendo ser chamada de tabela, onde cada tupla é uma linha e todas as linhas tem o mesmo número de campos e o esquema especifica o nome da relação, o nome de cada campo ou coluna e o domínio de cada campo (que possui um nome de domínio e um conjunto de valores associados), podendo ser definido como o tipo do campo, restringindo os valores que o campo pode receber.

É possível aos usuários criar, modificar e consultar dados através da linguagem SQL (*Structured Query Language* – Linguagem de Consulta Estruturada, em português).

3.4.7 ARQUITETURA CLIENTE-SERVIDOR

De acordo com Sommerville¹, “um sistema que segue o padrão cliente-servidor é organizado como um conjunto de serviços e servidores associados e clientes que acessam e usam os serviços”.

Os clientes podem ter de saber os nomes dos servidores disponíveis e os serviços que eles fornecem. No entanto, os servidores não precisam conhecer a identidade dos clientes ou quantos clientes estão acessando seus serviços. Os clientes acessam os serviços fornecidos por um servidor por meio de chamadas de procedimento remoto usando um protocolo de solicitação-resposta, tal como o protocolo HTTP usado na Internet. Essencialmente, um cliente faz uma solicitação a um servidor e espera até receber uma resposta.
(SOMMERVILLE)²

Ainda segundo Sommerville³, os principais componentes do modelo cliente-servidor são:

- a) um conjunto de servidores oferecendo serviços a outros componentes. Esses servidores podem ser servidores de impressão (oferecem serviços de impressão), servidores de arquivos (oferecem serviços de gerenciamento de arquivos) e servidores de compilação (oferece serviços de compilação de linguagens de programação);

¹ SOMMERVILLE, 2011, p. 113

² Id.

³ Id.

- b) um conjunto de clientes para chamar os serviços oferecidos pelos servidores, onde geralmente existe várias instâncias de um programa cliente executante em diversos computadores, simultaneamente;
- c) uma rede que permite aos clientes acessar os serviços.

Tabela 1 - O padrão cliente-servidor

Nome	Cliente-servidor
Descrição	Em uma arquitetura cliente-servidor, a funcionalidade do sistema está organizada em serviços — cada serviço é prestado por um servidor. Os clientes são os usuários desses serviços e acessam os servidores para fazer uso deles.
Quando é usado	É usado quando os dados em um banco de dados compartilhado precisam ser acessados a partir de uma série de locais. Como os servidores podem ser replicados, também pode ser usado quando a carga em um sistema é variável.
Vantagens	A principal vantagem desse modelo é que os servidores podem ser distribuídos através de uma rede. A funcionalidade geral (por exemplo, um serviço de impressão) pode estar disponível para todos os clientes e não precisa ser implementada por todos os serviços.
Desvantagens	Cada serviço é um ponto único de falha suscetível a ataques de negação de serviço ou de falha do servidor. O desempenho, bem como o sistema, pode ser imprevisível, pois depende da rede. Pode haver problemas de gerenciamento se os servidores forem propriedade de diferentes organizações.

Fonte: Sommerville¹

Para Sommerville², sistemas que são acessados pela internet geralmente são organizados como cliente-servidor. Dessa forma, o usuário interage com um programa executado em seu computador local, através de um navegador, por exemplo, e este interage com outro programa executado em computador remoto (como um servidor *web*, por exemplo). O computador remoto oferece serviços, como o acesso às páginas da *web*, disponíveis para clientes externos.

Os sistemas cliente-servidor dependem de haver uma separação clara entre a apresentação de informações e as computações que criam e processam essas informações. Consequentemente, você deve projetar a arquitetura dos sistemas cliente-servidor distribuídos para que eles sejam estruturados em várias camadas lógicas, com interfaces claras entre essas camadas. Isso permite que cada camada seja distribuída para um computador diferente.

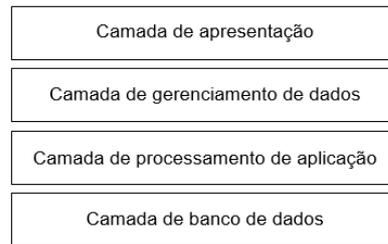
(SOMMERVILLE)³

¹ SOMMERVILLE, 2011, p. 113

² Ibid., p. 339

³ Ibid., p. 340

Figura 5 - Modelo de arquitetura em camadas para aplicações cliente-servidor



Fonte: Sommerville¹

A Figura 5, de acordo com Sommerville², mostra uma aplicação em quatro camadas:

- a) uma camada de apresentação, referente à apresentação de informações para o usuário;
- b) uma camada de gerenciamento de dados que são passados de e para o cliente;
- c) uma camada de processamento de aplicação, que fornece a funcionalidade necessária aos usuários finais;
- d) uma camada de banco de dados que armazena os dados e gerencia as transações.

As tecnologias Web [...] suportam o gerenciamento eficiente de apresentação de página Web e a computação local por meio de scripts. Isso significa que um browser pode ser configurado e usado como um cliente, com processamento local significativo. O software de aplicação pode ser pensado como um serviço remoto, que pode ser acessado de qualquer dispositivo que possa executar um browser-padrão. (SOMMERVILLE)³

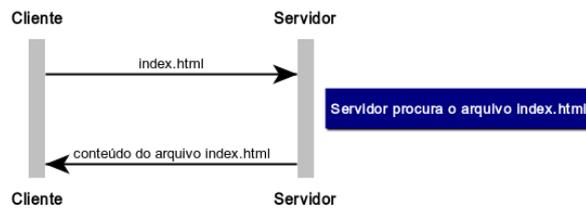
Segundo Gomes (2017), a comunicação na internet se dá quando o cliente gera uma requisição para o servidor, que a responde, sendo que o cliente solicita um recurso, como exemplo uma imagem ou arquivo .html, onde o cliente geralmente é representado por um navegador. Entre os navegadores mais conhecidos estão Google Chrome, Firefox, Internet Explorer e Safari. Da forma como o cliente é representado pelo navegador, o servidor é representado pelo *web server* (servidor *web*), onde Apache HTTP, IIS e Nginx estão entre os mais conhecidos.

¹ SOMMERVILLE, 2011, p. 340

² Id.

³ Ibid., p. 349

Figura 6 - Comunicação cliente-servidor na internet



Fonte: Gomes¹

Na arquitetura Cliente-Servidor [de aplicações web], o processamento das requisições é realizado por dois “lados” o Servidor (Server side) e Cliente (Client side). Assim, quando o usuário (cliente), através de um navegador (navegador) faz acesso a uma página específica, o servidor faz o processamento da requisição, processa-a e envia para o cliente arquivos em diferentes formatos, por exemplo, HTML ou CSS. (GOMES)²

Gomes³ ainda afirma que o código disponibilizado pelo lado do servidor (*server side*):

- pode ser qualquer código que pode ser executado no servidor e responder às requisições HTTP (*HyperText Transfer Protocol – Protocolo de Transferência de Hipertexto, em português*). Como exemplo estão códigos em Java, C# e PHP;
- permite a persistência de dados;
- não pode ser visualizado pelo usuário;
- responde às requisições HTTP através de uma URL (*Uniform Resource Locator - Localizador Uniforme de Recursos, em português*) particular;
- cria páginas que o usuário pode visualizar em um navegador.

Um Servidor Web tem como função receber requisições e, a partir delas, oferecer respostas para o cliente. Estas requisições podem ser realizadas através de um navegador de internet (browser) pelo usuário e, geralmente, recebem como resposta uma página HTML que é interpretada pelo browser e apresentada ao usuário. (GOMES)⁴

Gomes⁵ continua, explicando que o código disponibilizado no lado cliente (*client side*):

¹ GOMES, 2017

² Id.

³ Id.

⁴ Id.

⁵ Id.

- permite o uso de linguagens como HTML, CSS e JavaScript;
- é interpretado pelo navegador do usuário;
- reage às entradas do usuário;
- permite a leitura de arquivos do servidor através de requisições HTTP;
- torna possível a exibição de conteúdos para os usuários.

Os códigos do “lado” do cliente são analisados pelo próprio navegador do cliente e são, geralmente, gerados a partir de solicitações destes clientes ao servidor. Se por acaso o servidor hospedar um determinado documento HTML (processado do lado do cliente), o servidor então tira uma cópia deste arquivo HTML e o envia para o cliente.
(GOMES, 2017)

3.4.8 FRAMEWORKS

De acordo com Sklar (2016, p. 353), “um *framework* de aplicativo é um conjunto de funções, classes e convenções que facilita a execução de tarefas comuns [...]. Eles o ajudarão a, partindo do zero, chegar a um aplicativo *web* funcional.” Os *frameworks* fornecem maneiras-padrão de executar roteamento, mapeamento objeto-relacional e gerenciamento de usuários.

O autor ainda afirma que a utilização de um *framework* economiza tempo comparado à implantar as funcionalidades por conta própria, e pode tornar mais fácil o entendimento de novos desenvolvedores já familiarizados com o *framework*, porém há a desvantagem de que demanda tempo para aprender a utilizar o *framework* e se adaptar às suas convenções de como fazer as coisas.

4 DESENVOLVIMENTO

4.1 Análise do sistema

Conforme já citado, a proposta é a criação de um protótipo de uma aplicação para a *web*, destinado à SNCT do IFSC Câmpus Florianópolis, que poderá ser acessado pelos usuários, interno e externos, através do navegador.

O protótipo da aplicação possui integração com um banco de dados, ambiente para usuários externos, possibilitando o cadastro e acompanhamento de suas inscrições e área administrativa, utilizada por usuários internos, onde é possível realizar todos os cadastros internos, bem como fazer o gerenciamento das inscrições e atividades.

4.2 Levantamento de requisitos

O levantamento e a análise dos requisitos se deu mediante entrevistas com o Sr. Caio C. Martins e o Sr. Clóvis Antônio Petry, membros do Departamento de Pesquisa e Extensão do IFSC, Câmpus Florianópolis, pois serão usuários da aplicação e vivenciaram experiências em eventos anteriores, podendo definir, da melhor forma, as necessidades que a aplicação poderá suprir.

A primeira entrevista foi realizada em 15 de maio de 2017, onde foi possível saber mais detalhes de como a SNCT é realizada sem o auxílio de uma aplicação e quais os problemas foram encontrados através da experiência com os eventos anteriores, conforme já descrito no item 1.2 deste projeto.

A segunda entrevista realizou-se em 19 de junho de 2017, onde foi possível sanar algumas dúvidas sobre como a aplicação deveria proceder em situações específicas.

4.2.1 REQUISITOS FUNCIONAIS

Como requisitos funcionais, estabeleceu-se que é necessário que a aplicação permita a visualização de uma página *web*, que propicie o acesso às inscrições no evento para os participantes. Estes participantes poderão se cadastrar nas atividades de interesse, não sendo possível que o participante se inscreva em mais de uma atividade no mesmo horário. Essas atividades serão previamente

definidas pelos usuários internos e deverão ser exibidas na página *web* como programação do evento.

Os usuários internos terão um acesso exclusivo à área de administração da aplicação, onde poderão realizar os cadastros internos, controlar as inscrições e gerar listas dos participantes de cada atividade, gerenciar a carga horária dos participantes nas atividades, bem como gerar os certificados correspondentes.

4.2.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais da aplicação implicam que a mesma deverá estar acessível em qualquer navegador, com confiabilidade, comunicando-se a todo momento com o banco de dados.

4.3 Estudo comparativo com aplicações já existentes

Foi realizado um levantamento de aplicações que estão disponíveis na internet, para verificar se alguma delas atendia os requisitos informados. Dentre as aplicações comparadas, as que foram consideradas mais aptas, com maior número de requisitos atendidos, são: vpEventos e Even3.

A aplicação vpEventos¹ disponibiliza o cadastro de inscrições gratuitas e pagas, sendo que, para utilizar as inscrições pagas, é cobrada uma taxa administrativa, com a possibilidade de repassar o valor aos inscritos. Esta aplicação tem uma área administrativa acessada através de usuário e senha, onde é possível o controle de horas das atividades, emissão de certificados, controle para que o inscrito não se cadastre em atividades que ocorrem simultaneamente e a geração de lista de participantes das atividades. É possível criar diversos eventos, sendo que cada um gera um *link* de acesso diferente ao *site* da aplicação, porém não é possível personalizar o endereço, ficando algo como <http://www.meuevento.vpeventos.com>. Pela simulação realizada, foi possível identificar que o acesso do inscrito é bem confuso, possui muita propaganda do fornecedor e em alguns momentos a aplicação não funciona corretamente. Dessa forma, a utilização desta aplicação foi descartada.

¹ <http://www.vpeventos.com/>

A aplicação Even3¹ também disponibiliza inscrições gratuitas e pagas, não estando descrito se é cobrada taxa administrativa para as inscrições pagas. Esta aplicação tem uma área administrativa acessada através de usuário e senha, onde possibilita o controle de horas das atividades, emissão de certificados, controle para o inscrito não se cadastrar em atividades simultâneas e gera a lista de participantes. Porém é uma lista geral do evento, e não por atividade. Não permite gerar mais de um evento com o mesmo acesso e também não permite personalizar o endereço do *site*, ficando algo como <http://www.even3.com.br/meuevento>. De acordo com a simulação realizada, o gerenciamento é bem simples e fácil, bem como realizar as inscrições. Contudo, a aplicação não atende a todos os requisitos. Dessa forma, a utilização desta aplicação foi descartada.

As demais aplicações testadas (Doity², Eventbrite³ e Sympla⁴) não permitiam o cadastro de atividades no evento. Dessa forma, a utilização destas aplicações foi desconsiderada.

4.4 Tecnologias utilizadas

O protótipo da aplicação para a SNCT foi desenvolvido no *front-end*⁵ utilizando-se HTML, CSS e JavaScript, sem o uso de *frameworks*. Para o *back-end*⁶, foi utilizada a linguagem PHP, também sem o uso de *frameworks*.

Como SGBD foi utilizado o MySQL, pois, segundo informações do *site* MySQL, ele tem “o desempenho e a escalabilidade necessários para atender às necessidades dos aplicativos mais exigentes”. O *site* também informa que empresas como Facebook, Google e Youtube utilizam MySQL, bem como que “a facilidade de uso do MySQL tem sido uma meta desde sua concepção e um dos principais fatores de sua adoção e popularidade.”

¹ <https://www.even3.com.br>

² <https://doity.com.br>

³ <https://www.eventbrite.com.br>

⁴ <https://www.simpla.com.br>

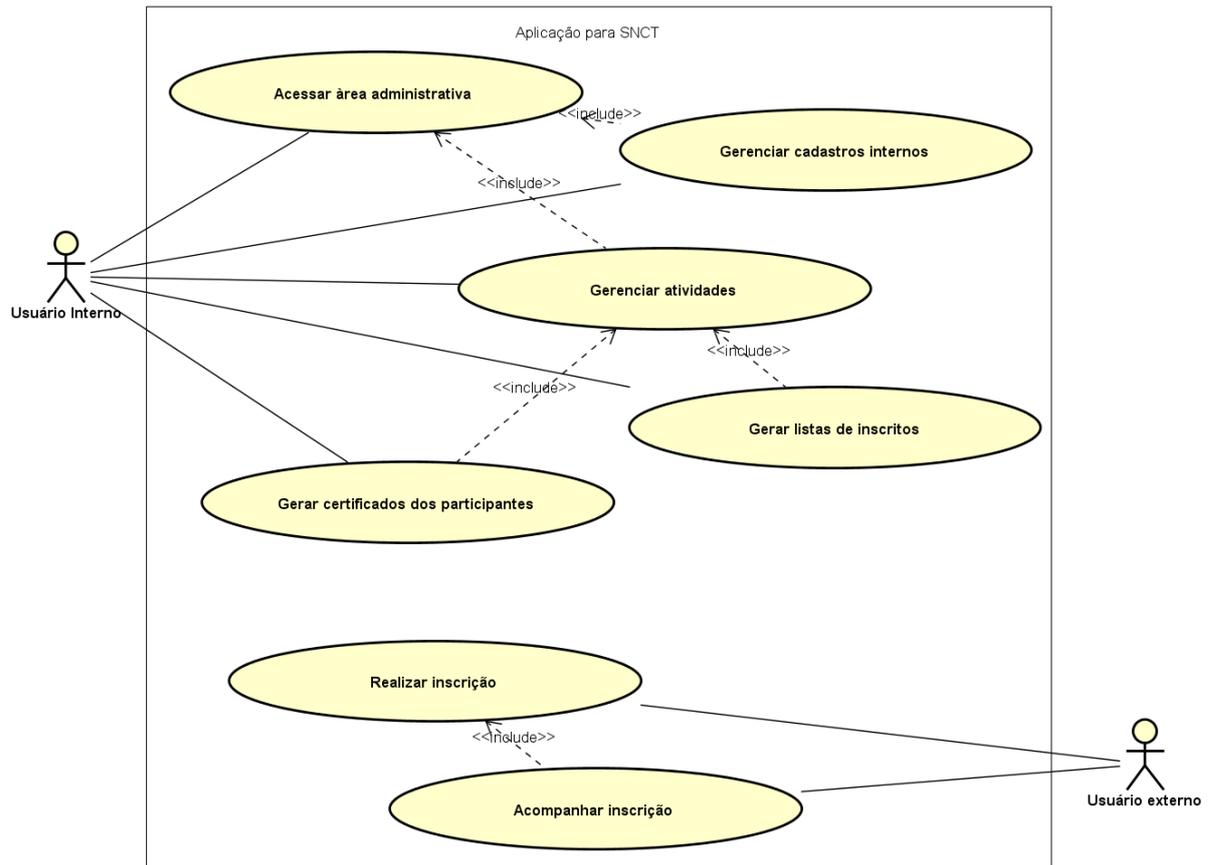
⁵ Parte da aplicação que interage diretamente com o usuário; interface.

⁶ Parte da aplicação que implementa a regra de negócio, a qual o usuário não visualiza.

4.4.1 DIAGRAMA DE CASOS DE USO

Para a elaboração do diagrama de casos de uso, foi utilizada a ferramenta Astah Professional.

Figura 7 - Diagrama de casos de uso da aplicação



Fonte: Elaborado pelo autor

Tabela 2 - Documentação do caso de uso "Acessar área administrativa"

Nome do caso de uso	Acessar área administrativa
Descrição	Esse caso de uso permite ao usuário interno acessar a área administrativa da aplicação
Ator principal	Usuário interno
Pré-condições	O usuário precisa estar conectado à internet e ter um usuário cadastrado

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário digitará o endereço do site na barra de endereços do navegador, acessará o menu "Área restrita", informará seus dados de acesso e clicará em "Entrar".	
	2. A aplicação consultará se os dados informados são válidos e exibirá a página acessada.

FLUXO ALTERNATIVO

Ações do ator	Ações do sistema
	1. Caso os dados informados sejam inválidos, a aplicação exibirá mensagem e usuário deverá preencher com dados válidos.

Fonte: Elaborado pelo autor

Tabela 3 - Documentação do caso de uso "Gerenciar cadastros internos"

Nome do caso de uso	<i>Gerenciar cadastros internos</i>
Caso de uso geral	Acessar área administrativa
Descrição	Esse caso de uso permite ao usuário interno gerenciar os cadastros internos da aplicação
Ator principal	Usuário interno
Pré-condições	O usuário precisa estar conectado à internet e executar o caso de uso "Acessar área administrativa"

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário selecionará o cadastro desejado.	
	2. A aplicação exibirá a tela do cadastro selecionado.
3. O usuário poderá utilizar as opções de novo registro e alteração ou exclusão de um registro existente. Ao cadastrar um novo ou alterar um existente, precisará utilizar a opção "Cadastrar".	
	4. A aplicação registrará os dados informados no banco de dados e exibirá a tela para novas ações.

Fonte: Elaborado pelo autor

Tabela 4 - Documentação do caso de uso "Gerenciar atividades"

Nome do caso de uso	<i>Gerenciar atividades</i>
Caso de uso geral	Acessar área administrativa
Descrição	Esse caso de uso permite ao usuário interno gerenciar as atividades cadastradas na aplicação
Ator principal	Usuário interno
Pré-condições	O usuário precisa estar conectado à internet e executar o caso de uso "Acessar área administrativa"

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário selecionará a atividade desejada.	
	2. A aplicação exibirá a tela com os dados da atividade e ações.
3. O usuário poderá utilizar a ação desejada, como, por exemplo, "Confirmar presença".	

Fonte: Elaborado pelo autor

Tabela 5 - Documentação do caso de uso "Gerar lista de inscritos"

Nome do caso de uso	<i>Gerar lista de inscritos</i>
Caso de uso geral	Gerenciar atividades
Descrição	Esse caso de uso permite ao usuário interno gerar listas dos participantes que se inscreveram nas atividades
Ator principal	Usuário interno
Pré-condições	O usuário precisa estar conectado à internet e executar o caso de uso "Gerenciar atividades"

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário clicará na opção "Gerar lista de inscritos".	
	2. A aplicação consultará o banco de dados e exibirá na tela a listagem dos inscritos na atividade selecionada.
3. O usuário poderá utilizar a ação desejada, como, por exemplo, "Imprimir".	

Fonte: Elaborado pelo autor

Tabela 6 - Documentação do caso de uso "Gerar certificados dos participantes"

Nome do caso de uso	<i>Gerar certificados dos participantes</i>
Caso de uso geral	Gerenciar atividades
Descrição	Esse caso de uso permite ao usuário interno gerar certificados dos que participaram das atividades
Ator principal	Usuário interno
Pré-condições	O usuário precisa estar conectado à internet e executar o caso de uso "Gerenciar atividades"

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário clicará na opção "Gerar certificados".	2. A aplicação consultará o banco de dados para verificar os participantes que têm a presença confirmada na atividade e disponibilizará ao inscrito a opção de visualizar e imprimir seus certificados ao executar o caso de uso "Acompanhar inscrição".

Fonte: Elaborado pelo autor

Tabela 7 - Documentação do caso de uso "Realizar inscrição"

Nome do caso de uso	<i>Realizar inscrição</i>
Descrição	Esse caso de uso permite ao usuário externo realizar sua inscrição na SNCT
Ator principal	Usuário externo
Pré-condições	O usuário precisa estar conectado à internet

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário digitará o endereço do site na barra de endereços do navegador e acessará a opção "Me inscrever".	
	2. A aplicação exibirá a tela com o cadastro dos participantes.
3. O usuário preencherá os dados, respeitando os campos obrigatórios e clicará em "Cadastrar".	
	4. A aplicação gravará os dados informados no banco de dados e exibirá uma mensagem de que o cadastro foi efetuado com sucesso e executará o caso de uso "Acompanhar inscrição".

Fonte: Elaborado pelo autor

Tabela 8 - Documentação do caso de uso "Acompanhar inscrição"

Nome do caso de uso	<i>Acompanhar inscrição</i>
Caso de uso geral	Realizar inscrição
Descrição	Esse caso de uso permite ao usuário externo gerenciar sua inscrição na SNCT
Ator principal	Usuário externo
Pré-condições	O usuário precisa estar conectado à internet e já ter executado o caso de uso "Realizar inscrição".

FLUXO PRINCIPAL

Ações do ator	Ações do sistema
1. O usuário digitará o endereço do site na barra de endereços do navegador e acessará a opção "Acompanhar inscrição".	
	2. A aplicação exibirá a tela de login.
3. O usuário preencherá seus dados de acesso e clicará em "Entrar".	
	4. A aplicação consultará no banco de dados se os dados informados são válidos e exibirá a página acessada.
5. O usuário poderá se inscrever em diversas atividades, alterar dados pessoais e de acesso e visualizar e imprimir certificados.	

FLUXO ALTERNATIVO

Ações do ator	Ações do sistema
	1. Caso os dados informados sejam inválidos, a aplicação exibirá mensagem e usuário deverá preencher com dados válidos.

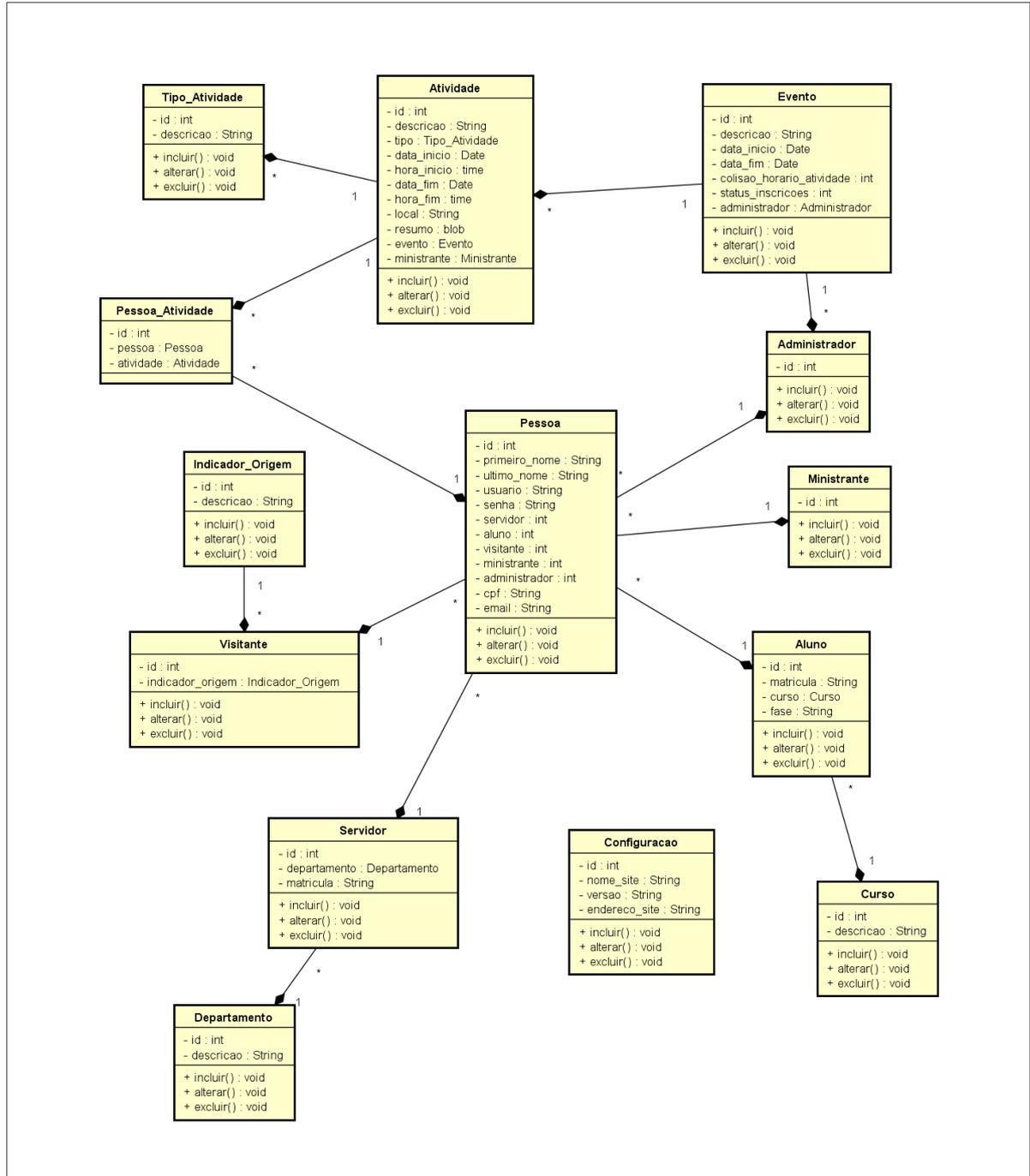
Fonte: Elaborado pelo autor

4.4.2 DIAGRAMA DE CLASSES

Para a elaboração do diagrama de classes foi utilizada a ferramenta Astah Professional.

Este diagrama, embora não esteja refletido na codificação final, foi de suma importância no auxílio de uma visão mais clara de quais funções o sistema deveria apresentar.

Figura 8 - Diagrama de classes



Fonte: Elaborado pelo autor

4.4.3 MODELAGEM DE DADOS

Para a elaboração da modelagem de dados do banco de dados da aplicação, foi utilizada a ferramenta MySQL Workbench, como mostrado no Apêndice A na página 59. Também, através dessa ferramenta, foi utilizado o modelo de dados visuais para transformação em banco de dados físico.

4.4.4 DESENVOLVIMENTO DO PROTÓTIPO

Para o desenvolvimento do protótipo em HTML, CSS, JavaScript e PHP foi utilizada a ferramenta Notepad++.

As telas podem ser visualizadas nos Apêndices deste documento, a partir da página 60.

4.4.5 INTERFACE GRÁFICA

A interface gráfica foi testada nos navegadores Google Chrome, Firefox, Internet Explorer e Opera.

As telas podem ser visualizadas nos Apêndices deste documento, a partir da página 76.

5 CONCLUSÃO

O projeto apresentou a criação de um protótipo de aplicação para a *web*, para o auxílio do Departamento de Pesquisa e Extensão do IFSC, Câmpus Florianópolis.

Esse projeto não foi implementado, na sua totalidade, por ser somente um protótipo e não estar concluído. Após a conclusão, estará disponível ao câmpus Florianópolis para as próximas edições da SNCT. Quando dessa implementação, possivelmente serão necessários ajustes ou inclusão de novas funcionalidades para a completa utilização da aplicação no auxílio ao Departamento de Pesquisa e Extensão do IFSC, Câmpus Florianópolis, podendo, inclusive, ser utilizado por outras unidades.

Percebe-se que, a partir da elaboração deste protótipo, diversas funcionalidades, antes não existentes, agora podem ser utilizadas, facilitando tarefas como: cadastro das informações dos inscritos, gerenciamento das horas das atividades e geração automática dos certificados. Dessa forma, o protótipo facilitará a gestão e a reutilização dessas informações, garantindo o controle das diversas etapas do evento e agilizando os processos e atividades onde estão envolvidos os usuários.

O desenvolvimento deste projeto possibilitou o conhecimento aprofundado de diversos assuntos tratados durante o curso, como: análise de sistemas, modelagem de dados, gerenciamento de projetos, programação *web*, banco de dados, gestão de inovações tecnológicas, processos de desenvolvimento de sistemas, entre outros.

Neste sentido, a presente pesquisa forneceu subsídios para que a compreensão de diversos conceitos da área tecnológica, particularmente de desenvolvimento *web*, fossem fixados e ampliados, mais notadamente, nos seguintes aspectos: a maneira como uma aplicação, utilizando a linguagem PHP, se comunica com um servidor de banco de dados; o funcionamento da arquitetura cliente-servidor; e o entendimento mais profundo de alguns métodos e técnicas que facilitam enormemente a criação de uma aplicação para a *web*.

Como sugestão para a continuidade deste projeto, pode-se considerar a inclusão de novas funcionalidades, como a possibilidade de o administrador personalizar o formulário de inscrição dos participantes e a geração de etiquetas ou crachás de identificação do participante inscrito nas atividades; a possibilidade de se utilizar um leitor de código de barras para validar a presença do participante em cada atividade, bem como a reescrita do código-fonte utilizando-se *frameworks*, para otimizar a manutenção da aplicação. Também, é possível realizar adaptações responsivas para diferentes tipos de dispositivos, como celulares ou *tablets* e padrões de acessibilidade, para que qualquer usuário possa acessar a aplicação.

Finalmente, convém destacar que a tecnologia possibilita, cada vez mais, a diminuição dos esforços para executarmos alguma atividade. Dessa forma, pode-se concluir que, com o uso dessa aplicação, será possível gerenciar a SNCT de forma que não haja retrabalhos, possibilitando, inclusive, tomadas de decisões importantes, como, por exemplo, a escolha do local onde será realizada a atividade, mediante a quantidade de participantes inscritos para aquela atividade; ou então, possivelmente, a geração de gráficos estatísticos relacionados a que tipos de eventos obtiveram maior público, de forma a servir de guia para a escolha de um conjunto de eventos mais atrativos para as próximas edições da SNTC. Estes aspectos reafirmam um dos pilares do curso superior CSTGTI, cujo foco reside na necessidade de criarmos ferramentas que auxiliem a tomada de decisão e a gestão mais eficiente da informação por meio do uso de ferramentas tecnológicas.

REFERÊNCIAS

ALTER, Steven. **Information system: a management perspective**. United States of America: Addison-Wesley Publishing Company, 1992.

BEIGHLEY, Lynn; MORRISON, Michael. **Use a cabeça! PHP & MySQL**. 2ª ed., Rio de Janeiro: Alta Books Editora, 2010.

BEUREN, Ilse Maria; MARTINS, Luciano Waltrick. **Sistema de informações executivas: suas características e reflexões sobre sua aplicação no processo de gestão**. Revista Contabilidade & Finanças, São Paulo, v. 12, n. 26, 2001. Disponível em: <http://www.scielo.br/scielo.php?pid=S1519-70772001000200001&script=sci_arttext&tlng=es>. Acesso em: 21 maio 2017.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2ª ed., Rio de Janeiro, 2000.

CROWTHER, Rob et al. **HTML5 em ação**. São Paulo: Novatec, 2014.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 9ª ed., Rio de Janeiro: Elsevier, 2003.

DEVMEDIA. **Conceitos Fundamentais de Bancos de Dados**. Disponível em: <<http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 22 out. 2017.

FEITOSA, Marcio Porto. **Fundamentos de Bancos de Dados: uma abordagem prático-didática**. São Paulo: 2013.

FLANAGAN, David. **JavaScript: o guia definitivo**. 6ª ed., Porto Alegre: Bookman, 2013.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4ª ed., São Paulo: Atlas S.A., 2002.

GOMES, Jackson. **Desenvolvimento de Software para a Web**. 2017. Disponível em: <<https://www.gitbook.com/book/fsw-ceulp/desenvolvimento-de-software-para-a-web>>. Acesso em: 24 out. 2017.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2ª ed., São Paulo: Novatec Editora, 2011.

KAHLMAYER-MERTENS, Roberto S. et al. **Como elaborar projetos de pesquisa: linguagem e método**. Rio de Janeiro: FGV, 2007.

MESSENLEHNER, Brian; COLEMAN, Jason. **Criando aplicações web com Wordpress**. São Paulo: Novatec Editora, 2014.

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES. **Semana Nacional de Ciência e Tecnologia**. Disponível em: <<http://snct.mctic.gov.br/portal>>. Acesso e: 20 out. 2017.

MYSQL. **10 principais motivos para usar o MySQL como um banco de dados incorporado**. Disponível em: <<https://www.mysql.com/why-mysql/white-papers/10-principais-motivos-para-usar-o-mysql-como-um-banco-de-dados-incorporado/>>. Acesso em: 26 out. 2017.

PHP. **Manual do PHP**. Disponível em: <http://php.net/manual/pt_BR/index.php>. Acesso em: 22 out. 2017.

PRODANOV, Cleber C.; FREITAS Ernani C. de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2ª ed., novo Hamburgo: Feevale, 2013.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de gerenciamento de banco de dados**. 3ª ed., Porto Alegre: AMGH, 2011.

SAMPAIO, Marcus Costa. **Casos de uso**. 2007. Disponível em: <<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm>>. Acesso em: 16 out. 2017.

SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. 9ª ed., São Paulo: Bookman, 2011.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4ª ed., Florianópolis: Universidade Federal de Santa Catarina - UFSC, 2005. Disponível em: <https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf>. Acesso em: 28 maio 2017.

SILVA, Maurício Samy. **Criando sites com HMTL**: sites de alta qualidade com HTML e CSS. 4ª ed., São Paulo: Novatec, 2014.

SIMÕES, Roberto Porto. **Relações públicas**: função política. 3ª ed., São Paulo, 1995.

SKLAR, David. **Aprendendo PHP**: introdução amigável à linguagem mais popular da web. São Paulo: Novatec, 2016.

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

VALENTIM, Marta. **Gestão, mediação e uso da informação**. São Paulo: Cultura Acadêmica, 2010.

VIANNA, Cleverson Tabajara. **Classificação das Pesquisas Científicas**: notas para os alunos. Florianópolis, 2013, 2p. Disponível em: <<https://pt.slideshare.net/cleversontabajara1/metodologia-cientifica-tipos-de-pesquisa-ultimate>>. Acesso em: 28 maio 2017.

W3C. **What is CSS?** Disponível em: <<https://www.w3.org/Style/CSS/Overview.en.html>>. Acesso em: 22 out. 2017.

W3C BRASIL. **Sobre o W3C**. Disponível em: <<http://www.w3c.br/Sobre>>. Acesso em: 20 out. 2017.

W3SCHOOLS. **HTML Introduction**. Disponível em: <https://www.w3schools.com/html/html_intro.asp>. Acesso em: 20 out. 2017.

GLOSSÁRIO

Aplicação web: Sistemas informatizados projetados para utilização em navegadores, acessados através da internet.

Automatizar: Tornar um processo automático, tornando-o mais ágil, eficiente e confiável.

Back-end: Parte da aplicação que implementa a regra de negócio, a qual o usuário não visualiza.

Banco de dados: Agrupamento de informações que se relacionam.

Código-fonte: Conjunto de palavras e símbolos, escritos de forma lógica, com instruções em uma linguagem de programação.

Framework: É um conjunto de funções, classes e convenções, que facilita a execução de tarefas comuns.

Front-end: Parte da aplicação que interage diretamente com o usuário; interface.

Hardware: Componentes da estrutura física de um computador.

Linguagem de programação: Método padronizado para passar instruções a um computador.

Navegador: Traduzido de *browser* ou *web browser*, é um sistema que transforma páginas codificadas em HTML em visualização compreensível para os usuários.

Protótipo: Na área da TI, considera-se o modelo de um sistema a ser apresentado.

Servidor web: Traduzido de *web server*, pode referir-se ao *hardware*, que é um computador que armazena os arquivos que compõem as páginas *web* e faz a entrega ao dispositivo do usuário final, ou ao *software*, que inclui diversos componentes que controlam a forma como o usuário acessa os arquivos hospedados de uma página *web*.

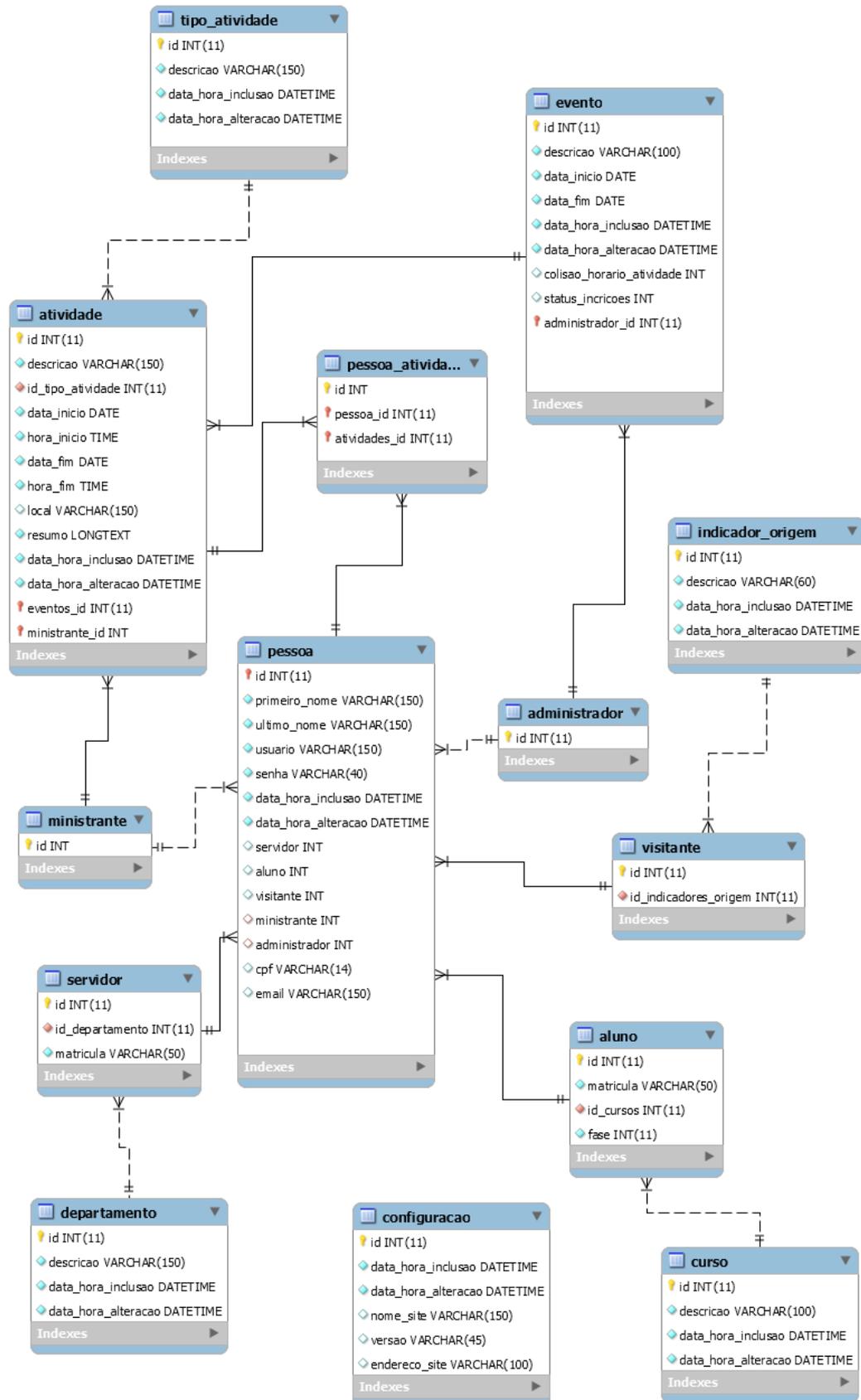
Software: Programas que permitem realizar atividades específicas em um dispositivo.

Stakeholders: Usuários finais e pessoas da organização que serão afetadas pelo *software*.

Usuário: Na área da TI, considera-se a pessoa que utiliza um dispositivo ou computador e que realiza diversas operações em uma aplicação.

Web: Traduzida como rede, a *web* ou *www* é o nome pelo qual a rede mundial de computadores é conhecida (*World Wide Web*).

APÊNDICE A – MODELAGEM DE DADOS



APÊNDICE B – CODIFICAÇÃO DA TELA INDEX.PHP

```

<?php

//Insero o cabeçalho da página
$title_pagina = 'Semana Nacional de Ciência e Tecnologia';
require_once('header.php');

?>

<body>

<header id="cabecalho">

    <!-- início menus -->
    <?php
        require_once('menus.php');
    ?>
    <!-- fim menus -->

    <div class="quebra_linha"></div>

    <!-- início título e período do evento -->
    <div>

        <!-- pega os dados do Evento selecionado (título e período) nas
        configurações -->
        <?php

            require 'includes/dados_conexao.inc.php';
            require 'includes/conectar.inc.php';
            require 'includes/selecionar_banco.inc.php';
            require 'includes/definir_charset.inc.php';

            $consulta_eventos = "SELECT descricao, data_inicio, data_fim
                FROM eventos
                WHERE id = 1";

            $resultado_eventos = mysqli_query($conexao, $consulta_eventos)
or die('Erro ao fazer consulta de Eventos no banco de dados');

            while ($dados_eventos = mysqli_fetch_array($resultado_eventos))
            {

                $descricao = $dados_eventos['descricao'];
                $data_inicio = $dados_eventos['data_inicio'];
                $data_fim = $dados_eventos['data_fim'];

                //definindo o local para português
                setlocale(LC_TIME, 'portuguese');
                date_default_timezone_set('America/Sao_Paulo');

                //transforma o mês para extenso
                $data_inicio_extenso = strftime("%Y-%B-%d",
                strtotime($data_inicio));
                $data_fim_extenso = strftime("%Y-%B-%d",
                strtotime($data_fim));

                //escreve título principal
                echo '<h1 class="titulo_principal">' . $descricao . '</h1>';

```

```

//separar datas - 0 é YYYY, 1 é B e 2 é d
$data_inicio_separada = explode("-", $data_inicio_extenso);
$data_fim_separada = explode("-", $data_fim_extenso);

//escreve período
if ($data_inicio_separada[1] == $data_fim_separada[1])
{
    echo '<h2 class="período">' . $data_inicio_separada[2] .
' à ' . $data_fim_separada[2] . ' de ' . $data_fim_separada[1] . ' de ' .
$data_fim_separada[0] . '</h2>';
}
else
{
    echo '<h2 class="período">' . $data_inicio_separada[2] .
' de ' . $data_inicio_separada[1] . ' à ' . $data_fim_separada[2] . ' de ' .
. $data_fim_separada[1] . ' de ' . $data_fim_separada[0] . '</h2>';
}
}
require 'includes/desconectar.inc.php';
?>
</div>
<!-- fim título e período do evento -->

<!-- início logos -->
<div class="logos">
    <div class="logo_ifsc">
        
    </div>
    <div class="logo_snct">
        
    </div>
</div>
<!-- fim logos -->

</header>

<div class="quebra_linha"></div>

<!-- início inscrição -->
<div class="inscricoes">

    <h3 id="inscricoes" class="titulos"> Inscrições </h3>
    <button class="botoes_inscricao"
onclick="location.href='cadastro_inscritos.php'"> Me inscrever </button>
    <button class="botoes_inscricao"
onclick="location.href='login_inscricoes.php'"> Acompanhar minha inscrição
</button>

</div>
<!-- fim inscrição -->

<div class="quebra_linha"></div>

<!-- início programação -->
<div id="programacao" class="programacao">

    <h3 class="titulos"> Programação </h3>

<?php
    require 'includes/dados_conexao.inc.php';
    require 'includes/conectar.inc.php';

```

```

require 'includes/selecionar_banco.inc.php';
require 'includes/definir_charset.inc.php';

$consulta_atividades = "SELECT
DATE_FORMAT(STR_TO_DATE(ativ.data_inicio, '%Y-%m-%d'), '%d/%m'),
TIME_FORMAT(ativ.hora_inicio, '%H:%i'),
DATE_FORMAT(STR_TO_DATE(ativ.data_fim, '%Y-%m-%d'), '%d/%m'),
TIME_FORMAT(ativ.hora_fim, '%H:%i'), tipo_ativ.descricao, ativ.ministrante,
ativ.local, ativ.descricao, ativ.resumo
FROM atividades ativ
INNER JOIN tipos_atividade tipo_ativ ON
ativ.id_tipo_atividade = tipo_ativ.id
INNER JOIN eventos ev ON ativ.id_evento = ev.id
WHERE ativ.id_evento = 1";

$resultado_atividades = mysqli_query($conexao,
$consulta_atividades) or die('Erro ao fazer consulta de Atividades no banco
de dados');

while ($dados_atividades =
mysqli_fetch_array($resultado_atividades))
{
    $data_inicio = $dados_atividades['0'];
    $hora_inicio = $dados_atividades['1'];
    $data_fim = $dados_atividades['2'];
    $hora_fim = $dados_atividades['3'];
    $tipo_atividade = $dados_atividades['4'];
    $ministrante = $dados_atividades['5'];
    $local = $dados_atividades['6'];
    $descricao = $dados_atividades['7'];
    $resumo = $dados_atividades['8'];

    echo '<div class="lista_atividades">';

        //verifica se a data de início e final da atividade é a
mesma
        if ($data_inicio == $data_fim)
        {
            echo '<p class="lista_atividades_data_horario">' .
$data_fim . ' - ' . $hora_inicio . ' às ' . $hora_fim . '</p><br>';
        }
        else
        {
            echo '<p class="lista_atividades_data_horario">' .
$data_inicio . ' à ' . $data_fim . ' - ' . $hora_inicio . ' às ' .
$hora_fim . '</p><br>';
        }

        echo '<p class="lista_atividades_titulo">' . $tipo_atividade
. ': ' . $descricao . '</p><br>
        <p class="lista_atividades_local_ministrante">' .
$local . ' - Com ' . $ministrante . '</p><br>
        <p class="lista_atividades_resumo">' . $resumo .
'</p></div>';
    }
    require 'includes/desconectar.inc.php';
    ?>
</div>
<!-- fim programação -->

<div class="quebra_linha"></div>

```

```

<!-- início localização -->
<div id="contato" class="localizacao">
  <div class="endereco">

    <!-- início bloco endereço -->
    <div class="enderecol">
      <table class="tabela_mapa">
        <tr>
          <td class="mapa_fonte_titulos " colspan="2"><b>
Instituto Federal de Santa Catarina </b></td>
        </tr>
        <tr>
          <td class="mapa_linha2 mapa_fonte_titulos "
colspan="2"> Campus Florianópolis </td>
        </tr>
        <tr>
          <td class="mapa_colunal"></td>
          <td class="mapa_coluna2.1 mapa_fonte_linhas"> Av.
Mauro Ramos, 950 - Florianópolis/SC</td>
        </tr>
        <tr>
          <td class="mapa_colunal"></td>
          <td class="mapa_coluna2 mapa_fonte_linhas"> (48) 3211-
6000 </td>
        </tr>
        <tr>
          <td class="mapa_colunal"></td>
          <td class="mapa_coluna2 mapa_fonte_linhas">
extensao@ifsc.edu.br </td>
        </tr>
      </table>
    </div>
    <!-- fim bloco endereço -->
  </div>

  <!-- início mapa -->
  <div class="mapa">
    <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d883.994571462257
7!2d-48.54360617083524!3d-
27.594202810372497!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x9527
3839681dbb4b%3A0x0!2zMjFjCzDMlJzMSLjEiUyA0OMKwMzInMzUuMCJX!5e0!3m2!1spt-
BR!2sbr!4v1501460200994" frameborder="0" style="border:0"
allowfullscreen></iframe>
    </div>
    <!-- fim mapa -->
  </div>
  <!-- fim localização -->

  <div class="quebra_linha"></div>

  <!-- início rodapé -->
  <?php
    require once('footer.php');
  ?>
  <!-- fim rodapé -->

```

APÊNDICE C – CODIFICAÇÃO DA TELA CADASTRO_INSCRITOS.PHP

```

<?php
//Insere o cabeçalho da página
$title_pagina = "Cadastro de inscritos";
require_once('header.php');
?>

<script language="javascript" type="text/javascript">
/* função para desabilitar campos quando não é aluno do IFSC */
function desabilita_campos()
{
    document.getElementById('matricula').disabled = true;
    document.getElementById('matricula').style.backgroundColor =
"#D3D3D3";
    document.getElementById('label_matricula').style.color =
"#BEBEBE";
    document.getElementById('id_curso').disabled = true;
    document.getElementById('id_curso').style.backgroundColor =
"#D3D3D3";
    document.getElementById('id_curso').style.cursor = "default";
    document.getElementById('label_curso').style.color = "#BEBEBE";
    document.getElementById('fase').disabled = true;
    document.getElementById('fase').style.backgroundColor = "#D3D3D3";
    document.getElementById('label_fase').style.color = "#BEBEBE";
}

/* função para habilitar campos quando é aluno do IFSC */
function habilita_campos()
{
    document.getElementById('matricula').disabled = false;
    document.getElementById('matricula').style.backgroundColor =
"#FFF";
    document.getElementById('label_matricula').style.color = "#000";
    document.getElementById('id_curso').disabled = false;
    document.getElementById('id_curso').style.backgroundColor =
"#FFF";
    document.getElementById('id_curso').style.cursor = "pointer";
    document.getElementById('label_curso').style.color = "#000";
    document.getElementById('fase').disabled = false;
    document.getElementById('fase').style.backgroundColor = "#FFF";
    document.getElementById('label_fase').style.color = "#000";
}
</script>

<body>

<?php

    require 'includes/dados_conexao.inc.php';
    require 'includes/conectar.inc.php';
    require 'includes/selecionar_banco.inc.php';
    require 'includes/definir_charset.inc.php';
    require 'includes/valida_cpf.php';

    $mensagem_erro = ""; //Limpa a mensagem de erro

    //verifica se o botão Cadastrar foi clicado, para exibir o
    formulário na página
    if (isset ($_POST['botao_cadastrar']))
    {

```

```

        $id_evento = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['id_evento'])));
        $nome = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['nome'])));
        $sobrenome = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['sobrenome'])));
        if (valida_cpf($_POST['cpf']))
        {
            $cpf = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['cpf'])));
        }
        else
        {
            $mensagem_erro = 'O CPF informado é inválido';
            exit();
        }
        if (strcmp($_POST['email'], $_POST['repetir_email']) == 0)
//Verifica se os 2 campos de email são iguais
        {
            $email = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['email'])));
            $repetir_email = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['repetir_email'])));
        }
        else
        {
            $mensagem_erro = "Os emails informados não são iguais.";
            exit();
        }
        if (strcmp($_POST['senha'], $_POST['repetir_senha']) == 0)
//Verifica se os 2 campos de senha são iguais
        {
            $senha = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['senha'])));
            $repetir_senha = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['repetir_senha'])));
        }
        else
        {
            $mensagem_erro = "As senhas informadas não são iguais.";
            exit();
        }
        $id_indicador_origem = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['id_indicador_origem'])));
        $matricula = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['matricula'])));
        $id_curso = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['id_curso'])));
        $fase = mysqli_real_escape_string($conexao,
trim(strip_tags($_POST['fase'])));
        $botao_cadastrar = $_POST['botao_cadastrar'];

        $mostrar_formulario = false;

//Certifica-se que ninguém se cadastrou com o mesmo cpf
        $consulta_cpf_igual = "SELECT *
            FROM inscritos
            WHERE cpf = '$cpf'";

```

```

$registros = mysqli_query($conexao, $consulta_cpf_igual);

if (mysqli_num_rows($registros) == 0)
{
    //O cpf é único, então insere os dados no banco
    $consulta_inscritos = "INSERT INTO inscritos
        (
            id_evento,
            nome,
            sobrenome,
            cpf,
            email,
            senha,
            id_indicador_origem,
            matricula,
            id_curso,
            fase
        )
        VALUES
        (
            '$id_evento',
            '$nome',
            '$sobrenome',
            '$cpf',
            '$email',
            SHA('$senha'),
            '$id_indicador_origem',
            '$matricula',
            '$id_curso',
            '$fase'
        )";

    $resultado_inscritos = mysqli_query($conexao,
    $consulta_inscritos) or die($conexao -> error);

    echo 'O inscrito <strong>' . $nome . ' ' . $sobrenome .
    '</strong> foi cadastrado com sucesso.';

    //Link para voltar à página do cadastro
    echo '</br></br><a class="voltar_formulario"
    href="cadastro_inscritos.php">&larr; Voltar ao formulário de cadastro</a>';

    require 'includes/desconectar.inc.php';

    exit();
}
else
{
    //Já existe uma cadastro com o cpf igual, então exibir
    mensagem de erro
    $mensagem_erro = "O cpf informado já está sendo utilizado.
    Por gentileza, informe um diferente.";

    $cpf = ""; //Limpa o campo para o usuário digitar novamente
}
require 'includes/desconectar.inc.php';
}
else
{
    $mostrar_formulario = true;
}
}

```

```

        //caso o botão Cadastrar não tenha sido clicado, exibe o
formulario
        if ($mostrar_formulario)
        {
            ?>

            <form id="formulario" class="formulario_padrao" method="post"
action="<?php echo $_SERVER['PHP_SELF']; ?>">

                <fieldset>
                    <legend align="center">Cadastro de Inscritos</legend>

                    <div class="campo">
                        <label for="id_evento">Evento:</label>
                        <label class="campo_obrigatorio"
for="id_evento">*</label>
                        <select id="id_evento" name="id_evento" required >

                            <!-- pega os dados já cadastrados da tabela Eventos
do banco de dados -->
                            <?php

                                require 'includes/dados_conexao.inc.php';
                                require 'includes/conectar.inc.php';
                                require 'includes/selecionar_banco.inc.php';
                                require 'includes/definir_charset.inc.php';

                                $consulta_eventos = "SELECT id, descricao
                                    FROM eventos";

                                $resultado_eventos = mysqli_query($conexao,
$consulta_eventos) or die('Erro ao fazer consulta de Eventos no banco de
dados');

                                while ($dados_eventos =
mysqli_fetch_array($resultado_eventos))
                                {
                                    ?>
                                    <option value="<?php echo
$dados_eventos['id']; ?>">
                                        <?php echo $dados_eventos['descricao']; ?>
                                    </option>
                                <?php
                                }

                                require 'includes/desconectar.inc.php';

                                ?>

                            </select>
                        </div>

                        <div class="campo">
                            <label for="nome">Nome:</label>
                            <label class="campo_obrigatorio" for="nome">*</label>
                            <input type="text" id="nome" name="nome" value="<?php
echo @$_nome; ?>" required />
                        </div>

```

```

        <div class="campo">
            <label for="sobrenome">Sobrenome:</label>
            <label class="campo_obrigatorio"
for="sobrenome">*</label>
            <input type="text" id="senha" name="sobrenome"
value="<?php @$sobrenome; ?>" required />
        </div>

        <div class="campo">
            <label for="cpf">CPF:</label>
            <label class="campo_obrigatorio" for="cpf">*</label>
            <input type="text" id="cpf" name="cpf" value="<?php
@$cpf; ?>" required />
        </div>

        <fieldset class="grupo">
            <div class="campo">
                <label for="email">Email:</label>
                <label class="campo_obrigatorio"
for="email">*</label>
                <input type="email" id="email" name="email"
value="<?php @$email; ?>" required />
            </div>

            <div class="campo">
                <label for="repetir_email">Repetir email:</label>
                <label class="campo_obrigatorio"
for="repetir_email">*</label>
                <input type="repetir_email" id="repetir_email"
name="repetir_email" value="<?php @$repetir_email; ?>" required />
            </div>
        </fieldset>

        <fieldset class="grupo">
            <div class="campo">
                <label for="senha">Senha:</label>
                <label class="campo_obrigatorio"
for="senha">*</label>
                <input type="password" id="senha" name="senha"
value="<?php @$senha; ?>" required />
            </div>

            <div class="campo">
                <label for="repetir_senha">Repetir senha:</label>
                <label class="campo_obrigatorio"
for="repetir_senha">*</label>
                <input type="password" id="repetir_senha"
name="repetir_senha" value="<?php @$repetir_senha; ?>" required />
            </div>
        </fieldset>

        <div class="campo">
            <label for="id_indicador_origem">Como ficou sabendo do
evento:</label>
            <label class="campo_obrigatorio"
for="id_indicador_origem">*</label>
            <select id="id_indicador_origem"
name="id indicador origem" required>

                <!-- pega os dados já cadastrados da tabela
Indicadores Origem do banco de dados -->

```

```

<?php
    require 'includes/dados_conexao.inc.php';
    require 'includes/conectar.inc.php';
    require 'includes/selecionar_banco.inc.php';
    require 'includes/definir_charset.inc.php';

    $consulta_indicador_origem = "SELECT id,
descricao
                                FROM indicadores_origem";

    $resultado_indicador_origem =
mysqli_query($conexao, $consulta_indicador_origem) or die('Erro ao fazer
consulta de Indicadores de Origem no banco de dados');

    while ($dados_indicador_origem =
mysqli_fetch_array($resultado_indicador_origem))
    {
        ?>
        <option value="<?php echo
$dados_indicador_origem['id']; ?>">
            <?php echo
$dados_indicador_origem['descricao']; ?>
        </option>
    }
    <?php
    require 'includes/desconectar.inc.php';
    ?>
</select>
</div>

<div class="campo">
    <label for="aluno_ifsc">É aluno do IFSC?</label>
    <label class="campo_obrigatorio"
for="aluno_ifsc">*</label><br>
    <input type="radio" id="aluno_ifsc" name="aluno_ifsc"
value="sim" checked onFocus="habilita_campos()" />Sim<br>
    <input type="radio" id="aluno_ifsc1" name="aluno_ifsc"
value="nao" onFocus="desabilita_campos()" />Não
    </div>

    <div class="campo">
    <label for="matricula">Matricula:</label>
    <input type="text" id="matricula" name="matricula"
value="<?php echo @$matricula; ?>" />
    </div>

    <div class="campo">
    <label for="id_curso" id="label_curso">Curso:</label>
    <select id="id_curso" name="id_curso">

        <!-- pega os dados já cadastrados da tabela Cursos
o banco de dados -->
    <?php

    require 'includes/dados_conexao.inc.php';
    require 'includes/conectar.inc.php';
    require 'includes/selecionar_banco.inc.php';
    require 'includes/definir_charset.inc.php';

```

```

        $consulta_cursos = "SELECT id, descricao
                            FROM cursos";

        $resultado_cursos = mysqli_query($conexao,
        $consulta_cursos) or die('Erro ao fazer consulta de Cursos no banco de
        dados');

        while ($dados_cursos =
        mysqli_fetch_array($resultado_cursos))
        {
            ?>
            <option value="<?php echo
        $dados_cursos['id']; ?>">
                <?php echo $dados_cursos['descricao']; ?>
            </option>
            <?php
            }
            require 'includes/desconectar.inc.php';
            ?>
        </select>
    </div>

    <div class="campo">
        <label for="fase" id="label_fase">Fase:</label>
        <input type="text" id="fase" name="fase" value="<?php
        echo @$fase; ?>" />
    </div>

    <div class="botoes">
        <input type="submit" id="botao_cadastrar"
        name="botao_cadastrar" value="Cadastrar"/>
        <input type="reset" id="botao_limpar"
        name="botao_limpar" value="Limpar dados"/></br></br>
    </div>

    <p>Campos com <strong style="color: red;">*</strong> são
    de preenchimento obrigatório.</p>
    </fieldset>
    </form></br>
    <?php
    echo '<p class="mensagem_erro">' . $mensagem_erro . '</p>';
    }
    ?>
</body>
</html>

```

APÊNDICE D – CODIFICAÇÃO DA TELA CADASTRO_ATIVIDADES.PHP

```

<?php
    //Insere o cabeçalho da página
    $titulo_pagina = "Cadastro de atividades";
    require_once('header.php');
?>

<body>

    <?php

        require 'includes/dados_conexao.inc.php';
        require 'includes/conectar.inc.php';
        require 'includes/selecionar_banco.inc.php';
        require 'includes/definir_charset.inc.php';

        $mensagem_erro = ""; //Limpar mensagem de erro

        //verifica se o botão Cadastrar foi clicado, para exibir o
        formulário na página
        if (isset ($_POST['botao_cadastrar']))
        {
            $id_evento = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['id_evento'])));
            $descricao = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['descricao'])));
            $id_tipo_atividade = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['id_tipo_atividade'])));
            $data_inicio = mysqli_real_escape_string($conexao,
            trim(strip_tags(implode('-',
            array_reverse(explode('/', $_POST['data_inicio'])))))); //Substitui as
            barras '/' por hifens '-' e inverte a ordem da data
            $hora_inicio = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['hora_inicio'])));
            $data_fim = mysqli_real_escape_string($conexao,
            trim(strip_tags(implode('-',
            array_reverse(explode('/', $_POST['data_fim'])))))); //Substitui as barras
            '/' por hifens '-' e inverte a ordem da data
            $hora_fim = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['hora_fim'])));
            $local = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['local'])));
            $ministrante = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['ministrante'])));
            $resumo = mysqli_real_escape_string($conexao,
            trim(strip_tags($_POST['resumo'])));
            $botao_cadastrar = $_POST['botao_cadastrar'];

            $mostrar_formulario = false;

            $consulta_atividades = "INSERT INTO atividades
            (
                id_evento,
                descricao,
                id_tipo_atividade,
                data_inicio,
                hora_inicio,
                data_fim,
                hora_fim,

```

```

        local,
        ministrante,
        resumo
    )
VALUES
    (
        '$id_evento',
        '$descricao',
        '$id_tipo_atividade',
        '$data_inicio',
        '$hora_inicio',
        '$data_fim',
        '$hora_fim',
        '$local',
        '$ministrante',
        '$resumo'
    )";

$resultado_atividades = mysqli_query($conexao,
$conserta_atividades) or die($conexao -> error);

echo 'A atividade <strong>' . $descricao . '</strong> foi
cadastrada com sucesso.';

//Link para voltar à página do cadastro
echo '</br></br><a class="voltar_formulario"
href="cadastro_atividades.php">&larr; Voltar ao formulário de
cadastro</a>';

require 'includes/desconectar.inc.php';

exit();
}
else
{
    $mostrar_formulario = true;
}

//caso o botão Cadastrar não tenha sido clicado, exibe o
formulario
if ($mostrar_formulario)
{
    ?>

    <form class="formulario_padrao" method="post" action="<?php
echo $_SERVER['PHP_SELF']; ?>">

        <fieldset>
            <legend align="center">Cadastro de Atividades</legend>

            <div class="campo">
                <label for="id_evento">Evento:</label>
                <label class="campo_obrigatorio"
for="id_evento">*</label>
                <select id="id_evento" name="id_evento" required>

                    <!-- pega os dados já cadastrados da tabela Eventos
do banco de dados -->

                    <?php

                        require 'includes/dados_conexao.inc.php';

```

```

require 'includes/conectar.inc.php';
require 'includes/selecionar_banco.inc.php';
require 'includes/definir_charset.inc.php';

$consulta_eventos = "SELECT id, descricao
                    FROM eventos";

$resultado_eventos = mysqli_query($conexao,
$consulta_eventos) or die('Erro ao fazer consulta de Eventos no banco de
dados');

while ($dados_eventos =
mysqli_fetch_array($resultado_eventos))
{
?>
<option value="<?php echo
$dados_eventos['id']; ?>">
    <?php echo $dados_eventos['descricao']; ?>
</option>
<?php
}

require 'includes/desconectar.inc.php';

?>

</select>
</div>

<div class="campo">
    <label for="descricao">Descrição:</label>
    <label class="campo_obrigatorio"
for="descricao">*</label></br>
    <input type="text" id="descricao" name="descricao"
value="<?php echo @$descricao; ?>" required />
</div>

<div class="campo">
    <label for="id_tipo_atividade">Tipo:</label>
    <label class="campo_obrigatorio"
for="id_tipo_atividade">*</label>
    <select id="id_tipo_atividade"
name="id_tipo_atividade" required>

        <!-- pega os dados já cadastrados da tabela Tipos
de Atividades do banco de dados -->
        <?php

            require 'includes/dados_conexao.inc.php';
            require 'includes/conectar.inc.php';
            require 'includes/selecionar_banco.inc.php';
            require 'includes/definir_charset.inc.php';

            $consulta_tipos_atividades = "SELECT id,
descricao
                                        FROM tipos_atividade";

            $resultado_tipos_atividades =
mysqli_query($conexao, $consulta_tipos_atividades) or die('Erro ao fazer
consulta de Tipos de Atividades no banco de dados');

```

```

        while ($dados_tipos_atividades =
mysqli_fetch_array($resultado_tipos_atividades))
        {
            ?>
                <option value="<?php echo
$dados_tipos_atividades['id']; ?>">
                    <?php echo
$dados_tipos_atividades['descricao']; ?>
                </option>
            <?php
        }

        require 'includes/desconectar.inc.php';
    ?>
</select>
</div>

<fieldset class="grupo">
    <div class="campo">
        <label for="data_inicio">Data de início:</label>
        <label class="campo_obrigatorio"
for="data_inicio">*</label>
        <input type="date" id="data_inicio"
name="data_inicio" value="<?php @$data_inicio; ?>" required />
    </div>

    <div class="campo">
        <label for="hora_inicio">Hora do início:</label>
        <label class="campo_obrigatorio"
for="data_inicio">*</label>
        <input type="time" id="hora_inicio"
name="hora_inicio" value="<?php echo @$hora_inicio; ?>" required />
    </div>
</fieldset>

<fieldset class="grupo">
    <div class="campo">
        <label for="data_fim">Data do final:</label>
        <label class="campo_obrigatorio"
for="data_fim">*</label>
        <input type="date" id="data_fim" name="data_fim"
value="<?php echo @$data_fim; ?>" required />
    </div>

    <div class="campo">
        <label for="hora_fim">Hora do final:</label>
        <label class="campo_obrigatorio"
for="hora_fim">*</label>
        <input type="time" id="hora_fim" name="hora_fim"
value="<?php echo @$hora_fim; ?>" required />
    </div>
</fieldset>

<div class="campo">
    <label for="local">Local:</label>
    <input type="text" id="local" name="local"
value="<?php echo @$local; ?>"/>
</div>

<div class="campo">
    <label for="ministrante">Ministrante:</label>

```

```

        <input type="text" id="ministrante" name="ministrante"
value="<?php echo @$ministrante; ?>"/>
    </div>

    <div class="campo">
        <label for="resumo">Resumo da atividade:</label>
        <label class="campo_obrigatorio"
for="resumo">*</label>
        <textarea id="resumo" name="resumo" rows="4"
cols="100"><?php echo @$resumo; ?></textarea>
    </div>

    <div class="botoes">
        <input type="submit" id="botao_cadastrar"
name="botao_cadastrar" value="Cadastrar"/>
        <input type="reset" id="botao_limpar"
name="botao_limpar" value="Limpar dados"/></br></br>
    </div>

    <p>Campos com <strong style="color: red;">*</strong> são
de preenchimento obrigatório.</p>
    </fieldset>
</form>

    <?php
        echo '<p class="mensagem_erro">' . $mensagem_erro . '</p>';
    }
?>
</body>
</html>

```

APÊNDICE E – INTERFACE GRÁFICA DA TELA INDEX.PHP

[HOME](#) | [INSCRIÇÕES](#) | [PROGRAMAÇÃO](#) | [ACESSO RESTRITO](#) | [CONTATO](#)

SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA

23 À 25 DE OUTUBRO DE 2017



**INSTITUTO FEDERAL
SANTA CATARINA**



**SEMANA
NACIONAL DE
CIÊNCIA E
TECNOLOGIA 2017**
A MATEMÁTICA ESTÁ EM TUDO!

Inscrições

Me inscrever

Acompanhar minha inscrição

Programação

Não existem atividades cadastradas no momento.



Instituto Federal de Santa Cat...
Av. Mauro Ramos, 950 - Centro,
Florianópolis - SC, 88010-400
5,0 ★★★★★ 4 comentários
[Visualizar mapa ampliado](#)

Instituto Federal de Santa Catarina
Campus Florianópolis

- 📍 Av. Mauro Ramos, 950 - Florianópolis/SC
- ☎ (48) 3211-6000
- ✉ extensao@ifsc.edu.br

©2017 Google - Dados cartográficos ©2017 Google - Termos de Uso - Informar erro no mapa

[f](#)
[t](#)
[g+](#)

APÊNDICE F – INTERFACE GRÁFICA DA TELA CADASTRO_INSCRITOS.PHP

CADASTRO DE INSCRITOS

Evento: *
Semana Nacional de Ciência e Tecnologia ▼

Nome: *

Sobrenome: *

CPF: *

Email: *

Repetir email: *

Senha: *

Repetir senha: *

Como ficou sabendo do evento: *
Internet ▼

É aluno do IFSC? *
 Sim
 Não

Matrícula:

Curso:
Gestão de TI ▼

Fase:

Campos com * são de preenchimento obrigatório.

APÊNDICE G – INTERFACE GRÁFICA DA TELA CADASTRO_ATIVIDADES.PHP

CADASTRO DE ATIVIDADES

Evento: *

Descrição: *

Tipo: *

Data de início: * Hora do início: *

Data do final: * Hora do final: *

Local:

Ministrante:

Resumo da atividade: *

Campos com * são de preenchimento obrigatório.

APÊNDICE H – DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO



INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o(a) estudante TALISSA REGINA OCCHI BUGKELZ,
matrícula nº 141002618-3, do Curso Superior de
GESTÃO DA TECNOLOGIA DA INFORMAÇÃO, defendeu o trabalho intitulado
PROJETO DE UMA APLICAÇÃO PARA A WEB: SEMANA NACIONAL
DE CIÊNCIA E TECNOLOGIA DO IFGC CAMPUS FLORIANÓPOLIS,
o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de
Santa Catarina, Câmpus Florianópolis.

Florianópolis, 01 de FEVEREIRO de 2018.

Herval Jaminelli

Prof. Orientador do TCC