

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-
GRANDENSE - IFSUL, *CAMPUS* PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

THIAGO BARBIERI DO AMARAL

**DESENVOLVIMENTO DE UM APLICATIVO PARA O CONTROLE DE
EMPRESTIMOS DE MATERIAIS DO IFSUL *CAMPUS* PASSO FUNDO,
APLICANDO O CONCEITO DE WEB DESIGN RESPONSIVO**

**PASSO FUNDO
2015**

THIAGO BARBIERI DO AMARAL

**DESENVOLVIMENTO DE UM APLICATIVO PARA O CONTROLE DE
EMPRESTIMOS DE MATERIAIS DO IFSUL *CAMPUS* PASSO FUNDO,
APLICANDO O CONCEITO DE WEB DESIGN RESPONSIVO**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense, Campus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Me. Adilso Nunes de Souza

**Passo Fundo
2015**

THIAGO BARBIERI DO AMARAL

**DESENVOLVIMENTO DE UM APLICATIVO PARA O CONTROLE DE
EMPRESTIMOS DE MATERIAIS DO IFSUL *CAMPUS* PASSO FUNDO,
APLICANDO O CONCEITO DE WEB DESIGN RESPONSIVO**

Trabalho de Conclusão de Curso aprovado em ____/____/____ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

Prof. Me. Adilso Nunes de Souza
(Orientador)

Prof. Me. Jorge Luis Boeira Bavaresco
Avaliador

Prof. Me. Jair José Ferronato
Avaliador

Prof. Me. Adilso Nunes de Souza
Coordenador do Curso

**PASSO FUNDO
2015**

*Aos meus pais pelo amor e incentivo,
isto tudo eu devo a vocês*

AGRADECIMENTOS

Quero começar agradecendo aos meus pais, João e Maria, que com o amor e incentivo deles fizeram de tudo para que eu nunca desistisse do meu objetivo e, que agora veio a se concretizar com a conclusão do curso. Também agradeço aos meus irmãos Bruna e Wilian e a toda a minha família que sempre me apoiou.

Agradeço também ao meu orientador, Professor Me. Adilso Nunes de Souza, que não mediu esforço e dedicação absoluta nas orientações deste trabalho.

Em uma parte desta longa caminhada, tive a oportunidade de participar de uma vida acadêmica ativa, sendo convidado a participar de dois projetos de extensões que, abriram as portas para novos desafios. Então agradeço a Maqueli E. Piva por esta oportunidade, pelo incentivo e apoio. Agradeço o Professor Me. André Fernando Rollwagen pela experiência que me proporcionou na participação do seu projeto. Agradeço a Adriana Schleder por tudo que fez por mim.

Quero agradecer a todos os meus amigos que tiveram paciência e compreenderam a minha ausência em muitos momentos. A todos vocês o meu muito obrigado. Agradeço também a Marta Almeida, secretária do *campus* que me auxiliou no levantamento dos requisitos do meu trabalho e, ajudou quando estava fazendo teste com a aplicação.

Por fim, quero agradecer a todos professores do *campus* e funcionários pela dedicação que têm com o trabalho e com todos os alunos.

“Cada sonho que você deixa pra trás,
é um pedaço do seu futuro que deixa de existir.”
Steve Jobs

RESUMO

O constante aumento do acesso à Internet por dispositivos móveis, aliado ao crescente número de aparelhos comercializados, foram os alicerces que impulsionaram a realização deste estudo. É interessante compreender como desenvolver aplicações voltadas para este público: o usuário de dispositivos móveis. Sobretudo, é preciso saber como desenvolver projetos para a plataforma web que atendam os mais variados tipos de resoluções de telas com um único layout. É exatamente este o principal objetivo deste projeto: desenvolver um sistema que atenda esta necessidade, propondo explorar o conceito do web design responsivo para tratar deste problema. Assim, desenvolveu-se um sistema para o controle de empréstimos de materiais do IFSUL *campus* Passo Fundo. Este serviu como um estudo de caso para detalhar a linguagem utilizada para a modelagem do sistema de acordo com os requisitos levantados, estruturando o projeto em diagramas. Também foram apresentadas as melhorias das atuais linguagens de desenvolvimento como: HTML5 e CSS3. Outras linguagens também foram abordadas, assim como frameworks que auxiliam na elaboração de sites que forneçam uma excelente visualização, navegação e leitura, redimensionamento mínimo com contingência de visionamento, sendo desenvolvido para vários dispositivos (desktops, tablets, smartphones etc.) e, também, apresentadas ferramentas como ScreenQueries, Reposinator e o Device Mode do Google Chrome para testar a adaptação do sistema desenvolvido em simuladores de dispositivos e rede de internet. Nos últimos capítulos deste trabalho é apresentado os resultados alcançados após a conclusão do estudo.

Palavras-chave: Dispositivos móveis. Resoluções de telas. Web design responsivo. Frameworks. Ferramentas de teste.

ABSTRACT

The frequent increase of Internet access by mobile devices, combined with growing number of equipment sold, were the foundation for accomplishing this study. It is interesting to understand how to develop applications for mobile users. Mostly, it is more important to know how to develop projects for web platforms that comply with the most varied types of screen resolutions with a single layout. This is exactly the objective of this project: to develop a system that meets this need, exploring the concept of responsive web design to resolve such problem. Therefore, a system that controls the loans of material at IFSUL campus Passo Fundo was developed. It served as a case study to detail the language used to model the system according to the raised requirements, structuring the project in diagrams. It was also presented the improvements on current development programming languages like HTML5 and CSS3. Other programming languages were also discussed, as well as frameworks that support the development of websites that provide an excellent view, navigation and reading, minimum resizing with vision contingency, being developed for many devices (desktops, tablets, smartphones) and also, presenting tools such as ScreenQueries, Reposinator and Google Chrome Device Mode to check the adjustment of the developed system in devices simulators and internet. In the last few chapters of this project, the results archived from this study were presented.

Keywords: Mobile devices. Screen resolutions. Responsive web design. Frameworks. Testing tools.

LISTA DE TABELAS

Tabela 1 - Lista de regras de keyframes e todas as propriedades de animações	24
Tabela 2 - Perguntas com as repostas do questionário aplicado aos usuários.....	75

LISTA DE FIGURAS

Figura 1 - Evolução do uso de conexão móvel.....	10
Figura 2 - Exemplo do conceito do <i>mobile first</i>	12
Figura 3 - Exemplo de representação de um site desenhado para ser responsivo	13
Figura 4 - Exemplos de seletores.....	22
Figura 5 - Exemplo de borda arredondada.....	23
Figura 6 - Exemplo de efeito de sombra em texto e elemento	23
Figura 7 - Exemplo de efeitos obtidos pelo uso das palavras chaves	24
Figura 8 - Exemplo de código em HTML.....	27
Figura 9 - Representação em árvore do DOM do documento da figura 8	27
Figura 10 - Funcionamento do PHP	33
Figura 11 - Arquivos do Bootstrap	38
Figura 12 - Opções de customização do Bootstrap	39
Figura 13 - Navegadores que suportam o Bootstrap.....	40
Figura 14 - Lista com os principais atributos não suportados pelo navegador IE versão 7 e 8	40
Figura 15 - Customização do Foundation	42
Figura 16 - Modelos de template do Foundation.....	43
Figura 17 - Exemplo de emulação do site do Bootstrap no ScreenQueries	45
Figura 18 - Exemplo de emulação no Reponsinator dos modelos do iPhone 5 e 6.....	46
Figura 19 - Exemplo de emulação do site Bootstrap no Device Mode do Google Chrome	47
Figura 20 - Etapas da metodologia utilizada	48
Figura 21 - Diagrama de casos de uso: secretaria	53
Figura 22 - Diagrama de casos de uso: professor/coordenador.....	54
Figura 23 - Diagrama de casos de uso: usuário	55
Figura 24 - Diagrama de classe do sistema proposto.....	56
Figura 25 - Modelo de funcionamento do sistema desenvolvido.....	59
Figura 26 - Tela inicial: login.....	60
Figura 27 - Tela inicial: devolver item	60
Figura 28 - Tela inicial: nova senha.....	61
Figura 29 - Simulação do acesso da área da secretaria em um dispositivo móvel	61

Figura 30 - Padrões do sistema: botão novo	62
Figura 31 - Padrões do sistema: formulário	62
Figura 32 - Padrões do sistema: botões alterar e excluir.....	63
Figura 33 - Padrões do sistema: registro por página e busca	64
Figura 34 - Padrões do sistema: paginação	64
Figura 35 - exemplo de um campo obrigatório	65
Figura 36 - Trecho de código para um campo obrigatório.....	65
Figura 37 - exemplo de um campo de e-mail.....	65
Figura 38 - Trecho de código de um campo e-mail.....	66
Figura 39 - Padrões do sistema: mensagem de sucesso	66
Figura 40 - mensagem de advertência	66
Figura 41 - mensagem de erro	67
Figura 42 - mensagem informativa do professor.....	67
Figura 43 - Teste do sistema no Mode Device do Google Chrome	68
Figura 44 - exemplo de listagem de pessoas	69
Figura 45 - media querie desenvolvida para a listagem dos dados da tabela de pessoas	70
Figura 46 - formulário de prédios	71
Figura 47 - Gráfico sobre as funcionalidades do sistema	72
Figura 48 - Gráfico sobre a agilidade do sistema desenvolvido.....	72
Figura 49 - Gráfico sobre o tempo de acesso ao sistema	73
Figura 50 - Gráfico sobre a utilização do sistema	73
Figura 51 - Gráfico sobre a navegação no sistema	74
Figura 52 - Gráfico sobre o acesso ao sistema utilizando dispositivo móvel	74

LISTA DE ABREVIACOES E SIGLAS

ANSI – *American Nacional Standards Institute*
API – *Application Programming Interface*
CSS – *Cascading Style Sheets*
DOM – *Document Object Model*
IP – *Internet Protocol*
GIF – *Graphics Interchange Format*
GPL – *General Public License*
HTML – *Hypertext Markup Language*
IDE - *Integrated Development Environment*
ISO – *International Standards Organization*
MIT – *Massachusetts Institute of Technology*
OOCSS – *Object-Oriented Cascading Style Sheets*
PDF – *Portable document Format*
PHP – *PHP: Hypertext Preprocessor*
SQL – *Structured Query Language*
URL – *Uniform Resource Locator*
XHTML – *eXtensible Hypertext Markup Language*
XML – *eXtensible Markup Language*
W3C – *World Wide Web Consortium*
WHATWG – *Web Hypertext Application Technology Working Group*

SUMÁRIO

1.	INTRODUÇÃO	7
1.1	MOTIVAÇÃO	8
1.2	OBJETIVOS	8
1.2.1	Objetivo geral	8
1.2.2	Objetivos específicos.....	8
1.2.3	Estrutura da dissertação.....	9
2	REFERENCIAL TEÓRICO	10
2.1	WEB MÓVEL	10
2.2	<i>MOBILE FIRST</i>	11
2.3	WEB DESIGN RESPONSIVO	13
2.3.1	<i>Layouts</i> fluidos.....	14
2.3.2	Tipos de medidas em CSS	15
2.3.3	Imagens e recursos flexíveis	16
2.4	UML.....	17
2.5	DIAGRAMAS DE CASO DE USO	18
2.6	DIAGRAMA DE CLASSE	18
2.7	HTML5	19
2.7.1	Os novos atributos.....	20
2.8	CSS3	22
2.8.1	Media queries.....	26
2.9	JAVASCRIPT.....	26
2.9.1	jQuery	29
2.9.1.1	jQuery mobile	30
2.10	PHP	32
2.10.1	Funcionamento do PHP.....	32
2.10.2	Capacidade do PHP.....	34
2.11	SQL: INTRODUÇÃO AO BANCO DE DADOS RELACIONAL	34
2.11.1	PostgreSQL.....	36
2.12	FRAMEWORK PARA O DESENVOLVIMENTO DE APLICAÇÕES RESPONSIVAS	37
2.12.1	Bootstrap.....	37

2.12.2	Foundation	41
2.12.3	Skeleton	44
2.13	FERRAMENTAS PARA AVALIAR O DESIGN RESPONSIVO	44
2.13.1	Screenqueries	45
2.13.2	Responsinator.....	45
2.13.3	Device Mode do Google Chrome	46
3	METODOLOGIA	48
3.1	CLASSIFICAÇÃO DA PESQUISA.....	48
3.2	ETAPAS DO PROJETO	48
3.2.1	Pesquisa bibliográfica	49
3.2.2	Análise do sistema existente	49
3.2.3	Modelagem do sistema proposto	50
3.2.3.1	Requisitos funcionais	50
3.2.3.2	Requisitos não funcionais.....	52
3.2.3.3	Diagrama de casos de uso.....	52
3.2.3.3.1	Diagrama casos de uso – área da secretaria.....	52
3.2.3.3.2	Diagrama de casos de uso – área do professor/coordenador	54
3.2.3.3.3	Diagrama de casos de uso – área do usuário	55
3.2.3.4	Diagrama de classes do sistema proposto	56
3.2.4	Desenvolvimento do sistema	57
3.2.5	Teste no sistema desenvolvido	58
4.	SISTEMA DESENVOLVIDO	59
4.1.	FUNCIONAMENTO DO SISTEMA	59
4.1.1.	Tela inicial	59
4.1.2.	Interface da área da secretaria.....	61
4.1.3.	Padrões do sistema	62
4.1.4.	Validações dos campos editáveis	64
4.1.5.	Mensagens do sistema	66
4.1.6.	Teste com o device mode do google chrome.....	67
4.1.7.	Teste com os usuários	71
5.	CONSIDERAÇÕES FINAIS	77
6.	REFERÊNCIAS.....	79
7.	APÊNDICES E ANEXOS.....	82

1. INTRODUÇÃO

Cada vez mais as pessoas estão acessando a internet por meio de dispositivos móveis. Este hábito, vem se consolidando em virtude do número de aparelhos vendidos, como smartphones, que cresce a cada ano. Só no Brasil, em 2014, mais de 43 milhões de pessoas estavam acessando à internet por estes dispositivos, de acordo com os dados do Instituto de pesquisas Datafolha. Ainda conforme a pesquisa, 1 em cada 4 entrevistados, pensavam em comprar um smartphone no primeiro semestre daquele ano. (DATAFOLHA, 2014)

Segundo pesquisa realizada no Brasil pelo Instituto Brasileiro de Pesquisas e Estatísticas (IBGE), divulgado em 2015, revelou que, em 2013, 57,3% das residências acessavam a internet por tablets ou smartphones, mas que, ainda assim, os microcomputadores eram os principais meios de acesso à internet. (SALES, 2015)

Com base nestes dados, é possível afirmar que, os usuários buscam fazer acesso à web pelo alto grau de sofisticação destes aparelhos. A capacidade de processamento dos tablets e smartphones que encontram-se disponível no mercado, podem igualar-se aos desktops. A evolução do hardware destes dispositivos não estão limitados quando comparado aos seus antecessores. Estes dispositivos móveis tem como principal característica, seu uso multifuncional.

Com este número crescente de pessoas acessando a internet, sobretudo por dispositivos móveis, observa-se problemas na experiência dos usuários quanto ao acesso dos websites, como a visualização em telas menores e o redimensionamento da janela do navegador, gerando assim, um desconforto na navegação do site. Conforme Souza e Igarashi (2012), os usuários de dispositivos móveis estão sendo prejudicados na navegação de websites, devido ao tamanho da dimensão do visor dos aparelhos, uma vez que estes possuem uma resolução inferior dos monitores nos quais os websites são produzidos para serem apresentáveis.

Desta forma, este trabalho tem como objetivo apresentar o conceito de web design responsivo como forma de solucionar o problema de navegação de websites por diversos tipos de dispositivos. Logo, foi desenvolvido um aplicativo de controle de empréstimos de materiais, para o IFSUL *campus* Passo Fundo, como objeto de estudo de caso, onde foi aplicado os conceitos a partir de estudo de metodologias, na aplicação e desenvolvimento de um único layout que atenda as resoluções de aparelhos que estiverem acessando o aplicativo.

1.1 MOTIVAÇÃO

A motivação pela elaboração deste trabalho surge a partir da constatação que cada vez mais as pessoas estão conectadas à internet por diferentes tipos de dispositivos. A cada ano, o número de vendas de aparelhos como smartphones só aumentam e com isto, o acesso à internet por estes dispositivos também, ou seja, as pessoas estão utilizando mais a internet por estes aparelhos.

Tendo como premissa esta afirmação, o presente trabalho foi elaborado afim de mostrar como desenvolver um sistema para web que atendam as requisições de telas dos dispositivos que precisam acessar aplicações. Desta forma, foi desenvolvido um sistema como objeto de estudo de caso, solucionando um problema do controle de empréstimos de materiais do IFSUL campus Passo Fundo, onde este controle é feito de forma manual e sem a utilização de recursos tecnológicos.

1.2 OBJETIVOS

Nesse capítulo, são apresentados o objetivo geral e os objetivos específicos a serem atingidos com a conclusão deste trabalho.

1.2.1 Objetivo Geral

Modelar e desenvolver um aplicativo para o controle de empréstimos de materiais do IFSUL *campus* Passo Fundo, aplicando o conceito do web design responsivo.

1.2.2 Objetivos específicos

- Conceituar o acesso à internet por dispositivos móveis e o web design responsivo;
- Realizar um levantamento dos conceitos das linguagens que irão compor o projeto;
- Pesquisar os tipos frameworks e ferramentas para o auxílio de implementação e testes do aplicativo;
- Fazer a análise e projeto do sistema proposto;
- Desenvolver, implantar e testar o sistema de controle de empréstimos no IFSUL.

1.2.3 Estrutura da monografia

O presente capítulo foi feito uma abordagem contextual destacando a motivação junto os objetivos desta monografia, os próximos capítulos estão organizados da seguinte forma:

- **Capítulo 2:** este é o capítulo com a pesquisa bibliográfica abordando inicialmente o acesso à internet por dispositivos móveis correlacionando ao desenvolvimento de aplicações web para estes dispositivos, descrevendo o conceito de design responsivo. Também serão abordadas as linguagens utilizadas no desenvolvimento deste projeto, bem como, os principais frameworks disponíveis para o desenvolvimento de aplicações responsivas, fazendo uma análise particular de cada um deles. Por fim, as mesmas análises serão feitas com três ferramentas de avaliação do design responsivo.
- **Capítulo 3:** é apresentado a metodologia utilizada no desenvolvimento deste projeto, quais foram suas etapas e o que foi realizado em cada uma delas.
- **Capítulo 4:** dedicado a demonstrar o sistema desenvolvido com os padrões do sistema e os resultados dos testes aplicados.
- **Capítulo 5:** por fim são apresentadas as considerações finais sobre o estudo e o projeto desenvolvido, destacando o aprendizado construído, as dificuldades encontradas, além de apresentar os estudos futuros que buscam complementar os conhecimentos necessários sobre o assunto.

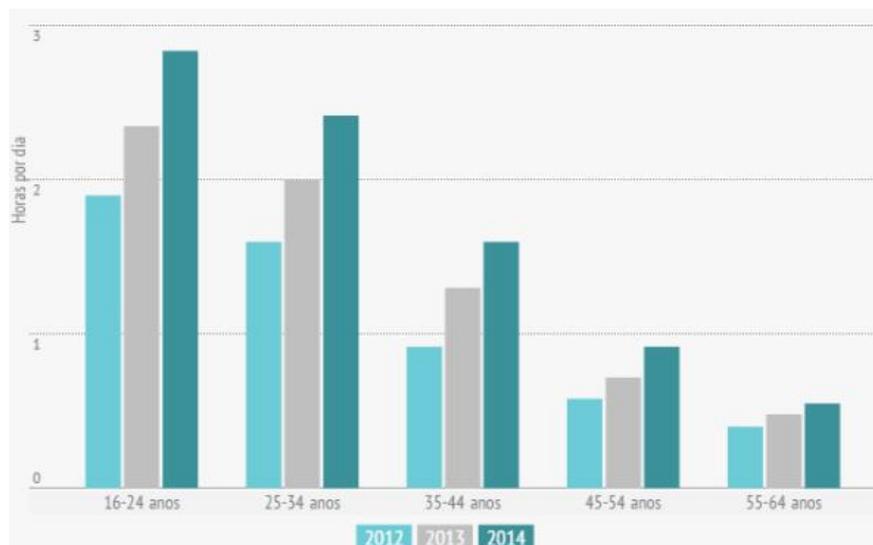
2 REFERENCIAL TEÓRICO

Neste capítulo são abordados assuntos relacionados ao design responsivo, as linguagens utilizadas tanto na modelagem da aplicação quanto no desenvolvimento e, também as características dos três principais frameworks e ferramentas disponíveis para o desenvolvimento e testes respectivamente de aplicações responsivas.

2.1 WEB MÓVEL

O acesso à internet não está mais restrito a desktops, outros aparelhos como smartphones e tablets, possuem funcionalidades de conexão com a rede. Segundo o jornal Correio Brasiliense (2015), a empresa GlobalWebIndex¹ especializada em internet e tecnologia, realizou um estudo em 2014 em mais de 32 países, incluindo o Brasil, sobre o uso de dispositivos digitais que contam com acesso à internet. Neste estudo, os participantes responderam perguntas sobre o uso do dispositivo móvel, o acesso à internet por estes aparelhos. Destaca-se aqui, o tempo de uso da internet móvel segundo a pesquisa, como mostra a figura 1:

Figura 1 - Evolução do uso de conexão móvel



Fonte: Correio Brasiliense, (2015)

Na figura acima, é possível notar que entre os anos de 2012 e 2014, houve um aumento em todas as faixas etárias quanto ao tempo de utilização da internet em dispositivos

¹ <https://www.globalwebindex.net/>

móveis. As pessoas estão acessando e passando mais tempo conectados na Web através destes aparelhos. Desta forma, é preciso levar em consideração o conceito de Web Mobile. Conforme Bouviér, o conceito de web mobile pode ser definido como:

[...]a forma de acessar a Web utilizando um dispositivo móvel, acessando e interagindo com qualquer página da Web independentemente do lugar de acesso, podendo ser de qualquer lugar do planeta que disponha de uma conexão. (2013, p. 19)

A experiência de navegação na Web é um ponto que chama a atenção, pois, existem sites que não possuem recursos de adaptação as resoluções de telas do dispositivo, trazendo uma experiência negativa para o usuário.

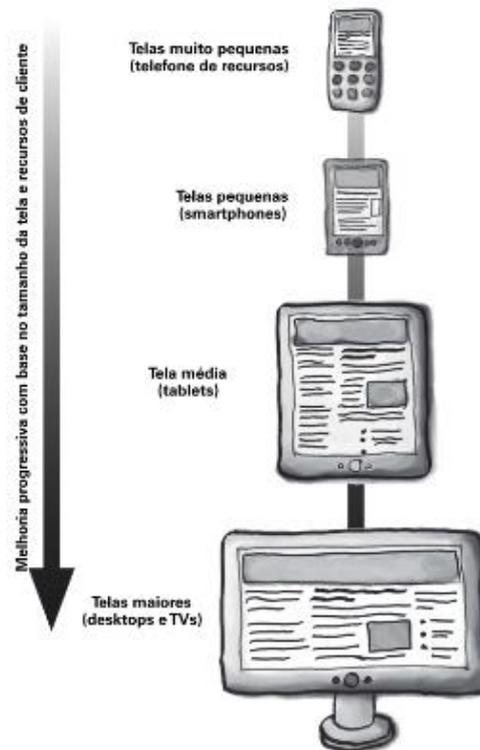
Na figura 1 observa-se que os usuários de uma maior faixa etária utilizam em menor número os dispositivos moveis para acesso à internet pois, estão melhor adaptados aos desktops. Já a geração mais nova surgiu na era dos smartphones facilitando e propiciando uma maior utilização destes dispositivos.

2.2 MOBILE FIRST

Com a popularização do uso dos dispositivos moveis sobre tudo, os smartphones, atrai a atenção para um novo formato a ser projetado nas aplicações web. O crescimento do acesso à internet por estes dispositivos, requerem aplicações que atendam as requisições quanto a visualização por estes aparelhos.

O *mobile first* é um padrão de desenvolvimento de aplicações web seguindo o conceito de web design responsivo onde, é desenvolvido um template baseado nas plataformas mobile e, progressivamente, o projeto segue uma serie de camadas afim de atender outros dispositivos conforme a necessidade e objetivo de cada projeto. A figura 2 demonstra a ilustração do *mobile first*:

Figura 2 - Exemplo do conceito do *mobile first*



Fonte: GARDNER, GRIGSBY – (2013)

Conforma Gardner e Grigsby (2013), ao projetar para telas “muito pequenas”, usa-se uma estrutura do *Hypertext Markup Language* (HTML) básico, projeta-se um layout básico onde deve ser utilizar imagens pequenas, com *Cascading Style Sheets* (CSS) e JavaScript limitados. Já para a estrutura de telas “pequenas”, algumas características começam a mudar em relação à anterior, como: desenvolver a aplicação utilizando os recursos da linguagem HTML5, considerando se, estes dispositivos dão suportes a linguagem, o layout pode ser simples, as imagens devem ser pequenas porém, podem ser maiores do que o projeto anterior e, contar com mais recursos do CSS e JavaScript. Na categoria “tela média”, considerando ter mais espaços que os dois projetos anteriores, podem ser acrescentados recursos opcionais como por exemplo, barras laterais, além de poder inserir no layout múltiplas colunas e imagens com alta resolução. Por fim, no projeto “telas maiores”, o layout pode ser do tipo *widescreen*² contando com imagens pesadas.

Sendo assim, o *mobile first* é um conceito para a inovação na maneira de como são projetadas as aplicações para web onde, deve-se planejar e construir primeiro para pequenos

² *Widescreen* é um termo utilizado da língua inglesa que em português diferencia os aparelhos como as TVs de telas maiores.

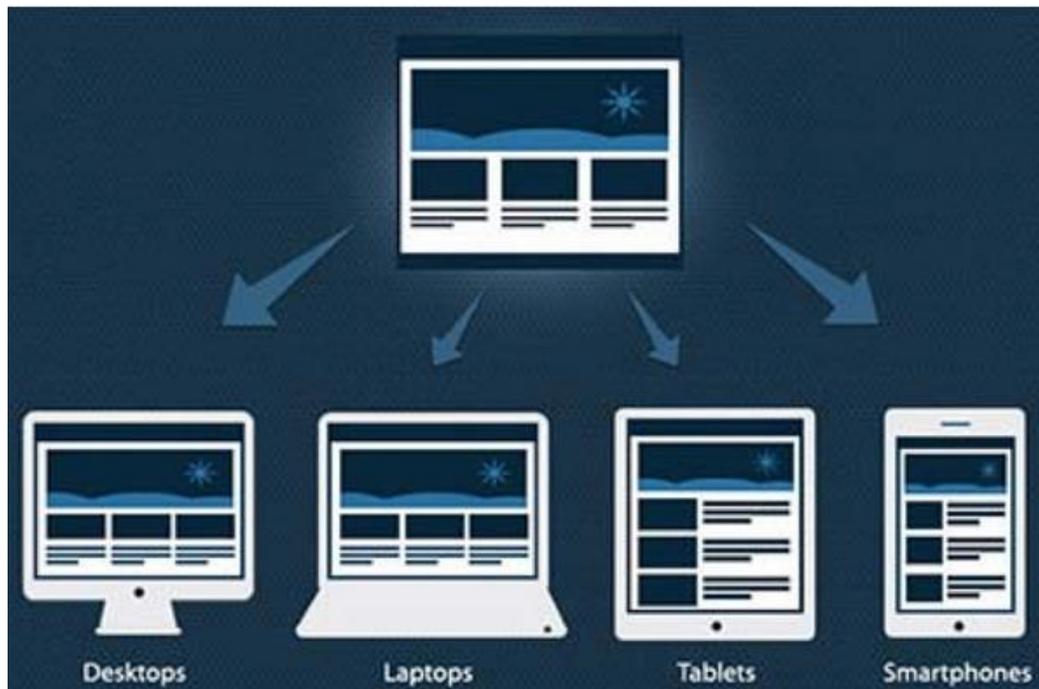
dispositivos e, gradativamente, acrescentado melhorias a fim de atender por último os dispositivos desktops que possuem resoluções de telas maiores.

2.3 WEB DESIGN RESPONSIVO

Segundo Zemel (2012), o termo web design responsivo surgiu em meados de 2010 quando o escritor Ethan Marcotte do site *A List Apart*³, escreveu um artigo intitulado de *Responsive Web Design*, o que acabou gerando um novo conceito de design para web.

Compreende-se web design responsivo um conjunto de técnicas e procedimentos que visam a criação de layouts de páginas webs que se adaptem em diferentes tipos de dispositivos e navegadores que estejam acessando o conteúdo (SILVA, 2013). O layout de um site, que foi desenhado seguindo os conceitos de web design responsivo, se adaptará aos mais variados tipos de resoluções de telas e navegadores, que está definido nas regras do CSS. A figura 3 mostra um exemplo de site desenhado para ser responsivo.

Figura 3 - Exemplo de representação de um site desenhado para ser responsivo



Fonte: HOPMA, (2014).

Silva (2013), no livro *jQuery Mobile*, descreve os principais aspectos do conceito web design responsivo:

³ <http://alistapart.com/>

- Desenvolver um código para dispositivo móvel que servira para outros dispositivos e plataformas;
- Botões, links e elementos interativos serão acessíveis nos dispositivos moveis;
- Desenvolver seguindo o conceito de *mobile first*, ou seja, desenvolver primeiro para dispositivo móvel e depois fazer melhoria progressiva para desktops;
- layout tem que ser adaptável a qualquer tamanho de tela;
- Usar as funcionalidades que já são nativas dos dispositivos moveis;
- As imagens precisam ser redimensionadas a qualquer tipo de resolução.

Portanto, o web design responsivo busca solucionar os problemas com as visualizações do site de forma que adapta-se com o tamanho da tela do dispositivo acessado. Esta é uma técnica inovadora e que passa a ser tendência, pois, o número de aparelhos móveis e com diferentes tipos de resoluções de telas cada vez mais estão conectados à internet.

2.3.1 *Layouts fluidos*

“*Layouts* fluidos estão sendo utilizados desde os primórdios, mesmo assim de uma forma muito restrita porque dependendo do tamanho do site são bem difíceis de planejar.” (EIS, 2011)

Os *layouts* fluidos são utilizados nas páginas web com medidas não fixas como os atributos *em*⁴ e porcentagem no CSS. Com este método, as páginas se comportam de acordo com as resoluções do dispositivo pelo qual estão sendo visualizadas. Esta é uma das maneiras de trabalhar com páginas responsivas.

Os sites que são desenvolvidos com este tipo de *layout* ocupam todo o espaço das telas, sem ficar com áreas em branco, diferente dos sites que são desenvolvidos com um layout de largura fixa em torno de 960 *pixels*, por exemplo.

Segundo Zemel (2012), existe uma fórmula simples para a conversão de sites com medidas absolutas em medidas relativas. Conforme o autor, segue a “fórmula mágica do web design responsivo”:

$$\text{Alvo/Contexto} = \text{Resultado}$$

⁴*em*: É uma unidade escalável de medida.

- Alvo: Representa a medida atual do elemento alvo;
- Contexto: Onde se encontra o elemento alvo (elemento pai);
- Resultado: valor relativo procurado.

Considere uma seguinte situação: o elemento *body* é o elemento pai sendo assim, ele representará 100% do *viewport* de um navegador. Neste caso, o *body* estará ocupando 100% do *viewport* independente de qual seja a dimensão da resolução da tela de um determinado dispositivo. Agora considerando um container sem definições das dimensões inserido diretamente dentro do *body*, automaticamente ele assumirá uma largura correspondente a da resolução do *viewport*. Após a inserção de mais um *container*, mas, desta vez, com uma largura declarada de 25%, este estará ocupando dentro do *viewport*, uma largura de um quarto do primeiro container inserido anteriormente. Neste caso, o segundo container está assumindo uma dimensão relativa ao do primeiro container, ou seja, está responsivo.

Finalizando, digamos que o elemento pai, neste exemplo o primeiro container, tenha a largura de 600px e, sabendo que o nosso segundo container tem uma largura que representa 25% da largura do primeiro container, ou seja, 150px, podemos então considerar que, o nosso alvo é o segundo container, que será dividido pelo nosso contexto (primeiro container), desta forma, teríamos um valor em porcentagem declarada para o segundo container de $0.25em$.

Assim conforme Zemel (2012), está é uma fórmula de cálculo rápido e simples para saber os resultados das conversões das medidas absolutas do CSS para medidas relativas. Desta forma, “[...]isso vale tanto para o cálculo de tamanho de fontes[...], quanto para medidas de layout!” (ZEMEL, 2012, p. 26)

2.3.2 Tipos de medidas em CSS

Um dos itens mais importantes quando se fala em design responsivo é compreender adequadamente o CSS, pois este, representa a forma como o layout do site será visualizado no navegador, sendo um layout fixo ou fluido.

Desta forma é importante compreender um dos elementos mais relevantes em relação ao CSS como as unidades de medidas. Aqui será especificado às quatro unidades principais segundo Zemel (2014):

- Ems (*em*): O *em* representa uma unidade escalável. A unidade 1em representa o tamanho real do elemento pai. Por exemplo, uma fonte 12pt representa 1em. O *em* é escalável, assim 2em seria igual 24pt ou 0.75em igual a 8pt.
- Pixel (px): Esta é a unidade fixa mais utilizada no CSS. O pixel é um ponto indivisível para a tela de exibição. Os desenvolvedores costumam utilizar esta unidade pelo fato de ser considerada de *pixel perfect*, sendo assim, um layout pode ser representado identicamente ao layout do projeto.
- Ponto (point): Geralmente utilizados em CSS de impressão. Um ponto representa 1/72 polegadas e assim como pixel também são indivisíveis.
- Porcentagem (%): Esta unidade é muito parecido com em porém existem algumas diferenças como, 12pt representa 100% na unidade de porcentagem. Assim, em dispositivos móveis o texto consegue ser escalável.

Depois da descrição destes elementos de mediadas em CSS é preciso observar os dois elementos, *em* e porcentagem que possuem características muito parecidas. Assim segundo a explicação de Zemel (2014, p. 25):

[...] apesar de ser possível usar qualquer um dos tipo de medidas relativas, existe uma espécie de consenso entre os desenvolvedores[...]: **usar porcentagem para lidar com tamanhos no layout** (larguras, margens, espaçamentos, etc) e **usar ems para lidar com fontes**. em pode até ser usado fora de textos, mas vai ser sempre uma **medida relativa ao font-size**; já o *porcento* é relativo ao *font-size* quando usada em *font-size*, mas, quando usada com outras mediadas, é **relativa à largura do elemento pai**.

A compreensão destas unidades de medidas é importante para determinar em quais situações devem ser usadas. Além disto, a medida tem que levar em conta o objetivo do contexto em que será aplicada, de forma que a visualização do site tenha uma boa usabilidade.

2.3.3 Imagens e recursos flexíveis

As imagens e recursos flexíveis é uma maneira de tratar estes tipos de mídias de forma que não fiquem distorcidas à medida que as resoluções de telas vão se alternando. Esta técnica de desenvolvimento tem que levar em conta qual é o melhor tipo de extensão de mídia. Segundo Zemel (2012), precisa-se encontrar uma solução para o problema de representação com imagens usando recursos flexíveis.

Existem ferramentas como Rloadr que oferece controle de como as imagens vão se comportar e, com controle de carregamento sob demanda além de não fazer múltiplas requisições para a mesma imagem, também dando suporte aos navegadores com o JavaScript desabilitado. Outro exemplo de ferramenta é o jQuery Picture que é um plugin do jQuery que auxilia no suporte de imagens responsivas nos layouts, igualmente, dá suporte para os elementos `figure` e `picture`. (ZEMEL, 2014). Este tratamento cria extensões necessárias para que as imagens não fiquem distorcidas à medida que as resoluções de telas vão diminuindo.

2.4 UML

Para o desenvolvimento de um sistema orientado a objetos, é preciso fazer uma representação da aplicação, descrevendo através de diagramas, como cada etapa que vai compor este sistema com base no levantamento de requisitos. Assim, é possível ter uma visão de toda aplicação pode-se determinar cada componente antes de começar a programar. Portanto, a *Unified Modeling Language* (UML) é o projeto base de um software.

A união dos três métodos de modelagens: *Object-Oriented Software Engineering* (OOSE), de Rumbaugh, método Booch e o método *Object Modeling Technique* (OMT), fez com que surgisse a UML. Para a união destes três métodos, foi fundamental o apoio e financiamento da Rational Software. Sendo até a década de 1990, os métodos mais conhecidos e utilizados pelos desenvolvedores nas modelagens de softwares orientados a objetos. O projeto se iniciou com a união do método OMT com o Booch, sendo lançado em 1995 o Método Unificado. Em seguida ao lançamento deste método, Jacobson (método OOSE), se juntou aos outros dois métodos na Rational Software, sendo incorporado na nova metodologia. Assim, em 1996, surgiu a primeira versão da UML. Já em 1997, após o lançamento da primeira versão da UML, empresas atuante na modelagem e desenvolvimento de softwares, começaram a contribuir no projeto com sugestões, propondo melhorias na linguagem. A *Object Management Group* (OMG), adotou em 1997, a UML como linguagem padrão de modelagem. (GUEDES, 2011)

2.5 DIAGRAMAS DE CASO DE USO

O diagrama de caso de uso representa por meio de uma linguagem simples, o comportamento externo do sistema: destacando suas principais funcionalidades e, a interação com o usuário. Na UML, o diagrama de caso de uso é considerado o mais informal, pela forma abstrata como é representado. É o diagrama usado no início da modelagem do sistema, junto com as análises e levantamento de requisitos. Pode ser modificado e alterado no decorrer do desenvolvimento do processo. (GUEDES, 2011)

No diagrama de caso de uso está representando os requisitos do sistema referindo-se, as tarefas ou funcionalidades que foram identificados dentro da análise de requisitos que podem ser necessários no desenvolvimento do software (GUEDES, 2011). Desta forma, os atores, que são os agentes que vão interagir com o sistema, são utilizados para definir qual o seu papel na utilização do sistema. Da mesma maneira, a documentação descreve todas as etapas necessárias a serem desenvolvidos no sistema como, as validações e restrições tanto quanto para o usuário e sistema.

Sua representação é uma elipse que contém um texto expressando qual a sua finalidade e a qual caso de uso se refere. De acordo com Guedes (2011), “[...] não existe um limite determinado para o texto que possa estar contido dentro da elipse, mas em geral, a descrição de um caso de uso costuma ser bastante sucinta.”. (p. 54)

2.6 DIAGRAMA DE CLASSE

O diagrama de classes é um dos principais diagramas para a visualização das classes que irão compor o projeto. Na sua estrutura é definido a interação entre as classes, seus atributos e métodos, transmitindo informações de comportamento e organização. Neste diagrama, é representado de forma estática, a relação entre as classes. O diagrama de classes pode servir ainda, para a elaboração de outros diagramas da UML. (GUEDES, 2011, p. 101)

De acordo com Booch, Rumbaugh e Jacobson (2012):

Os diagramas de classes também são a base para um par de diagramas relacionados: os diagramas de componentes e os diagramas de implantação. Os diagramas de classes são importantes não só para a visualização, a especificação e a documentação de modelos estruturais, mas também para a construção de sistema executáveis por intermédio de engenharia direta e reversa. (p. 115)

2.7 HTML5

A linguagem de marcação HTML5 teve seu início em 2004, por conta da decisão da *World Wide Web Consortium* (W3C) em encerrar o desenvolvimento do HTML em favor do *eXtensible Hypertext Markup Language* (XHTML). Posteriormente, a decisão foi revogada para a retomada dos estudos no desenvolvimento do HTML5, tendo como base, o que já vinha sendo trabalhado pela *Web Hypertext Application Technology Working Group* (WHATWG). O desenvolvimento do HTML5 é feito de forma conjunta entre a W3C e a WHATWG onde, ambas as partes, matem em seus sites as especificações da linguagem com alguns detalhes diferentes. A WHATWG é menos restritiva do que a W3C como por exemplo: nas especificações com ilustrações e informações de suporte e funcionalidades que não constam na versão apresentada pela W3C. (Silva, 2011)

Conforme Silva (2014), o HTML5 é uma tecnologia nova e, ao contrário das versões anteriores do HTML, está sendo desenvolvida em ritmo crescente com engajamento das comunidades de desenvolvedores e empresas de desenvolvimento de softwares que estão utilizando as funcionalidades da linguagem em suas aplicações.

O HTML5 tem como principal objetivo a facilitação na manipulação dos elementos possibilitando, a modificação dos objetos de forma não intrusiva tornando assim transparente ao usuário final. Outro ponto importante, é a fornecimento de ferramentas do HTML5 para o CSS e o JavaScript poderem fazer da maneira melhor possível o seu trabalho. As *Application Programming Interface* (APIs) da linguagem, permitem a manipulação das características destes elementos, mas tornam as aplicações ou websites leves e funcionais. O HTML5 também criou novas tags e modificou outras nas seções comuns e específicas como a criação de rodapés, cabeçalhos, *slidebar*, menus entre outros. (FERREIRA, EIS)

Ainda segundo Ferreira e Eis:

O HTML5 modifica a forma como escrevemos código e organizamos a informações na página. Seria mais semântica com menos código. Seria mais interatividade sem a necessidade de instalação de plugins e perda de performance. É a criação de código Inter operável, pronto para futuros dispositivos[...] (p. 10)

A nova versão do HTML trouxe mudanças significativas no desenvolvimento de conteúdos para web. As principais características são a possibilidade de desenvolvimento de

páginas dinâmicas e leves e a inserção de novos atributos que não possuíam na versão anterior ao HTML5.

2.7.1 Os novos atributos

O HTML5 introduziu novos elementos na marcação com o objetivo de oferecer opções melhores na estruturação do documento. Conforme Silva (2011), os desenvolvedores podem contar com os seguintes atributos:

- **article**: é o atributo de marcação autossuficiente em uma página, documento ou aplicação. A marcação do conteúdo por este atributo é, reusável podendo de forma independente ser reusável como por exemplo: via RSS. Os conteúdos que podem ser marcados pelo atributo são: post de fóruns, artigos de revista ou jornal, matéria que é publicada em blog, comentário feito por visitante e, ainda, *widget* ou *gadget* interativo;
- **aside**: destinado a marcação de conteúdo separado, mas que esteja relacionado com outros conteúdos a sua volta. Nas tipografias, são os conteúdos em blocos que ficam nas barras laterais. Os exemplos de conteúdo a serem marcados pelo atributo são: propagandas em textos ou banners, barras laterais e, conteúdos que possam ser considerados separados do conteúdo principal;
- **audio**: é o elemento utilizado para incorporar som ou *stream* de *audio*;
- **canvas**: destinado para a criação dinâmica de imagens como: gráficos estáticos ou jogos e gráficos dinâmicos;
- **command**: este é um atributo que se destina ser evocado pelo usuário. Pode ser um comando isolado na página ou ainda, fazer parte de um menu de contexto ou barra de ferramentas criadas pelo elemento menu;
- **menu**: é destinado a marcar uma lista de comandos utilizados com elemento container para o atributo *command*;
- **datalist**: o elemento foi criado com a finalidade de ser utilizado em conjunto com o elemento input. Assim, o elemento destina-se a fornecer uma lista de sugestões no preenchimento do campo input permitido ao usuário, a seleção das opções definidas na lista;
- **details**: o elemento fornece ao usuário informações textuais ou formulários adicionais sobre determinada página ou conteúdo que se seja específico da página em que se

encontra. Como padrão, o conteúdo fornecido por este elemento não fica à vista do usuário, mas, é mostrado mediante a ação do usuário através do clique sobre o elemento;

- **summary**: trata-se do fornecimento de rótulos ou sumários com informações adicionais sobre o *details*;
- **figure**: este elemento pode ser utilizados como marcações de ilustrações, diagramas, imagens, gráficos, fotos, vídeos, linhas de códigos ou similares. Destinado a servir containers de conteúdo independente ao fluxo do documento principal;
- **figcaption**: é destinado para o uso de legenda do elemento marcado como figure;
- **footer**: usado para fazer a marcação do rodapé da página ou sessão de um documento com as informações sobre os mesmos;
- **hgroup**: este elemento é destinado a agrupação dos elementos h1-h6 quando existir mais de um destes elementos em uma mesma página ou sessão;
- **keygen**: utilizado na manipulação de um par de chaves criptografadas, públicas ou privadas para a comunicação através do envio de formulários ao servidor;
- **mark**: é destinado a marcação de palavras em que se deseja dar ênfase dentro de um texto servindo como referência;
- **meter**: o elemento é destinado a marcação de mediadas escaláveis compreendidas por limites conhecidos;
- **nav**: este elemento é destinado a marcação de uma seção da página que contenha links de navegação para outras páginas ou, para outras partes dentro da própria página;
- **Output**: usado para representar um cálculo dentro de um contêiner;
- **progress**: mostra o progresso de uma tarefa marcada pelo elemento de forma indeterminada (quando não há uma medida indicando quanto falta para realização de uma determinada tarefa) e, determinada (quando pode mostrar em porcentagem, o progresso da realização da tarefa);
- **ruby**: utilizado na marcação ou anotações sobre o texto, com informações corretas de pronúncias. Trata-se de elemento utilizado na escrita oriental;
- **section**: destina-se a marcação de uma seção da página;
- **source**: podem ser especificados vários arquivos utilizando este elemento para disponibilizar video ou audio que foram codificados para vários tipos de navegadores em formatos diferentes;
- **track**: define as trilhas externas dos elementos audio e vídeo podendo ser personalizado as trilhas gravadas nos arquivos externos e posteriormente, inseridos no vídeo;

- **video**: é o elemento destinado para incorporação de filmes ou vídeos.

2.8 CSS3

O CSS3 representa uma evolução do CSS que, assim como o HTML5, teve algumas mudanças significativas. Segundo Castro e Hyslop:

Nos últimos anos, um dos desafios enfrentados pelos autores de sites foi o número limitado de opções para produzir layouts elaborados utilizando o CSS. Na maioria dos casos, isso significava usar HTML e CSS adicionais e muitas imagens. Como resultado, as páginas ficaram mais complicadas, menos acessíveis, levavam mais tempo para abrir e eram simplesmente mais frágeis difíceis de manter. (2013, p. 371)

Isto significa que, com a nova versão da linguagem, com a adição de novas propriedades, os desenvolvedores contam com recursos para a criação de efeitos impossível na versão anterior, resultando em criação de conteúdos mais leves. Apesar do CSS3 contar com recursos adicionais como, o uso de gradientes através de códigos, não significa que sua utilização é fácil. Isso se deve ao fato de que muitos navegadores não dão suporte para todas as propriedades do CSS3. (CASTRO e HYSLOP, 2013)

A sintaxe de utilização do CSS é composta por: seletor e propriedade. O seletor representa a estrutura, determinando a formatação do elemento. Os seletores podem ser encadeados, agrupados ou complexos. Já a propriedade, determina as características dos seletores. Na figura 4, é demonstrado os dois tipos de seleção mais simples utilizados no CSS, desconsiderando o tipo de seleção complexa⁵:

Figura 4 - exemplos de seletores

Lado A: seletor encadeado	Lado B: seletor agrupado
<pre>seletor 1 seletor 2 seletor 3 { propriedade: valor; }</pre>	<pre>seletor 1, seletor 2, seletor 3 { propriedade: valor; }</pre>

Fonte: Do Autor

- Lado A: este seletor está informando a formatação do *seletor 1* que está dentro do *seletor 2* que, por sua vez, encontra-se dentro do *seletor 3*;

⁵ Lista de tipos de seleção complexa atualizada poderá ser vista no link: <http://www.w3.org/TR/css3-selectors/#selectors>

- Lado B: neste exemplo, é selecionado três tipos de seletores e separados por vírgula para que herdem a mesma formatação de valores.

O CSS3 vem com muitas mudanças significativas nas propriedades. Uma delas, é a opção de bordas arredondadas, como mostra a figura 5:

Figura 5 - Exemplo de borda arredondada



Fonte: Do Autor

Este efeito é conseguido através da propriedade `border-radius` onde, pode ser determinado o valor em pixel a ser aplicado no seletor. Criar efeitos de sombras em textos e elementos também ficaram mais práticos. As propriedades `text-shadow` e `box-shadow` criam os efeitos em textos e bordas respectivamente. A figura 6 mostra um exemplo prático das propriedades `text-shadow` e `box-shadow`:

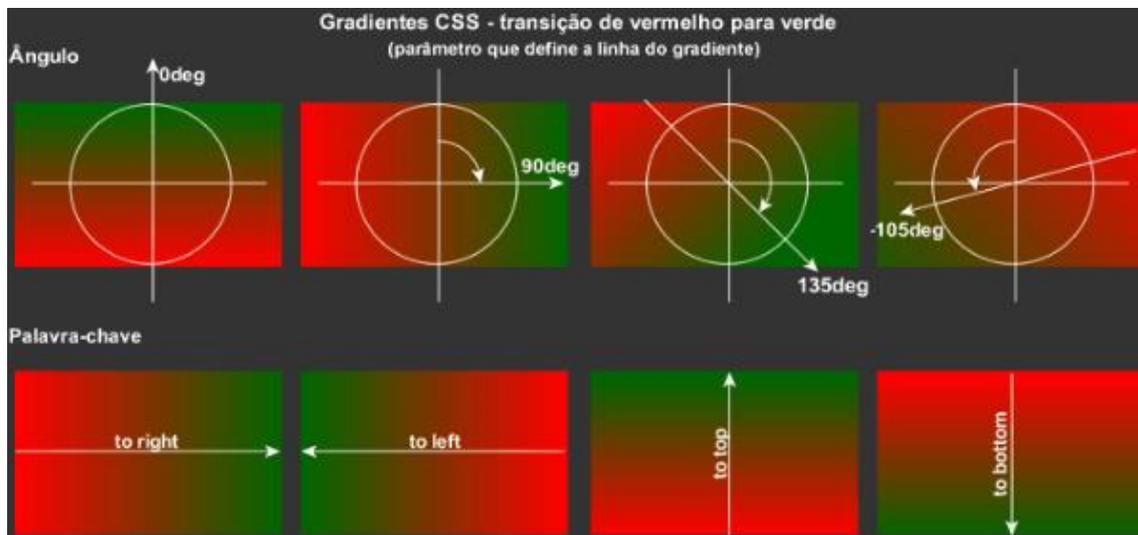
Figura 6 - Exemplo de efeito de sombra em texto e elemento



Fonte: Do Autor

Outro recurso interessante no CSS3 é a criação do efeito de opacidade em seletores ou imagens com a propriedade `opacity`. Para o mesmo efeito no IE⁶ 8 ou anteriores, deve-se utilizar a propriedade `filter:alpha(opacity = valor da opacidade que deseja)`. O efeito perspectiva, é definido pela propriedade `perspective` onde, o valor de profundidade é declarado em pixel, quanto menor for o valor, maior será a profundidade atribuída ao seletor. Esta nova versão do CSS, traz o uso de gradientes que, é atribuído pela propriedade `linear-gradient` podendo variar de acordo com a combinação das palavras chaves como mostra a figura 7

Figura 7 - Exemplo de efeitos obtidos pelo uso das palavras chaves



Fonte: SILVA (2014).

A criação de animações é um recurso disponível pela propriedade `animation`, permitindo que um elemento mude para diferentes estilos. A propriedade `keyframes` assegura os estilos que um elemento pode assumir em determinados momentos. As propriedades de animações do CSS3 são listadas na tabela 1, como pode ser visto logo abaixo:

Tabela 1 - Lista de regras de keyframes e todas as propriedades de animações

Propriedade	Descrição
@keyframes	Especifica o código da animação
Animation	Um atalho para definir todas as propriedades de animação (exceto animação-play-state e-

⁶ IE é a abreviação de Internet Explorer desenvolvido pela Microsoft.

	fill-modo de animação)
animation-delay	Especifica um atraso para o início de uma animação
animation-direction	Especifica se uma animação deve desempenhar no sentido inverso ou ciclos alternados
animation-duration	Especifica quantos segundos ou milissegundos uma animação leva para completar um ciclo
animation-fill-mode	Especifica um estilo para o elemento quando a animação não está jogando (quando ele for concluído, ou quando se tem um atraso)
animation-iteration-count	Especifica o número de vezes que uma animação deve ser jogada
animation-name	Especifica o nome da animação keyframes
animation-play-state	Especifica se a animação está em execução ou em pausa
animation-timing-function	Especifica a curva velocidade da animação

Fonte: http://www.w3schools.com/css/css3_animations.asp

Independência da posição de estrutura com relação ao código HTML e uma das características mais importantes do CSS3. Conhecido como *Object-Oriented Cascading Style Sheets* (OOCSS), faz a separação da estrutura e do visual e a independência do container em relação ao conteúdo. (EIS, 2011)

De acordo com Eis (2011), a separação da estrutura e do visual se deve porque:

A maioria dos elementos estilizados em uma página web tem diferentes características visuais que são repetidas em diferentes contextos e situações. Algumas características são fáceis de identificar como cores, títulos, gradientes, bordas etc. Essas são características visuais. Contudo, existem também as características de estrutura, que é onde nós “montamos” os elementos, definindo tamanhos, distâncias, medidas etc. Essas características também são repetidas em diversos elementos no decorrer do site.

A autonomia do container em relação ao conteúdo, possibilita por exemplo, aplicação de propriedades diferentes no mesmo seletor, sem que este afete os demais, tornando aplicação independentes sem a necessidade de duplicação de códigos.

Contudo isto, é preciso destacar que, muitos destes novos recursos como o uso de animações, não são suportadas por algumas versões dos navegadores. Mesmo assim, a capacidade de produção de aplicações com alta qualidade é notória.

2.8.1 Media queries

“Media Queries é a utilização de Media Types como uma ou mais expressões envolvendo características de uma mídia para definir formatações para diversos dispositivos.” (EIS, 2011)

Com a utilização da media queries é possível capturar a resolução da tela do dispositivo, direcionando conteúdos que melhor se adaptam a ele. Outras características da media queries é definir o tamanho máximo da página web, entre outros recursos.

Este é um dos pontos fortes para se trabalhar com web design responsivo. Existem diversos recursos úteis na utilização da media queries que podem ajudar no desenvolvimento do site. Dessa forma, pode-se utilizar estes recursos como, por exemplo, para pegar as características do dispositivo e melhorar a forma como o conteúdo será disposto para o usuário.

2.9 JAVASCRIPT

O JavaScript é uma linguagem de programação criada por Brendan Eich, em 1995. É baseada em script do lado cliente e dinâmica orientada a objetos com a sintaxe semelhante à linguagem C. Segundo Flanagan (2013), JavaScript é uma linguagem de programação Web. O autor ainda define que:

A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos – em computadores de mesa, consoles de jogos, tablets e smartphones – incluem interpretadores de JavaScript, tornando-a a linguagem de programação mais onipresente da história. (FLANAGAN, 2013, p. 1)

Apesar da semelhança no nome com o Java, são duas linguagens opostas. O JavaScript usa alguns métodos do Java, mas o seu nome se deve a uma jogada de marketing para a popularização da linguagem. Também emprega recursos que não existem em outras linguagens orientadas a objetos como a linguagem C++ e Java. O JavaScript possui scripts que facilitam a manipulação do *Document Object Model* (DOM).

O DOM é uma API utilizada para manipular os conteúdos de documentos como: HTML e *eXtensible Markup Language* (XML). Os elementos HTML ou XML, são representados em um DOM como uma árvore de objetos. As marcações da linguagem HTML, representam os nós da árvore contendo *strings* de textos e também os comentários a ser considerados nós para manipulação. (Flanagan, 2013)

Para uma ilustração, observe um código simples em HTML como mostra a figura 8 abaixo:

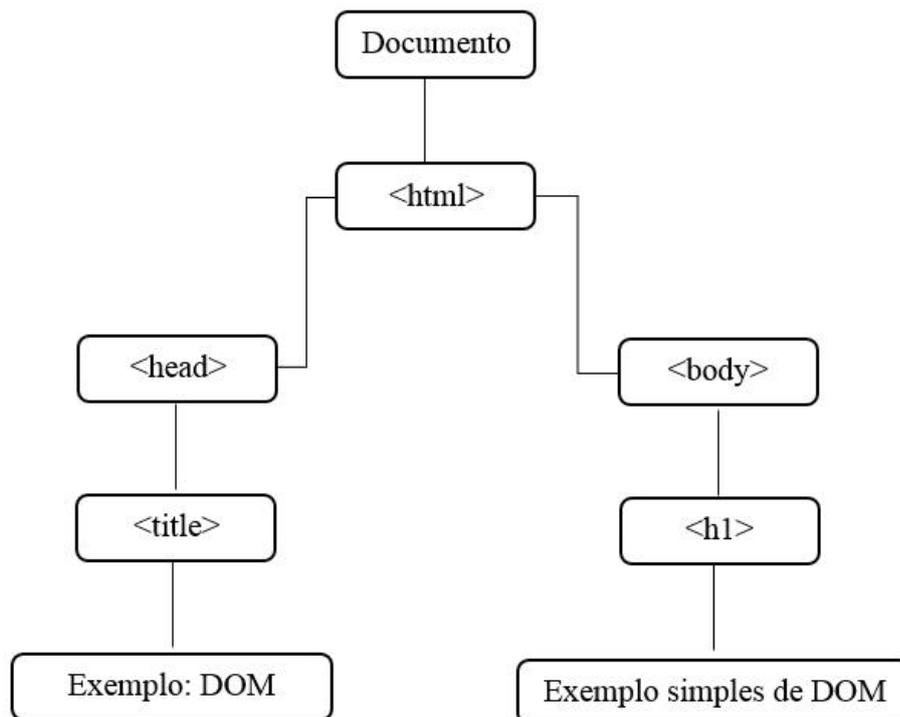
Figura 8 - Exemplo de código em HTML

```
<html>
  <head>
    <title>Exemplo: DOM</title>
  </head>
  <body>
    <h1> Exemplo simples de DOM </h1>
  </body>
</html>
```

Fonte: Do Autor

Seguindo o exemplo da figura 8 anterior, a representação do DOM deste documento ficaria como demonstra a figura 9:

Figura 9 - Representação em árvore do DOM do documento da figura 8



Fonte: Do Autor

A figura acima representa a árvore do arquivo HTML onde, existem três tipos de nós diferentes. O Documento é a raiz do arquivo portanto, em JavaScript é conhecido pela subclasse *Node* como *Document*. Os atributos em HTML representam o *Element* do arquivo e o texto é o elemento *Text*. Os elementos *Document* e *Text* são as principais subclasses de manipulação do DOM.

Segundo Flanagan (2013), a manipulação da maioria dos programas em JavaScript no lado cliente, é feito através de um ou mais elementos de documento, assim, na execução do programa, pode-se utilizar de variável global *document* fazendo referência ao objeto *Document*. Mas, para a manipulação deste documento, é preciso selecionar os objetos *Element* que referenciam os elementos de documento. Ainda conforme Flanagan (2013, p. 354), o DOM define várias maneiras para a manipulação dos elementos como:

- **Fazendo a seleção dos elementos pela identificação:** neste caso, qualquer elemento em HTML pode ter um atributo de identificação único: *id*. O valor de referência não pode se repetir dentro do documento. A seleção do elemento é utilizado pelo uso do método: `getElementById()`;
- **Selecionando os elementos pelo nome:** ao contrário do elemento *id*, *name* é um elemento utilizado em HTML para a identificação em formulários quando estes são enviados para o servidor, não precisando ser único dentro do documento, uma vez que, é muito utilizado em nomes de botões e também, em caixas de seleção dentro do formulário. Além disso, o atributo *name* é valido somente em alguns elementos do HTML como: formulários, elementos de formulários e imagens. Para a seleção dos elementos em HTML como base o atributo *name*, é utilizado do objeto *Document* o método: `getElementsByName()`;
- **Fazendo a seleção pelo tipo do elemento:** Todos os elementos de um tipo específico de marcação em HTML ou XML, podem ser selecionados pelo uso do método `getElementsByName()` do objeto *Document*. A seleção por um tipo específico pode ser constituído da seguinte maneira: `getElementsByTagName("span")`. Neste exemplo, está selecionando todos os elementos *span* dentro documento. Desta forma, o método `getElementByTagName()` assim como `getElementsByName()`, retornam um objeto *NodeList*⁷;
- **Selecionando os elementos com a classe do CSS específica:** este tipo de seleção é feito pelo atributo *class* de um HTML que possui uma lista separada de zero ou mais

⁷ *NodeList* são objetos do tipo de retorno dos métodos `getElementsByName()` e `getElementsTagName()`, semelhantes a um *array* somente para leitura.

identificadores por espaço. Assim, o atributo pode descrever a definição dos conjuntos de elementos que são relacionados no documento onde, todos os elementos que possuem o mesmo identificador no seu atributo pertencem ao mesmo conjunto. O *class* é uma palavra reservada do JavaScript e, no lado cliente é utilizado pela palavra *className* contendo o atributo do HTML *class*. O método para seleção é o `getElementsByClassName()`;

- **Fazendo a seleção dos elementos com seletores CSS:** os seletores do CSS tem uma estrutura de sintaxe de seleção muito bem estruturados que, descrevem bem os tipos de elementos ou conjuntos, com intenção de manipulação dentro do documento. Os elementos podem ser descritos pelos nomes (no caso as tags), grupos (onde pode ser definido pelo atributo *class*) e, também pelo identificador único: *id*. A W3C fez a padronização dos seletores para o CSS3, descrevendo sua API de seletores específicos e, no JavaScript, obtêm-se os elementos que coincidem com determinados seletores. O método `querySelectorAll()` do objeto *Document*, recebe argumento em *strings* que contém o seletor CSS e, retorna um objeto do tipo *NodeList* que possui todos os elementos correspondentes ao seletor.

2.9.1 JQuery

O JQuery é uma biblioteca JavaScript de desenvolvimento rápido criada por John Resig. Está disponibilizado como software livre e de código aberto sob a licença do *Massachusetts Institute of Technology* (MIT) e *General Public License* (GPL). Assim, “isto significa que pode ser utilizada gratuitamente tanto para desenvolver projetos pessoais como comerciais.” (SILVA, 2010, p. 23)

Antes da criação do primeiro plug-in do JQuery em 2006, era necessário escrever muitas linhas de códigos em JavaScript para se obter simples efeitos como: mascaras em campos de formulários (CPF/CNPJ, data, valores, entre outros), utilização do AJAX junto com JQuery, etc. Agora, com poucas linhas e a inclusão da biblioteca no projeto em desenvolvimento, obtêm-se os efeitos e se mantém uma organização mais limpa no código fonte. Praticidade e simplicidade são duas palavras que podem resumir bem esta biblioteca.

JQuery serve para dar dinamismo nas páginas web com mais interatividades, com funcionalidades que visam incrementar de forma simples e objetiva os designs que seguem os conceitos de usabilidade e acessibilidade. Ainda descrevendo a funcionalidade do JQuery, na Opinião de Silva (2010), o framework pode ser usado nas páginas web para:

- Ter interatividade;
- Usar animações e efeitos visuais;
- Fazer requisições no servidor sem a necessidade de recarregar a página;
- Manipulação do DOM através do acesso;
- Simplificação de tarefas com uso de JavaScript;
- Alteração de conteúdo;
- Modificação das apresentações e estilizações das páginas web.

Desta forma, é possível fazer acesso, manipulação do DOM e adicionar animações de forma fácil, de maneira que designer e desenvolvedores com pouca experiência conseguiram criar aplicações de maneira fácil e descomplicada.

A principal característica do JQuery é a interação no comportamento da página web. Assim, “[...]é lícito concluir que o princípio de funcionamento do JQuery fundamenta-se em sua capacidade de encontrar os elementos de HTML que constituem a página e a estes anexar seus métodos.” (SILVA, 2010, p. 41)

2.9.1.1 JQuery mobile

John Resig o responsável pela criação da biblioteca do JavaScript conhecida como JQuery, anunciou em 2011, que estava desenvolvendo um framework junto com uma equipe, baseado nas bibliotecas de JQuery e JQuery UI para dispositivos moveis. Naquela ocasião destacou que empresas como a Mozilla, Palm Inc., Blackberry, Nokia, Adobe, grupo Filament e a DeviceAtlas estavam apoiando o projeto em desenvolvimento. (SILVA, 2013)

Na mesma ocasião comunicou que este framework teria uma interface unificada e que assim poderia servir os navegadores dos smartphones mais usados como uma das principais estratégias de serviço do framework. Assim em 16 de outubro de 2010, foi lançado a primeira versão do framework que viria a ser conhecido como JQuery mobile versão 1.0 Alfa.

Mas o grande destaque deste trabalho é as versões a partir da 1.3 do JQuery mobile que começou a focar no desenvolvimento do web design responsivo. Até a data da elaboração deste projeto a última versão do JQuery mobile é a 1.4. Este é um framework desenvolvido especificamente para o acesso de qualquer plataforma como desktops, tablets e smartphones, ou seja, uma aplicação acessível independentemente do dispositivo.

Assim o desenvolvimento com JQuery mobile está voltado para o fornecimento de:

[...]mecanismos capazes de criar sistemas unificados de interface de usuário, baseados em HTML5 e CSS3 e capazes de ser funcionais em todas as plataformas móveis tendo como base de construção as bibliotecas jQuery e jQuery UI. Tais sistemas não apenas incluem funcionalidades para a criação do layout (listas, painéis, layers etc), mas também disponibilizam um rico conjunto de controles e widgets (sliders, toggles, abas etc.). (SILVA, 2013, p. 20)

Outro ponto interessante sobre o JQuery mobile é a possibilidade de personalização do framework de acordo com as necessidades de cada projeto onde, posteriormente, pode ser baixado o arquivo personalizado. Conforme Silva (2013), as principais características que se destacam do JQuery mobile são:

- O seu desenvolvimento teve como base a biblioteca JQuery sendo assim, incorporados as funcionalidades e mecanismos da JQuery UI. O aprendizado do framework é rápido por ter sua sintaxe baseada na biblioteca JQuery, não exigindo uma programação sofisticada que, é a característica do JQuery;
- Tem compatibilidade com a maioria dos dispositivos moveis: tablets, e-readers e, também das plataformas desktops e todos os navegadores modernos;
- Mínima dependência de imagens o que proporciona carregamento rápido devido ao tamanho reduzido;
- Possui uma arquitetura modular permitindo a personalização de acordo com a seleção dos componentes a serem utilizados no projeto;
- Desenvolvimento ágil com a mínima dependência de scripts resultado da marcação do HTML5 na estrutura e inserção de comportamentos da página;
- Desenvolvimento de aplicações ricas com a máximo de funcionalidades avançadas para os dispositivos modernos com utilização de técnicas que, não deixam bloquear dispositivos mais antigos;
- Escalonamento automático de interfaces das aplicações com desenvolvimento de técnicas para servir dispositivos como: smartphones ou tablets e, também, desktops;
- Uso de navegação em AJAX que permitem fazer animações nas transições de páginas sem o comprometimento dos botões de retrocesso e, o armazenamento dos favoritos e o controle de URLs;
- Utilização de APIs simples para dar suporte à interação ao toque, em eventos do mouse e do foco;

- Widgets padronizados para a otimização e interação da interface de usuário, sendo sensível ao toque, estilizável e fáceis de usar além de serem multiplataforma;
- Disponibilização de framework nas aplicações de temas, ThemeRoller⁸, fácil de usar e, personalizável;
- Personalização dos scripts do framework com a utilização de APIs próprias.

2.10 PHP

O PHP é acrônimo para PHP: *Hypertext Preprocessor* que surgiu como uma linguagem representada por um conjunto de *scripts* executados pela linguagem Perl⁹, desenvolvido em 1994 por Rasmus Lerdof. O PHP faz referência ao conjunto de scripts onde, mais tarde, integraria com a ferramenta *Forms Interpreter* (PHP/FI), que é uma ferramenta de análise sintática para consultas em *Structured Query Language* (SQL). De modo geral, o PHP é definido como uma linguagem de criação de scripts junto com HTML que é executada pelo servidor. (RAMOS, SILVA e AFONSO, 2007)

Conforme Niderauder (2011), diferentemente das linguagens como JavaScript e HTML5, o PHP trabalha juntamente ao servidor fazendo o processamento das páginas dinâmicas de acordo com as requisições do cliente. Por possuir código aberto, acabou se popularizando e sendo utilizado por milhões de websites. Outra característica é a sua portabilidade e facilidade de integração com diferentes tipos de bases de dados.

2.10.1 Funcionamento do PHP

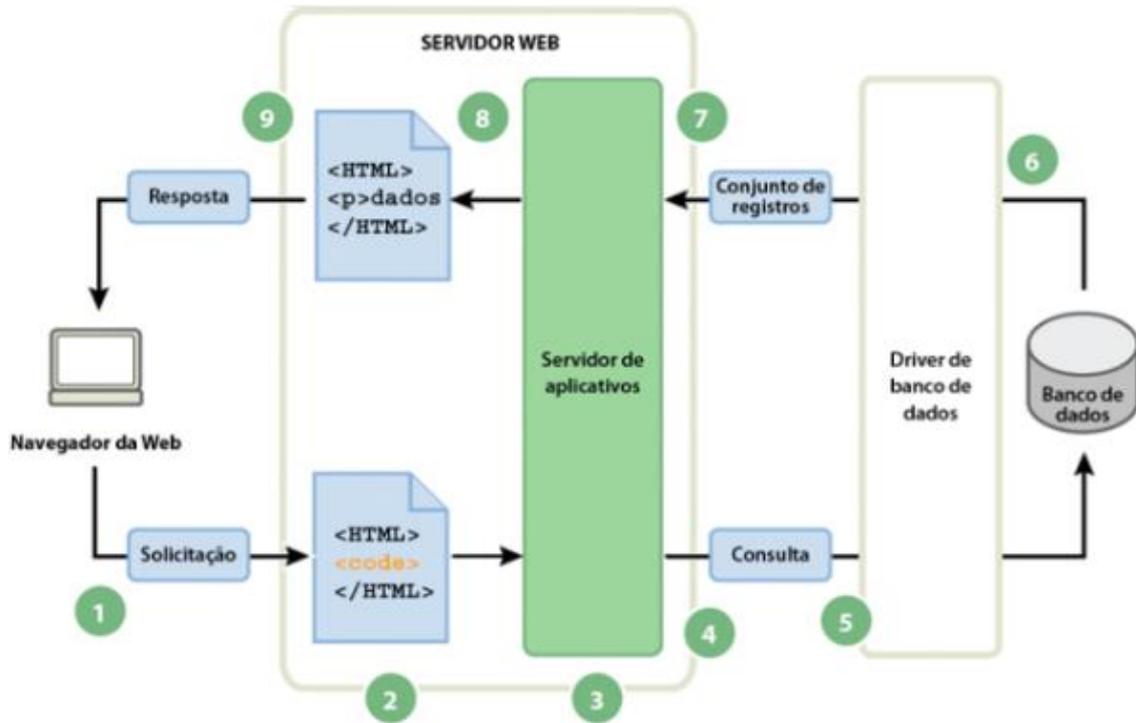
O PHP é uma linguagem de programação server-side (lado servidor), muito utilizado em páginas web por oferecer interatividade entre o usuário e a aplicação, tornando às páginas dinâmicas. Seu código fica embutido junto a linguagem de marcação do HTML. As requisições são submetidas através da *Uniform Resource Locator* (URL) onde, vão para o servidor que se encarrega de interpretar o código em PHP que faz a montagem da página em HTML devolvendo para o navegador. Através destas requisições também é possível fazer consultas em base de dados.

⁸ ThemeRoller é um aplicativo do jQuery onde é possível criar os próprios temas do seu aplicativo.

⁹ Perl é uma linguagem de programação que existe a mais de 27 anos.

Para ilustração do modelo funcionamento do PHP, segue abaixo, a figura 10 ilustrando as etapas do processo de requisição e resposta dada quando o usuário faz uma interação com a página de acordo com Malcher (2014):

Figura 10 - Funcionamento do PHP



Fonte: Malcher (2014)

- 1: Através do navegador web e feito a solicitação a página dinâmica;
- 2: O servidor web faz a localização passando a página para o servidor de aplicativo;
- 3: Depois que o servidor de aplicativo recebeu a página, começa a ser feito o rastreamento na busca por instruções;
- 4: Feito o rastreamento, o servidor de aplicativo faz o envio para o driver do banco de dados;
- 5: O driver faz a consulta no banco de dados;
- 6: Conforme a consulta, a base de dados devolve a resposta ao driver;
- 7: Depois de receber a resposta, o driver passa para o servidor de aplicativos;
- 8: O servidor de aplicativo pega os dados recebidos e insere na página e, repassa para o servidor web;
- 9: Por fim, o servidor web devolve a página montada para o navegador que fez a solicitação.

2.10.2 Capacidade do PHP

Conforme Welling e Thomson (2005), o PHP tem muitas vantagens em relação aos seus concorrentes e, entre estas vantagens estão:

- **Desempenho:** com um servidor de baixo custo, é possível atender muitos acessos por dia;
- **Integração de banco de dados:** o PHP possui nativamente muitos drivers disponíveis para a maioria das bases de dados mais utilizadas;
- **Bibliotecas integradas:** como o PHP foi desenvolvido voltado atender as aplicações web, ele já conta com muitas funcionalidades integradas como: geração de documentos em *Portable document format* (PDF), criação de *Graphics Interchange Format* (GIF) instantâneos, fazer conexão a outros serviços de redes, enviar e-mails, trabalhar com cookies, entre outras funções com poucas linhas de códigos;
- **Baixo custo:** o PHP é gratuito e pode ser feito o download da última versão no site <http://www.php.net>;
- **Facilidade de aprendizagem e utilização:** por ter uma sintaxe baseada nas linguagens de programação principalmente C e Perl, qualquer desenvolvedor assimilara sua sintaxe conhecendo estas ou, que tenham experiência em outras linguagens como: Java, C++, entre outras;
- **Suporte a orientação a objetos:** a partir da versão 5 o PHP apresentou recursos bem elaborados quanto a orientação de objetos. Algumas funcionalidades já se encontravam disponíveis nas versões anteriores 3 e 4, porém, a versão 5 tem um suporte orientados a objetos mais completo;
- **Portabilidade:** o PHP encontra-se disponível para a maioria dos sistemas operacionais existentes. E um “código bem escrito normalmente funcionará sem modificação em um sistema diferente executando o PHP.” (WELLING, THOMSON, 2005, p. XXIX);
- **Disponibilidade de código fonte:** o código fonte é aberto caso se deseje fazer alguma modificação ou adicionar a linguagem;
- **Disponibilidade de suporte:** atualmente o PHP é mantido e financiado pela *Zend Technologies* e, oferece suporte e softwares relacionados numa base comercial.

2.11 SQL: INTRODUÇÃO AO BANCO DE DADOS RELACIONAL

A SQL é uma estrutura padrão de acesso aos dados em um sistema gerenciador de banco de dados relacional. Foi criada pela IBM no início dos anos 80. A *American Nacional*

Standards Institute (ANSI), iniciou em 1980 o trabalho para a padronização de uma linguagem de banco de dados. Foi publicado nos anos de 1986 e 1987, a primeira padronização do SQL pela ANSI e a *International Standards Organization* (ISO) e, depois, juntaram esforços para a padronização de um conjunto maior do SQL, criando o popularmente conhecido SQL ANSI-92 ou ainda SQL/92/SQL2. Existe um grupo de trabalho tentando criar um novo padrão do SQL2 para que, incorpore as características de orientação a objeto estendendo para o SQL3. (NETO, 2007)

De acordo com Neto (2007), a maioria dos sistemas de gerenciamento de banco de dados (SGBDs), implementam o SQL2 assim como, PostgreSQL, Oracle, DB/2, Interbase, entre outros, porém, cada um deles possuem características e funções proprietárias. Ainda conforme o autor:

Estas características funcionais impostas pelo pelos mercados, aceitas e implementadas pelos desenvolvedores de SGBDs, têm suas vantagens e desvantagens. Ao atender a essas exigências de mercado, os softwares de banco de dados oferecem facilidades aos seus usuários, porém, em projetos que demandam o uso de mais de um desses gerenciadores (por exemplo, em projetos multiplataforma, em que um gerenciador não conseguirá atender a todas as plataformas a usar), os desenvolvedores terão muitas dificuldades, usando essas facilidades, principalmente ligadas à incompatibilidade nos códigos fontes, à não-implementação de algumas facilidades por um ou outro SGBD a usar, tanto no uso de SQL quanto nas Linguagens de Programação (um gerenciador implementa funções estendidas inexistentes ao outro gerenciador a ser usado ou ainda, um SGBD implementa algumas facilidades numa plataforma de sistema operacional, as quais não existem nas outras plataformas). (NETO, 2007, p. 25)

É possível ressaltar que os bancos de dados relacionais são os mais utilizados pela facilidade de aprendizado, pois “[...] não precisa entender teoria relacional para utilizar um banco de dados relacional [...], mas precisa entender alguns conceitos de banco de dados. ”. (WELLING e THOMSON, 2005, p. 156)

As bases de dados têm em comum conforme Welling e Thomson (2005):

- **Tabelas:** compõem as relações do banco de dados onde terá um nome, colunas correspondentes a cada fragmento dentro da tabela;
- **Colunas:** possuem nomes únicos dentro da tabela e contém dados diferentes tendo um tipo de dado associado;
- **Linhas:** representam os valores diferentes, porém, os mesmos atributos. Também é conhecido por registros ou *tuplas*;

- **Valores:** conjunto individual do valor que corresponde a coluna sendo especificado o tipo de dados pela sua coluna;
- **Chaves:** é o atributo de identificação único de cada registro na tabela. Existem dois tipos de chaves: primaria e estrangeira. A chave primaria faz a identificação do registro dentro da própria tabela e, a chave estrangeira é utilizada dentro da tabela para identificar uma relação entre tabelas, apontando para o identificador de outra tabela que está compondo esta relação;
- **Esquemas:** é utilizado para modelar em forma de desenho a estrutura do projeto da base de dados;
- **Relacionamentos:** estão representados pelas chaves estrangeiras.

2.11.1 PostgreSQL

O PostgreSQL é um SGBD baseado no padrão SQL2. Este SGBD permite a utilização da linguagem SQL, Triggers e programação idênticas a usados pela maioria de outros SGBDs disponível no mercado. A construção do PostgreSQL começou em 1986, na Universidade de Berkeley, Califórnia, a partir do seu descendente Ingres. Porém o PostgreSQL somente obteve um interpretador de SQL em 1995, um trabalho que foi realizado pelos profissionais Andrew Yu e Jolly Chen. Em 1996 teve a abertura do código fonte e, é a partir desta época que o SGBD começa a ser conhecido como PostgreSQL. Assim, várias contribuições estão sendo integradas ao SGBD tornando muito competitivo em nível de funcionalidades comparados aos outros SGBDs. (NETO, 2007)

O PostgreSQL é considerado um dos bancos de dados de código aberto mais avançado do mundo, se comparando com os bancos de dados comerciais da Oracle e DB2. Alguns recursos oferecidos pelo PostgreSQL serão descritos a seguir conforme (DRAKE e WORSLEY, 2011):

- Possui um sistema de controle de concorrência – *Multi-Version Concurrency Control* (MVCC) – uma tecnologia usada pelo PostgreSQL evitando o bloqueio desnecessário quando, um leitor estiver tentando ler algo da base de dados ao mesmo tempo que, alguém está tentando escrever gerando uma atualização;

- O PostgreSQL conta com uma API flexível que permite aos fornecedores oferecerem apoio ao desenvolvimento do PostgreSQL pela interface RDBMS que incluem as linguagens: Pascal, Perl, Python, PHP, ODBC, Java/JDBC, Ruby, entre outras;
- Por ser altamente extensível o PostgreSQL dá suporte as operações definidas pelo usuário como: métodos, funções e tipos de dados;
- A arquitetura do PostgreSQL é do tipo *process-per-user* cliente/servidor onde é fornecido uma conexão para cada cliente tentar se conectar no PostgreSQL;
- O PostgreSQL fornece suporte a integridade referencial como modo de proteção das validades de dados.

2.12 FRAMEWORK PARA O DESENVOLVIMENTO DE APLICAÇÕES RESPONSIVAS

No início do desenvolvimento orientados a objetos foi sugerido que os objetos eram as abstrações mais apropriadas para o reuso. Só que as experiências mostraram que os objetos são especializados e pequenos de mais para a utilização em aplicações específicas. Com tudo, notou-se que os objetos são melhores utilizados no reuso de processos apoiando os processos orientados a objetos nas abstrações de aplicações maiores conhecido como framework. (SOMMERVILLE, 2007)

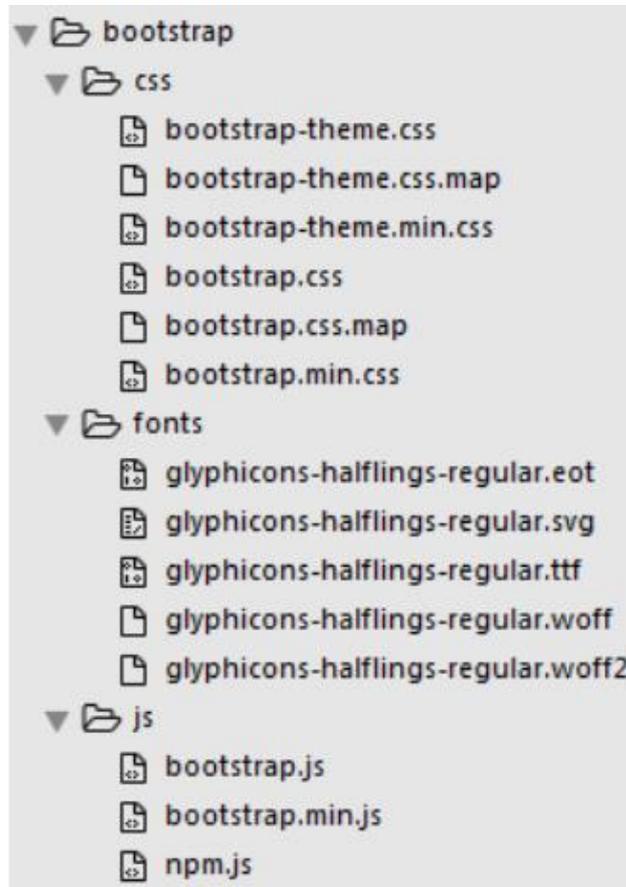
A utilização de um framework no desenvolvimento é muito útil, o que acaba acelerando o processo de desenvolvimento abstraindo muito código. No caso deste projeto, será analisado nas próximas seções alguns frameworks para o desenvolvimento do aplicativo onde, será feitas considerações sobre os três frameworks – Bootstrap, Skeleton e Foundation e, posteriormente escolhido um para o auxílio na implementação do aplicativo.

2.12.1 Bootstrap

O Bootstrap é framework de auxílio no desenvolvimento *fornt-end* composto pelos documentos HTML e CSS e também, por arquivos em JavaScript. Criado em 2011 por Mark Otto e Jacob Thontom para o auxílio no desenvolvimento de aplicações do Twitter e posteriormente aberto a comunidade de desenvolvedores que, contribuem com o projeto fornecendo códigos, tornando o Bootstrap um dos framework *open-source* mais utilizados na categoria.

A estrutura de composição do Bootstrap este dividido em pastas da seguinte forma:

Figura 11 - Arquivos do Bootstrap



Fonte: Do Autor

- **CSS:** nesta pasta contém os arquivos de folhas de estilos pré-definidos que compõem a estrutura básica do *template* fornecido pelo framework;
- **JS:** é a pasta onde se encontram os arquivos desenvolvidos em JavaScript. Estes arquivos tratam das requisições do layout e suas validações como por exemplo, tratar de atributos que não são suportados por alguns navegadores;
- **Fonts:** aqui ficam arquivos com alguns temas pré-definidos no Bootstrap.

O arquivo completo do Bootstrap com todos os temas e bibliotecas não ultrapassam 1MB, sendo considerado um arquivo leve para desenvolvimento de aplicações web. Existe também a possibilidade de personalizar os arquivos que se deseja utilizar do framework como mostra a figura 12:

Figura 12 - Opções de customização do Bootstrap

Common CSS	Components	JavaScript components
<input checked="" type="checkbox"/> Print media styles	<input checked="" type="checkbox"/> Glyphicons	<input checked="" type="checkbox"/> Component animations (for JS)
<input checked="" type="checkbox"/> Typography	<input checked="" type="checkbox"/> Button groups	<input checked="" type="checkbox"/> Dropdowns
<input checked="" type="checkbox"/> Code	<input checked="" type="checkbox"/> Input groups	<input checked="" type="checkbox"/> Tooltips
<input checked="" type="checkbox"/> Grid system	<input checked="" type="checkbox"/> Navs	<input checked="" type="checkbox"/> Popovers
<input checked="" type="checkbox"/> Tables	<input checked="" type="checkbox"/> Navbar	<input checked="" type="checkbox"/> Modals
<input checked="" type="checkbox"/> Forms	<input checked="" type="checkbox"/> Breadcrumbs	<input checked="" type="checkbox"/> Carousel
<input checked="" type="checkbox"/> Buttons	<input checked="" type="checkbox"/> Pagination	
<input checked="" type="checkbox"/> Responsive utilities	<input checked="" type="checkbox"/> Pager	
	<input checked="" type="checkbox"/> Labels	
	<input checked="" type="checkbox"/> Badges	
	<input checked="" type="checkbox"/> Jumbotron	
	<input checked="" type="checkbox"/> Thumbnails	
	<input checked="" type="checkbox"/> Alerts	
	<input checked="" type="checkbox"/> Progress bars	
	<input checked="" type="checkbox"/> Media items	
	<input checked="" type="checkbox"/> List groups	
	<input checked="" type="checkbox"/> Panels	
	<input checked="" type="checkbox"/> Responsive embed	
	<input checked="" type="checkbox"/> Wells	
	<input checked="" type="checkbox"/> Close icon	

Fonte: <http://getbootstrap.com/2.3.2/customize.html> - (2015)

A figura acima é um exemplo da estrutura geral personalizável, mas existem outros componentes que podem ser personalizados antes de compilar e baixar o arquivo como: *plugins* da biblioteca jQuery, a personalização das variáveis do CSS, cores, definição de atributos dos estilos globais do framework, tipografias que definem as cores do corpo do texto e títulos, alterações do nomes e cores da Iconografia, definição do preenchimento dos componentes comum, definição de tabelas com os valores a serem atribuídos, botões entre outras coisas.

O Bootstrap foi desenvolvido para dar suporte na visualização tanto em desktops quanto para dispositivos moveis levando em considerações as últimas versões dos navegadores, que compilam as linguagens HTML5 e CSS3. Porém, a visualização da página em navegadores antigos também é possível de forma diferente, mas totalmente funcional. Abaixo a figura 13 mostra as versões suportadas por cada navegador:

Figura 13 - Navegadores que suportam o Bootstrap

	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	✓ Supported	✓ Supported	N/A	✗ Not Supported	N/A
iOS	✓ Supported	N/A		✗ Not Supported	✓ Supported
Mac OS X	✓ Supported	✓ Supported		✓ Supported	✓ Supported
Windows	✓ Supported	✓ Supported	✓ Supported	✓ Supported	✗ Not Supported

Fonte: <http://getbootstrap.com/getting-started/> - (2015)

Segundo as informações apresentadas no site do Bootstrap, outros navegadores para o sistema operacional Linux - Chrome e o Firefox, apresentam suportes satisfatórios quanto a apresentação nestes navegadores. No entanto, para as versões 7 e 8 do IE, embora tenham resultados positivos quanto a visualização neles, alguns atributos do HTML5 e CSS3 não são suportados:

Figura 14 - Lista com os principais atributos não suportados pelo navegador IE versão 7 e 8

Feature	Internet Explorer 8	Internet Explorer 9
border-radius	✗ Not supported	✓ Supported
box-shadow	✗ Not supported	✓ Supported
transform	✗ Not supported	✓ Supported, with <code>-ms</code> prefix
transition	✗ Not supported	
placeholder	✗ Not supported	

Fonte: <http://getbootstrap.com/getting-started/> - (2015)

Para tentar solucionar estes problemas com a compatibilidade, a utilização do Respond que “[...] um *polyfill* que faz com que IE6-8 reconheçam `min-width` e `max-width` para o uso de média queries.” (ZEMEL, 2013, p. 138). Com tudo, salienta-se a importância quanto a utilização do Respond devido ao conflito de interpretação do navegador, para isto, é importante ler atentamente a documentação do framework.

2.12.2 Foundation

O Foundation é um framework desenvolvido pela companhia ZURB que foi fundada por Bryan Zmijewski com sede na cidade Campbell no estado da Califórnia Estados Unidos. A companhia é da área de desenvolvimento web e lançou em 2011 a versão 2.0 do Foundation. Atualmente o framework está na versão 5 onde conta, entre outros recursos, funcionalidades que contemplam a acessibilidade – navegação pelo teclado e acesso para leitores de telas. Em junho de 2014 a companhia ZURB comunicou através do seu blog que, a partir da versão 5 o framework passaria a se chamar Foundation *for sites* e, a partir do 6 Foundation *for apps*, onde terá o foco no desenvolvimento de aplicações responsivas. Mas desde a versão 2.0 o framework já vem adotando o desenvolvimento responsivo e, a partir da versão 4.0 adota o conceito de *mobile first*. (SILVA, 2014)

O framework é *open-source* com a licença registrada no MIT. Sua utilização consiste em baixar o arquivo do site podendo também ser customizado conforme a necessidade de cada projeto. Na customização poderá ser definido o tipo de grid, componentes como tabelas, barra de progressos entre outros. Também poderá ser selecionados os arquivos em JavaScript que irá compor o projeto, além da personalização dos componentes – troca de cores, mensagens de validações além das restrições e mascaras do campo. A figura 15 demonstra um exemplo de customização:

Figura 15 - Customização do Foundation

Customize Foundation

Pick and choose the features you want or have the whole enchilada.

Choose Your Components:

All Foundation Components

GRID:

Grid (?) Block Grid (?)

NAVIGATION:

Pagination (?) Breadcrumbs (?)
 Side Nav (?) Sub Nav (?)
 Icon Bar (?)

BUTTONS:

Buttons (?) Dropdown Buttons (?)
 Button Groups (?) Split Buttons (?)

FORMS:

Forms (?) Switches (?)

TYPOGRAPHY:

Type Inline Lists
 Keystrokes Labels

OTHER CSS COMPONENTS:

Panels Pricing Tables
 Progress Bars Thumbs
 Tables Flex Video
 Visibility

JAVASCRIPT COMPONENTS:

Abide Magellan
 Accordion Off Canvas
 Alert Boxes Orbit (DEPRECATED)
 Clearing Reveal
 Dropdown Slider
 Equalizer Tabs
 Interchange Tooltips
 Joyride Top Bar

[Download Custom Build](#)

Set Your Defaults:

THE GRID

of Columns: Gutter:

Max-Width:

COLORS

Primary Color: Secondary Color:

Alert Color: Success Color:

Body Font Color: Header Font Color:

GLOBAL RADIUS

Radius:

TOP BAR BREAKPOINT

Breakpoint:

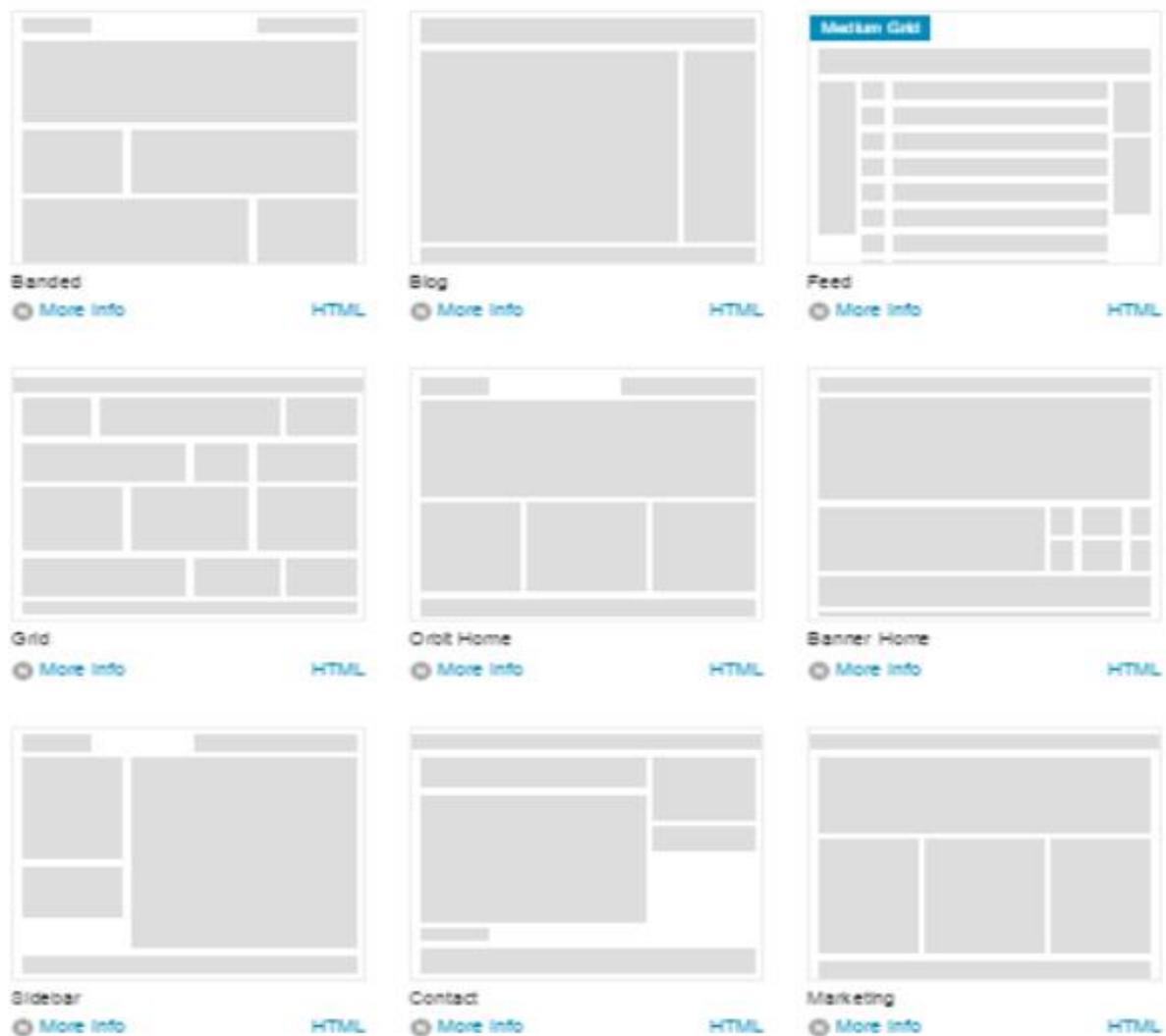
TEXT DIRECTION

Left-to-right Right-to-left

Fonte: <http://foundation.zurb.com/develop/download.html> - (2015)

Além da customização o framework fornece alguns modelos de *templates* prontos para serem adaptados de acordo com o projeto, como mostra a figura 16:

Figura 16 - Modelos de template do Foundation



Fonte: <http://foundation.zurb.com/templates.html> - (2015)

Entre outras características do Foundation, destaca-se as seguintes conforme Silva(2014):

- **Semântica:** a marcação HTML foi projetada para preservar o valor semântico dos elementos, o que resulta em uma estrutura limpa com um carregamento otimizado;
- **Mobile first:** utilização do conceito mobile first no design responsivo, o que acaba priorizando o desenvolvimento primeiro para dispositivos pequenos e, acrescentando melhorias progressivas para atender os dispositivos com resoluções maiores até chegar aos desktops, tendo assim boas práticas no design responsivo;

- **Personalização:** o desenvolvedor pode personalizar o framework definindo o download de apenas dos plugins que deseja ser usado no projeto, bem como definir cores, tamanhos de fontes entre outros.

2.12.3 Skeleton

Desenvolvido por Dave Gamache em 2011, o Skeleton é framework no auxílio de desenvolvimento de aplicações web focado no design responsivo. Este framework é composto basicamente por um arquivo CSS com um grid que vai até 960px, além de formulários, botões e *media queries*. Por utilizar consultas de media queries do CSS3, ele não dá suporte para as versões antigas dos navegadores.

O Skeleton segue o conceito de *mobile-first* e não possui a opção de personalização do código. Basicamente entre no site faz o download do arquivo e começa a utilização seguindo a estruturação da linguagem do HTML5. As consultas definidas nas queries são as seguintes:

- Desktop HD: 1200px;
- Desktop: 1000px;
- Tablet: 750px;
- Phablet: 550px;
- Mobile: 400px.

2.13 FERRAMENTAS PARA AVALIAR O DESIGN RESPONSIVO

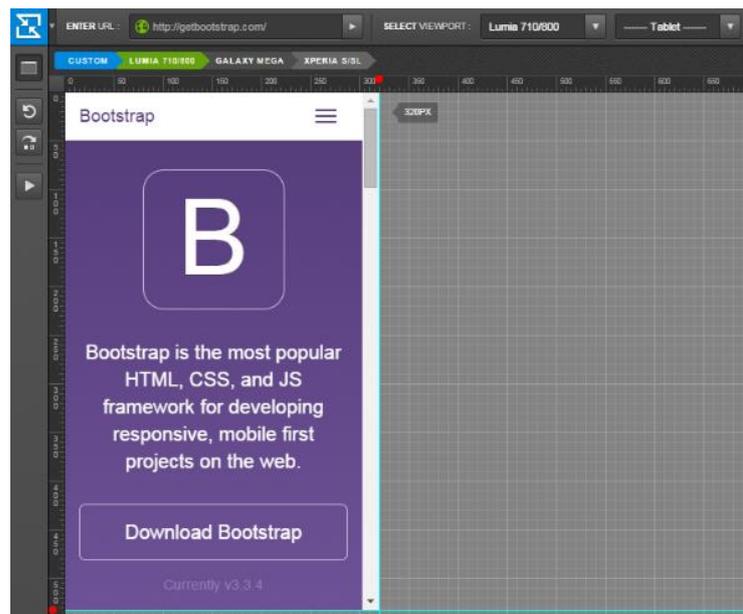
Para auxiliar o teste de uma aplicação simulando a apresentação em diferentes tipos de dispositivos, é essencial a utilização de ferramentas que serão apresentadas a seguir. Estas ferramentas dão apoio na elaboração do projeto por, fazerem simulação como se estivesse sendo acessados por dispositivos com as características reais de cada um que estão disponíveis dentro de cada ferramenta.

2.13.1 ScreenQueries

O ScreenQueries¹⁰ é uma ferramenta *open-source* disponível na web para testes de design responsivo onde é possível escolher em uma lista, os navegadores padrões de vários aparelhos que deseja fazer a emulação da aplicação, podendo também inverter a posição do simulador em modo retrato/paisagem.

Para utilizar a ferramenta de teste é preciso criar uma conta e após, fazer login no sistema. Dentro do sistema o usuário informa a URL da aplicação que deseja testar e espere até que seja carregada. A utilização é simples, porém, é recomendado pelos desenvolvedores desta ferramenta, que a visualização do teste serve apenas como referência, sendo o seu resultado não ser totalmente confiável quando se comparado com os dispositivos reais. A figura 17 demonstra um exemplo de emulação do site do Bootstrap:

Figura 17 - Exemplo de emulação do site do Bootstrap no ScreenQueries



Fonte: <http://beta.screenqueri.es/#> - (2015)

2.13.2 Responsinator

O Responsinator¹¹ é uma ferramenta *online* que conta com uma versão *open-source* para teste de do design responsivo. Sua utilização consiste em inserir a URL da aplicação

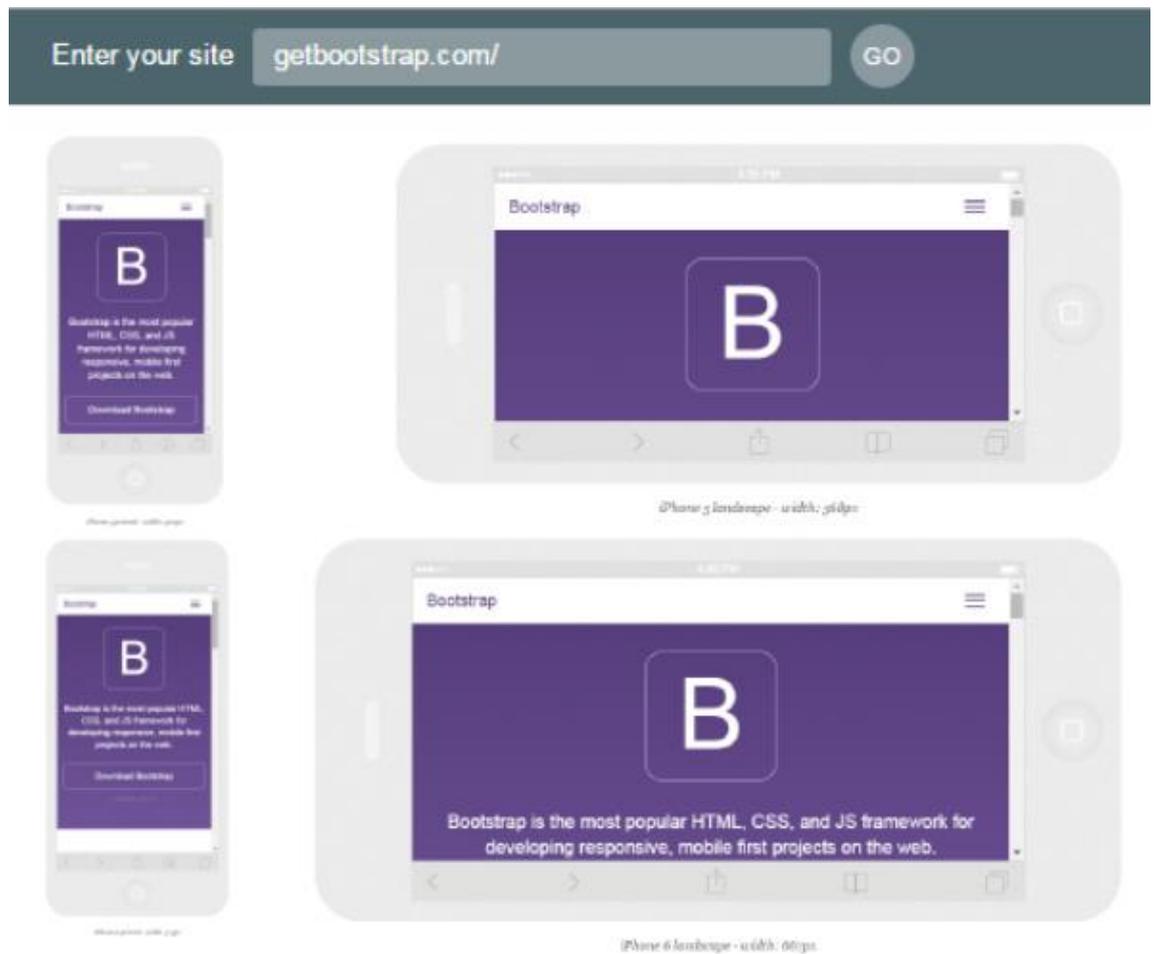
¹⁰ <http://beta.screenqueri.es/>

¹¹ <http://www.responsinator.com/>

(podendo ser local), onde após é carregado toda a aplicação em seus dispositivos de teste de uma só vez.

Ao contrário do ScreenQueries o Responsinator emula apenas em seis dispositivos em modo retrato e paisagem todos ao mesmo tempo. Porém, o usuário consegue interagir com aplicação em teste com todos os dispositivos. A figura 18 mostra exemplo de emulação do site Bootstrap nos modelos de iPhone 5 e 6 representado no Responsinator:

Figura 18 - Exemplo de emulação no Responsinator dos modelos do iPhone 5 e 6



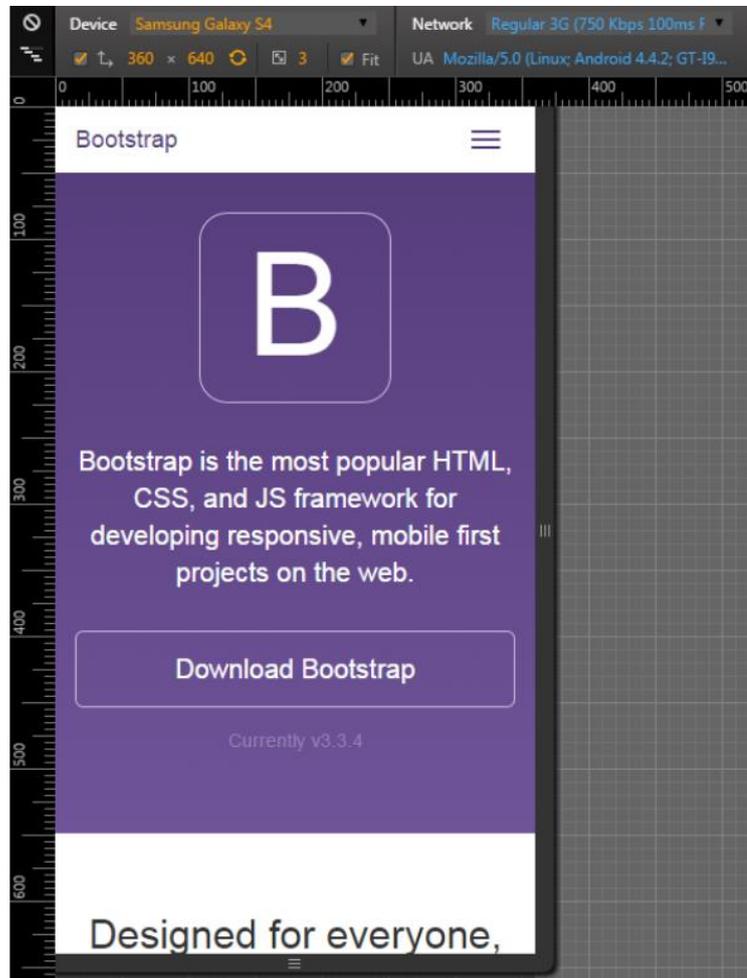
Fonte: <http://www.responsinator.com/?url=http://getbootstrap.com/> - (2015)

2.13.3 Device mode do Google Chrome

O navegador Google Chrome possui uma extensão o Device Mode que está disponível em inspecionar elementos. Nesta função, o usuário pode assumir a navegação de uma aplicação em diferentes tipos de dispositivos disponível em uma lista. Também é possível escolher um tipo de rede para testar a velocidade de conexão quanto ao carregamento

da aplicação. A figura 19 mostra a emulação do carregamento do site do Bootstrap em um dispositivo Samsung Galaxy S IV, com uma conexão de rede 3G¹²:

Figura 19 - Exemplo de emulação do site Bootstrap no Device Mode do Google Chrome



Fonte: Do autor

¹² 3G significa terceira geração de rede de celulares e internet.

3 METODOLOGIA

Neste capítulo é apresentado a classificação da pesquisa e a divisão desta metodologia em etapas. Cada uma das etapas compôs um método de estudo e elaboração da monografia.

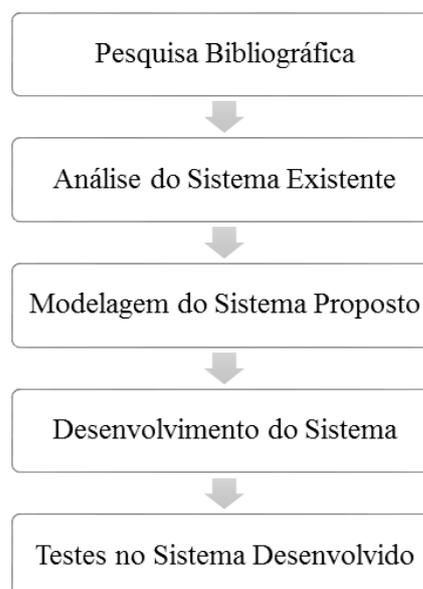
3.1 CLASSIFICAÇÃO DA PESQUISA

Este trabalho é classificado como uma pesquisa exploratória devido ao fato de fazer um levantamento bibliográfico referente ao tema, propondo um estudo de caso para exemplificar o trabalho proposto. Conforme Gil (2002), as pesquisas exploratórias tentam familiarizar os problemas como objetivo principal ou ainda, constitui-se hipóteses com vistas mais explícitas. Assim, o “[...] planejamento da pesquisa exploratória seja bastante flexível, na maioria dos casos assume a forma de pesquisa bibliográfica ou de estudo de caso[...]”. (GIL, 2002, p. 41)

3.2 ETAPAS DO PROJETO

Este trabalho seguiu as seguintes etapas de desenvolvimento conforme apresentado na figura 20:

Figura 20 - Etapas da metodologia utilizada



Fonte: Do Autor

3.2.1 Pesquisa bibliográfica

A pesquisa bibliográfica foi a primeira etapa no desenvolvimento deste Projeto. Nesta etapa, foram feitas pesquisas sobre o tema do trabalho. Esta pesquisa se constitui no levantamento do tema em livros, artigos publicados em periódicos e revistas e na internet, blogs e fóruns com assuntos correlacionados.

O referencial teórico começa destacando o cenário da web móvel. Logo a monografia, traz o contexto de desenvolvimento de aplicações seguindo o conceito do *mobile first* para, posteriormente abordar o design responsivo, dividindo este referencial em partes, de modo que, seja possível compreender o desenvolvimento de aplicações responsivas e suas estruturas.

As linguagens de programação e de desenvolvimento de aplicações web, tiveram um destaque sobre suas últimas versões até a data de escrita deste trabalho. O referencial focou os últimos recursos que algumas linguagens incorporaram, destacando assim, suas novas características quanto suas implementações. Também foi apresentada uma definição da linguagem UML, a qual foi utilizado para a etapa de análise e modelagem do sistema.

Concluindo, o trabalho apresentou os três tipos de frameworks que podem ser utilizados no desenvolvimento rápido de aplicações responsivas e, três ferramentas para realização de testes. Neste estudo, focou-se nas características de cada um dos componentes de cada framework e ferramenta.

3.2.2 Análise do sistema existente

O controle de empréstimos do IFSUL *campus* Passo Fundo é feito de maneira manual, ou seja, todos os registros são anotados em uma ficha para o controle e registro das informações pertinentes ao solicitante, junto com os materiais emprestados. Dentro deste registro, são anotadas informações referentes a data de solicitação, a hora da retirada, o nome do solicitante e o material retirado. Estas são informações de entrada do registro. Outras informações como motivo, data e hora e, por quem foi devolvido também constam neste registro. A ficha de movimentação pode ser conferida no Anexo A.

Todas as informações ficam disponíveis para conferência da secretária do *Campus*, porém, a dificuldade de controle se deve ao grande número de solicitações realizadas nos três

períodos do horário de trabalho no *Campus*. Os agentes envolvidos nos processos são: os funcionários administrativos, professores e, funcionários terceirizados.

Através das entrevistas feitas com os envolvidos no processo de empréstimos, foram definidos os requisitos do sistema que será apresentado no próximo item do capítulo.

3.2.3 Modelagem do sistema proposto

A descrição de um sistema acaba compondo os requisitos de aplicação, definindo o que o sistema tem que realizar, quais serão os serviços disponibilizados e, que restrições os sistemas terão. Desta forma, os requisitos servem de parâmetros para a modelagem da aplicação antes mesmo de começar o desenvolvimento. Sendo assim, o requisito “[...]não é usado de forma consistente pela indústria de software. Em alguns casos, o requisito é apenas uma declaração abstrata em alto nível de um serviço que o sistema deve oferecer ou a restrição a um sistema. ”. (SOMMERVILLE, 2011, p. 57)

Segundo Sommerville (2011), os requisitos de softwares são classificados frequentemente em requisitos funcionais e não funcionais sendo:

- Requisitos funcionais: são os serviços declarados que o sistema deve fornecer, como o sistema deve reagir em determinadas entradas específicas e, ainda como é seu comportamento em algumas situações.
- Requisitos não funcionais: descreve as restrições aos serviços e as funções que são oferecidas pelo sistema. Isto incluem, “[...]restrições de timing, restrições no processo de desenvolvimento e restrições impostas pela norma. ”. (SOMMERVILLE, 2011, p. 59)

Para que fosse possível desenvolver o sistema de gerenciamento de empréstimos dos materiais, foram feitas entrevistas com a secretaria do campus onde, foram definidos os requisitos do sistema e, posteriormente, a modelagem conforme as especificações técnicas da UML.

3.2.3.1 Requisitos funcionais

O sistema deverá ter opções de acordo com o tipo de categoria do usuário. Sendo assim, o administrador do sistema cadastra a pessoa fornecendo a ela um login e senha. Para

os funcionários efetivos do campus, o login utilizado será o crachá com a identificação do funcionário, para os demais, é fornecido um login estabelecido pelo administrador.

Sendo assim, foi desenvolvido uma área específica de acordo com a categoria do usuário composta pelos requisitos a seguir:

- Área da secretaria
 - Fazer login;
 - Manter categorias;
 - Manter pessoas;
 - Manter prédios,
 - Manter locais,
 - Manter tipos de materiais,
 - Manter materiais.

- Área do professor
 - Fazer login;
 - Manter dados pessoais: alterar senha, telefone e e-mail;
 - Consultar horários;
 - Efetuar retirada do material.

- Área do coordenador
 - Manter disciplinas,
 - Manter turmas,
 - Manter cursos,
 - Manter horários,
 - Efetuar retirada de material.

- Área do usuário
 - Fazer login;
 - Manter dados pessoais: alterar senha, telefone e e-mail;
 - Efetuar retirada de material.

3.2.3.2 Requisitos não funcionais

- O sistema deve ser acessível no ambiente web por qualquer dispositivo que possua navegador e acesso à internet;
- A aplicação foi desenvolvida utilizando o SGBD PostgreSQL.

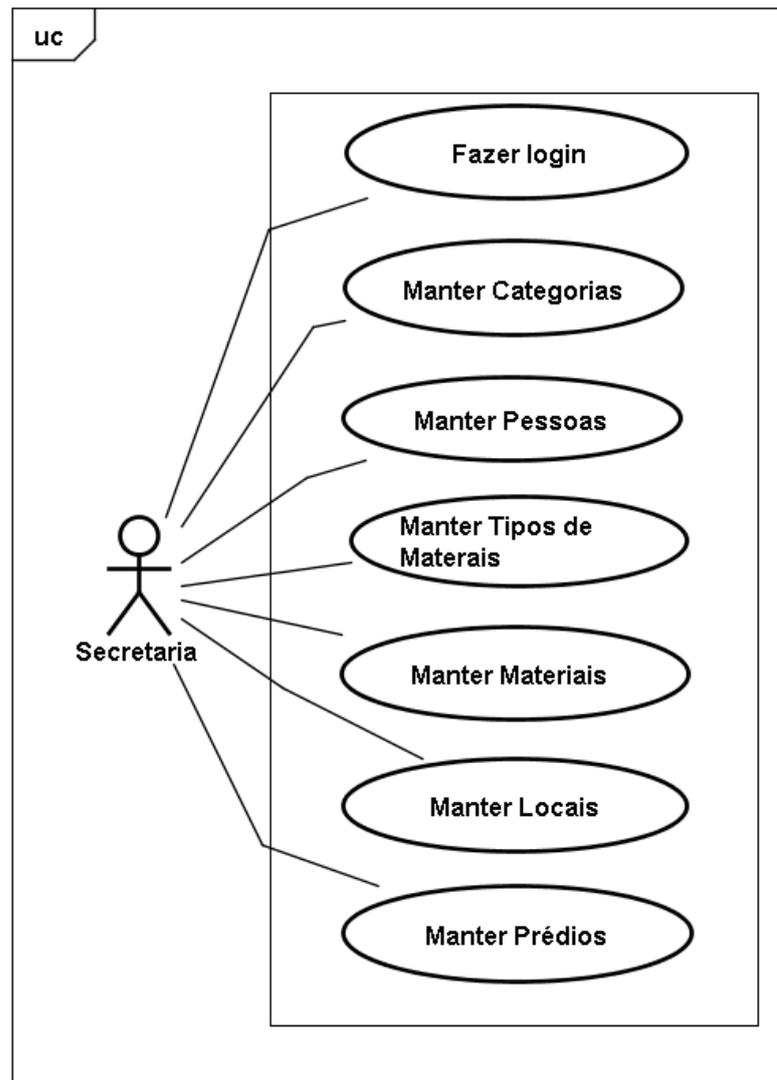
3.2.3.3 Diagrama de casos de uso

Os diagramas de casos de uso ajudam a descrever as sequencias de interações do usuário com o sistema. Dentro deste modelo de diagrama, é utilizado um ator representando suas ações no sistema de acordo com a elaboração dos requisitos funcionais. Sendo assim, abaixo serão apresentados os diagramas de casos de uso elaborado para este projeto.

3.2.3.3.1 Diagrama casos de uso – área da secretaria

Para acessar o sistema a secretaria precisa fazer sua autenticação, informando o login e senha. Ela é a responsável pelo cadastro de usuários. Dentro do cadastro do usuário ela preencherá as informações básicas como: selecionar a categoria da pessoa, informar o nome, e-mail, telefone, situação (ativo/não ativo), permissão e definir uma senha que, depois pode ser alterado pelo usuário. Assim, sua representação no caso de uso fica desta forma, como mostra a figura 21 logo abaixo:

Figura 21 - Diagrama de casos de uso: secretaria



Fonte: Do Autor

Além de ser responsável pelo cadastro do usuário, também é de sua responsabilidade manter as categorias. As categorias referem-se aos cargos que os usuários possuem dentro do campus: professor, coordenador do curso, funcionário administrativo, entre outros.

Manter os tipos de materiais segue a mesma lógica das categorias. Os materiais são classificados por tipos: projetor, apagador, extensão elétrica, etc. Assim, no cadastro de um novo material é informado seu código, selecionado seu tipo e, caso necessário, pode ser descrito alguma observação, mas que não é obrigatória. E o mesmo caso acontece para prédios e locais. Os prédios correspondem onde os locais se encontram. Dentro do cadastro de prédios são informados sua descrição (exemplo: prédio da informática), o número de salas que este prédio possui e, também existe um campo observação, caso exista alguma

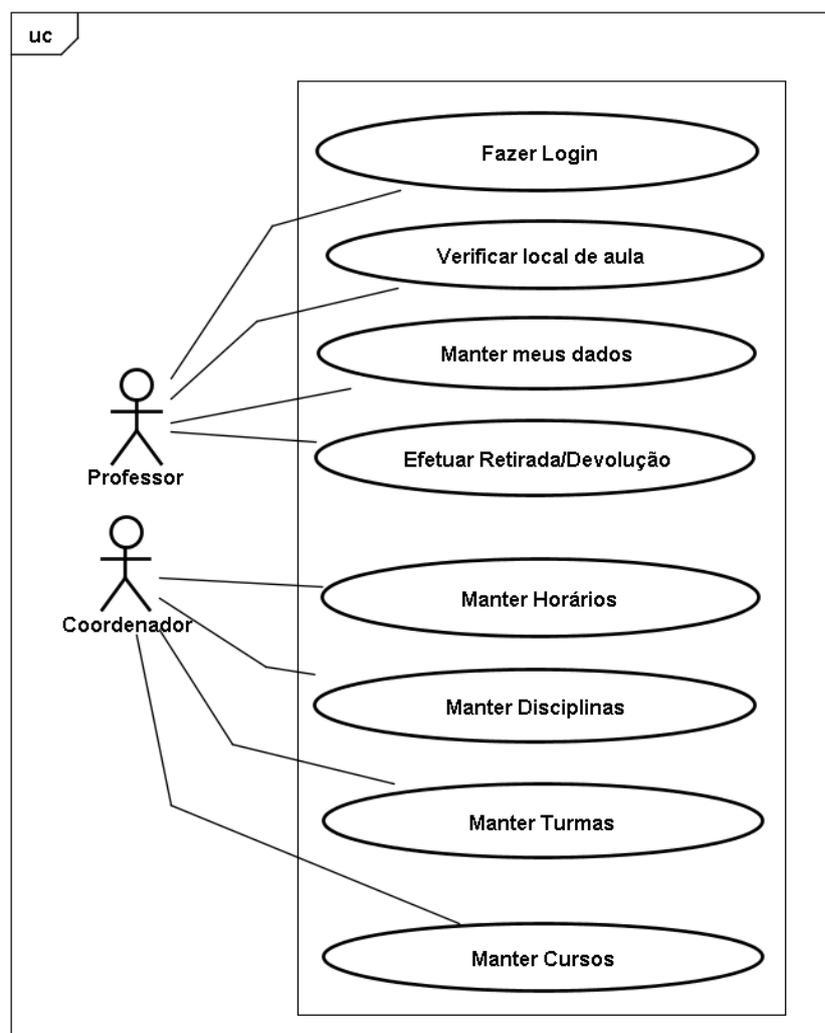
informação extra que precise ser informado. Por sua vez, os locais estão relacionados diretamente com algum prédio e, o seu cadastro contém informações como: código, descrição e a qual prédio pertence.

Portanto estas são as interações que a secretaria tem dentro do sistema. A documentação deste caso de uso está disponível no Apêndice A.

3.2.3.3.2 Diagrama de casos de uso – área do professor/coordenador

O diagrama de casos de uso da área do professor/coordenador, também elaborado de acordo com os requisitos funcionais, possui a seguinte situação. Veja a figura a seguir:

Figura 22 - Diagrama de casos de uso: professor/coordenador



Fonte: Do Autor

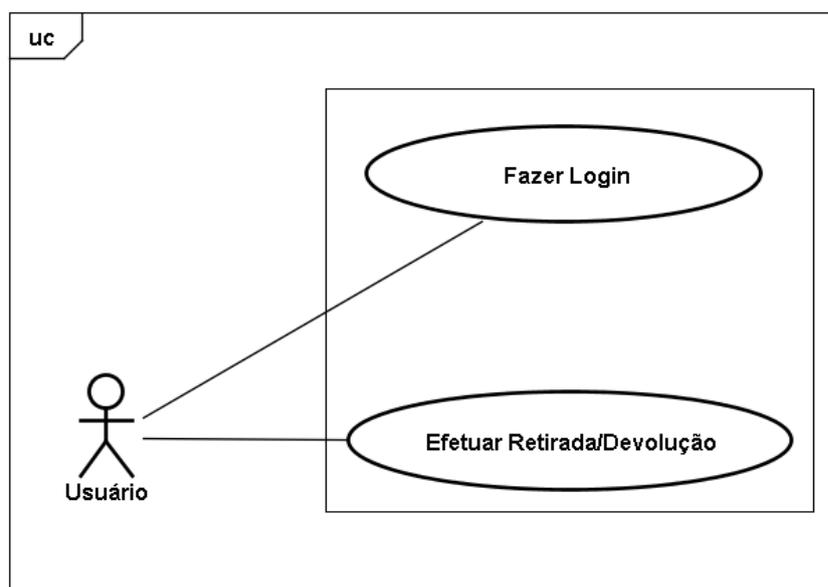
Este modelo de caso de uso representa a ação de uma pessoa que pode assumir a ação de dois atores no sistema: a pessoa pode ser o professor e coordenador ao mesmo tempo. Sendo assim, ao efetuar o login, o professor pode consultar o local de aula, fazer as alterações dos seus dados pessoais como senha, telefone e e-mail e, também, fazer a retirada do material.

Já o coordenador do curso é o encarregado de elaborar os horários dos professores. Para isto, o coordenador precisa cadastrar as disciplinas informando a descrição e a abreviatura. Para as turmas e cursos é preciso informar apenas suas descrições. A partir destes cadastros básicos, é elaborado os horários individuais de cada professor contendo as seguintes informações: turma, curso, semestre, dia da semana, professor, disciplina, local e os períodos de aulas. Também é possível fazer a edição ou exclusão de um horário selecionado. A documentação deste caso de uso encontra-se no Apêndices B e C.

3.2.3.3.3 Diagrama de casos de uso – área do usuário

Este é o diagrama de casos de uso mais simples dentre projeto que têm em comum a retirada de materiais e, também, pode verificar os dados pessoas como explicado nos modelos anteriores. Sua documentação está no Apêndice D e, sua representação ficou da seguinte conforme a figura 23:

Figura 23 - Diagrama de casos de uso: usuário



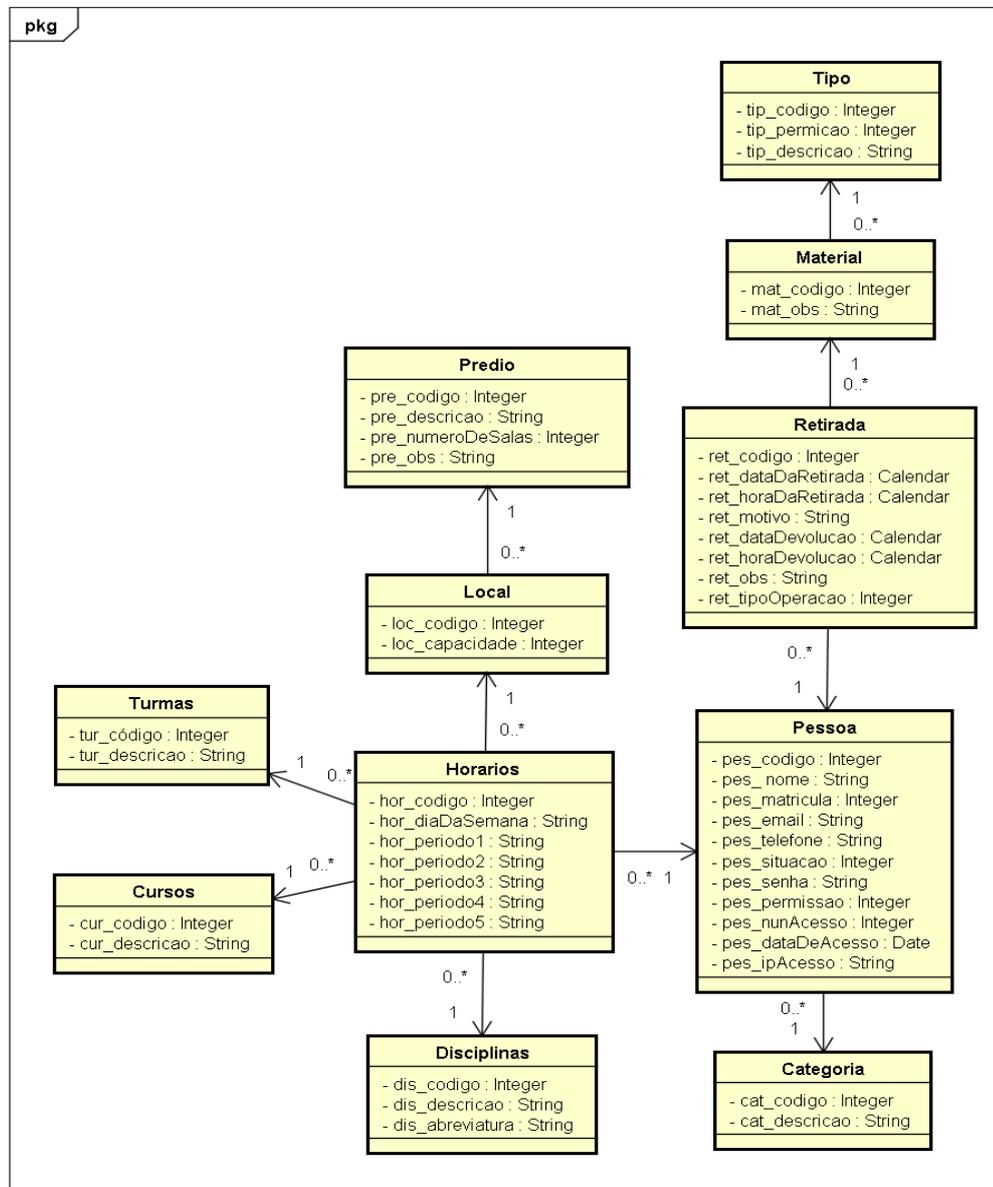
Fonte: Do Autor

Este usuário pode ser um funcionário do campus/terceirizado e sua interação consiste através da autenticação no sistema.

3.2.3.4 Diagrama de classes do sistema proposto

Para poder visualizar as classes que compõem este projeto, foi elaborado um diagrama de classes descrevendo cada classe do sistema junto com seus atributos e relações entre as classes. Desta forma, o diagrama ficou assim:

Figura 24 - Diagrama de classe do sistema proposto



Como demonstrado na figura 24, o diagrama de classes possui a classe pessoa com alguns atributos não mencionados no documento casos de uso da secretária. Estes atributos: data de acesso, número de acessos e IP (*Internet Protocol*), são preenchidos automaticamente pelo sistema cada vez que um usuário faz um acesso.

Outro fato importante é a relação entre retirada e material e, retirada e local, porque, esta relação é do tipo não obrigatória. Quando um usuário solicita um item para a secretaria, ao concluir a retirada, o sistema verifica se o código informado é do tipo material ou local, fazendo a relação correspondente ao código informado.

3.2.4 Desenvolvimento do sistema

No desenvolvimento do sistema foi utilizado a *Integrated Development Environment*¹³ (IDE) Netbeans¹⁴ versão 8.02 e o Sublime Text 3¹⁵. Estas duas ferramentas são *open-source* e tem um ambiente de configuração de projetos muito simples.

Foram utilizadas as linguagens HTML5, CSS3, PHP, JavaScript e SQL, além da UML para montagem do projeto usando os diagramas casos de uso e o diagrama de classe visto anteriormente. O HTML5 para o desenvolvimento das interfaces fazendo os estilos das páginas com a inserção do CSS3. Foi implementado as *medias queries* do CSS3 para definir estilos de acordo com as resoluções de telas. Por sua vez, o PHP por ser uma linguagem interpretada do lado do servidor e por ter características dinâmicas, foi utilizado para criação de scripts de autenticação de usuário e criação de menus de acordo com a sua permissão, fazendo o uso de sessões, testes antes de persistir os dados no banco de dados.

O JavaScript compôs essencialmente as validações dos campos de senha, máscara no campo telefone e ajudou a focalizar campos como na troca de abas da tela inicial. Mas o uso do JavaScript codificado foi substituído pela inserção da biblioteca JQuery. Para a criação de listagem de dados, utilizou-se o DataTable do JQuery. E por fim, o uso do SQL para a criação das tabelas da base de dados tendo como SGBD o PostgreSQL.

¹³ IDE é uma sigla em inglês para *Integrated Development Environment* que em português significa Ambiente para Desenvolvimento Integrado. São ferramentas de computadores para o desenvolvimento de softwares que tem como característica, agilizar este processo.

¹⁴Disponível em <https://netbeans.org/>

¹⁵ Disponível em <http://www.sublimetext.com/>

Para o desenvolvimento rápido de aplicações responsivas, este projeto fez a escolha da utilização do Bootstrap versão 3.

3.2.5 Teste do sistema desenvolvido

A melhor forma para testar uma aplicação é aplicar os testes em máquinas reais. Todavia, testar o design responsivo em vários dispositivos tornaria este processo simplesmente complicado. Então a solução encontrada, foi utilização de ferramentas que, simulam dispositivos reais, como mencionado no capítulo 2.13.

Na avaliação das ferramentas optou-se neste trabalho pela utilização da ferramenta do browser Google Chrome, Device Mode. A escolha desta ferramenta se deve ao fato de estar vinculada ao navegador e sua utilização está no item inspecionar elemento ou f12 do teclado. Esta ferramenta apresenta uma série de recursos não disponíveis nas outras ferramentas apresentadas: Responsinator e ScreenQueries.

Os recursos disponíveis pelo Device Mode além dos simuladores de dispositivos, são as opções de simulação de navegação escolhendo um tipo de rede de internet. É um recurso fundamental para testar o design responsivo, assim, o desenvolvedor pode verificar por exemplo, o tempo que uma requisição leva para atender conforme o tipo de rede de internet simulada, acompanhado o tempo na janela network da ferramenta.

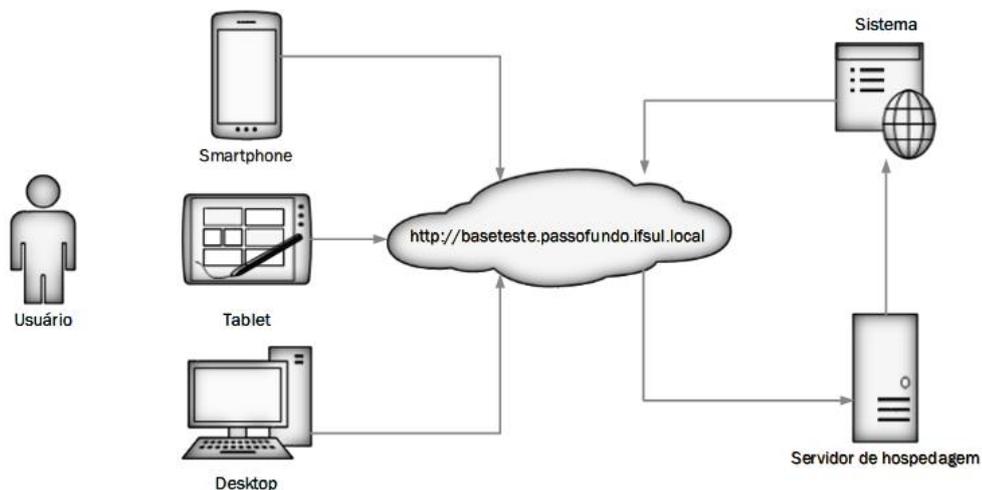
4. SISTEMA DESENVOLVIDO

Este capítulo apresentara o funcionamento do sistema junto com algumas telas e os padrões do sistema.

4.1. FUNCIONAMENTO DO SISTEMA

O sistema desenvolvido ficou disponível para o acesso internamente no *campus* onde o usuário acessava através de um navegador como ilustra a figura 25:

Figura 25 - Modelo de funcionamento do sistema desenvolvido



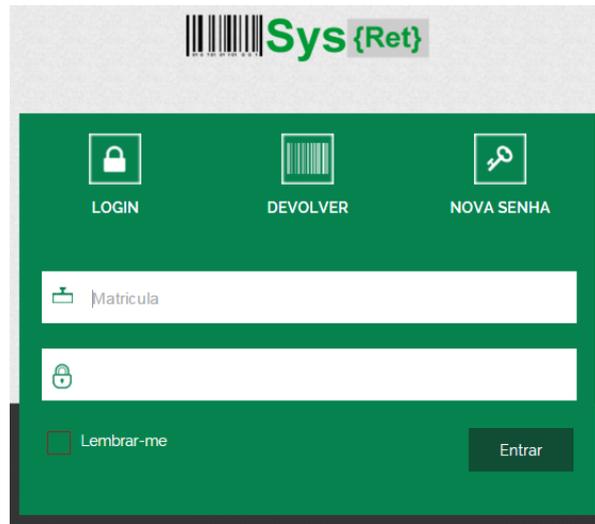
Fonte: Do Autor

Para fazer a retirada de um material o usuário precisa acessar o sistema na recepção onde encontra-se uma máquina disponível para utilização, solicita o item desejado, informa o código de barras por um leitor e finaliza seu pedido confirmando no formulário de retirada. O usuário pode fazer outras consultas/acesso de qualquer lugar no campus e independentemente de dispositivo.

4.1.1. Tela inicial

A tela inicial do sistema foi desenvolvida para a realização da autenticação do usuário onde o mesmo informe o número do seu crachá/registro e sua senha. Também é possível salvar o login e senha selecionando a opção lembrar-me.

Figura 26 - Tela inicial: login

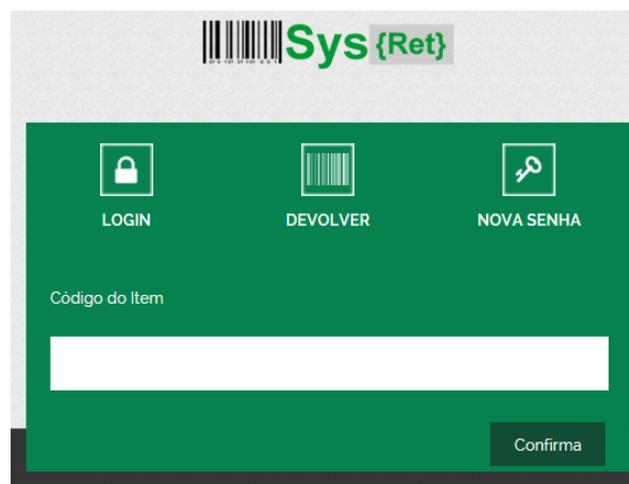


The screenshot shows the login interface for 'Sys {Ret}'. At the top, there is a logo consisting of a barcode and the text 'Sys {Ret}'. Below the logo, there are three main navigation buttons: 'LOGIN' (with a lock icon), 'DEVOLVER' (with a barcode icon), and 'NOVA SENHA' (with a key icon). The central part of the screen contains a white input field for 'Matricula', a password field with a lock icon, a 'Lembrar-me' checkbox, and an 'Entrar' button.

Fonte: Do Autor

Outra funcionalidade na tela inicial é fazer devolução do material clicando na opção devolver. Nesta opção, o usuário não precisa realizar a autenticação no sistema para efetuar a devolução do material, basta simplesmente informar o código de barras do item a ser devolvido, como mostra a figura 27:

Figura 27 - Tela inicial: devolver item

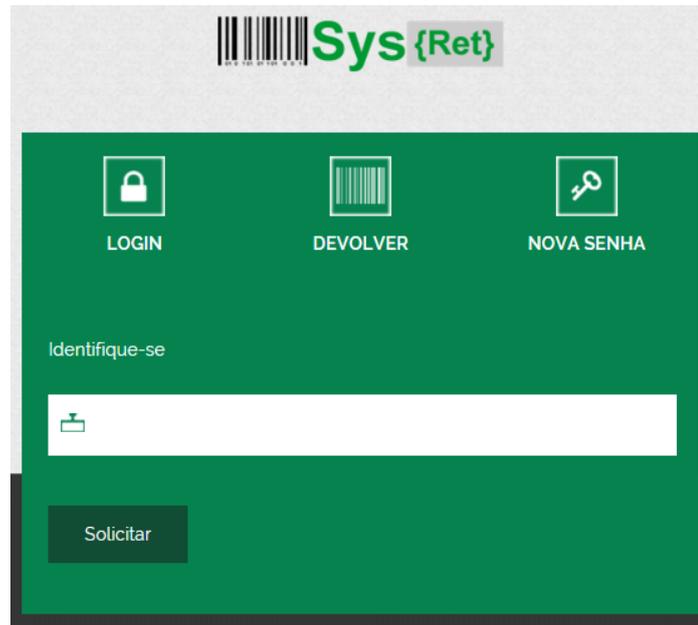


The screenshot shows the 'devolver item' interface for 'Sys {Ret}'. At the top, there is a logo consisting of a barcode and the text 'Sys {Ret}'. Below the logo, there are three main navigation buttons: 'LOGIN' (with a lock icon), 'DEVOLVER' (with a barcode icon), and 'NOVA SENHA' (with a key icon). The central part of the screen contains a white input field for 'Código do Item' and a 'Confirma' button.

Fonte: Do Autor

A tela inicial também conta com a opção de gerar uma nova senha, para isto, o usuário precisa informar o número de sua matricula e o sistema gera uma nova senha que é enviada para o e-mail cadastrado. A figura 28 mostra um exemplo:

Figura 28 - Tela inicial: nova senha

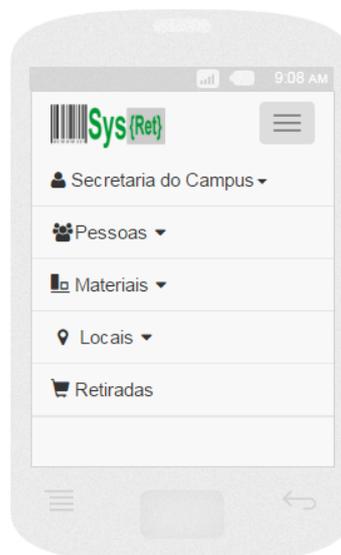


Fonte: Do Autor

4.1.2. Interface da área da secretaria

Como mencionado no capítulo 3 item 3.2.3.3.1 a secretaria é a responsável pelo cadastro de pessoas além dos materiais e locais do *campus*. Os materiais são entregues por ela na recepção onde, ela pode acompanhar as retiradas de cada usuário dentro da sua área.

Figura 29 - Simulação do acesso da área da secretaria em um dispositivo móvel



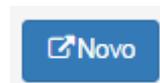
Fonte: Do Autor

A figura 29 acima mostra simulação do acesso a área da secretaria em um aparelho com resolução de 240px. Nesta imagem é possível observar as opções de menus carregados.

4.1.3. Padrões do sistema

Os padrões do sistema referem-se as opções que os usuários têm em relação as interações com aplicação. Como opção de novo onde, o sistema carrega um formulário para cadastro de informações.

Figura 30 - Padrões do sistema: botão novo



Fonte: Do Autor

O botão novo encontra-se no topo da listagem de cada item. Clicando neste botão é carregado um formulário correspondente ao item vinculado. A figura 31 ilustra um exemplo de formulário:

Figura 31 - Padrões do sistema: formulário

Um formulário web para o cadastro de materiais. No topo esquerdo há um botão azul "Voltar" com uma seta para trás. Abaixo dele, o título "Formulário de Materiais" precede uma barra horizontal. Abaixo da barra, há três campos de entrada: "Patrimônio" (campo de texto vazio), "Categoria" (menu suspenso com "Cabos" selecionado) e "Observação" (campo de texto grande). Na base do formulário, há dois botões: "Gravar" em verde e "Limpar" em azul.

Fonte: Do Autor

Este é um exemplo de formulário dos materiais, mas todos os formulários têm funções semelhantes como:

- **Botão voltar:** este botão fica no topo do formulário no canto esquerdo tendo a função de retornar ao item anterior, no caso a listagem dos itens;

- **Caixa de combinação:** a caixa de combinação pode ser carregada com itens definidos estaticamente pelo sistema ou, dinamicamente com no caso de categoria. A categoria é carregada de acordo com os tipos de materiais cadastrados no sistema;
- **Botão gravar:** este botão confirma os dados preenchidos no formulário para que o sistema se encarregue da inserção na base de dados;
- **Botão limpar:** está é uma funcionalidade de limpar o formulário preenchido através do clique neste botão.

Outros dois botões possuem funções essenciais no sistema que aparecem da listagem de cada item como mostra a figura 32 abaixo:

Figura 32 - Padrões do sistema: botões alterar e excluir

Código	Descrição	Permissão	Alterar	Excluir
33	Projektor	3		

Fonte: Do Autor

Estes botões ficam disponível para cada item listado na tabela com as seguintes funções:

- **Botão alterar:** faz a edição do item selecionado. Ao clicar neste botão é carregado o formulário com os campos preenchidos os valores que estão na listagem, o usuário faz a edição do campo que deseja e salva;
- **Botão excluir:** ao clicar neste o botão o item correspondente é excluído da base de dados.

Existem outras funções também disponíveis nas listagens como a ordenação por item como pode ser vista na figura anterior. As setas que aparecem com as descrições dos itens possuem esta funcionalidade, basta clicar no item que deseja ordenar de forma crescente (seta para cima) ou, decrescente (seta para baixo).

Também é possível fazer uma busca no sistema ou mesmo selecionar a quantidade de itens que deseja visualizar por página.

Figura 33 - Padrões do sistema: registro por página e busca

Exibindo 10 ▼ registro por página Buscar

10
25
50
100

Fonte: Do Autor

Para exibir a quantidade de itens basta selecionar a caixa de combinação e escolher uma das opções disponíveis. Por padrão o sistema mostra dez registros por página. Logo ao lado, o usuário pode fazer uma busca pesquisando por um item em: buscar. A paginação é feita de forma dinâmica de acordo com número de listagem por página e a quantidade de itens existentes na base de dados. A figura 34 mostra um exemplo de paginação:

Figura 34 - Padrões do sistema: paginação

Página 1 de 4

Anterior 1 2 3 4 Seguinte

Fonte: Do Autor

Uma mensagem no lado esquerdo mostra em qual página se encontra e quantas páginas existem. No lado direito, é possível fazer a navegação entre páginas clicando no valor correspondente ou, seguir alterando a página nas opções: anterior e próximo.

4.1.4. Validações dos campos editáveis

Os campos editáveis são os campos dos formulários preenchível pelo usuário do sistema. Portanto, a validação é importante para que as informações correspondam corretamente com o previsto. E fazer validações em formulários desenvolvidos usando HTML5 ficou ainda mais fácil pelo simples fato de não precisar utilizar o JavaScript. Agora com os novos atributos: *pattern*, *require* e *type* atribuído com a *tag input*, possui um mecanismo de validação já incorporada na linguagem. Exemplo:

Figura 35 - exemplo de um campo obrigatório

Nome

! Preencha este campo.

Fonte: Do Autor

A figura 35 mostra um exemplo de um campo obrigatório em que, se não estiver preenchido é disparado uma mensagem padrão da própria linguagem HTML5. Para que seja feita a validação, é importante incluir o atributo *required* dentro da *tag input*, como mostra a figura 36 logo abaixo:

Figura 36 - Trecho de código para um campo obrigatório

```
<input name="pes_nome" type="text" value="<?php echo $res->fields['pes_nome'];?>" required="required">
```

Fonte: Do Autor

A linguagem HTML5 incorporou outros tipos de validações para campos de e-mail, data, telefone, campos numéricos entre outros. Veja o exemplo abaixo de um campo de e-mail:

Figura 37 - exemplo de um campo de e-mail

Email

! Inclua um "@" no endereço de e-mail. "thiago" está com um "@" faltando.

Fonte: Do Autor

Neste exemplo da figura 37 é ilustrado a mensagem disparada quando, o campo não é preenchido corretamente como o esperado. Está validação também é bem simples, é preciso selecionar dentro do atributo *type* o dado correspondente ao campo. A figura 38 a seguir, ilustra um exemplo do trecho de código correspondente a imagem acima:

Figura 38 - Trecho de código de um campo e-mail

```
<input name="pes_email" type="email" value="<?php echo $res->fields['pes_email'];?>" required="required">
```

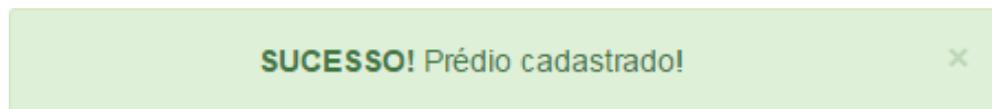
Fonte: Do Autor

4.1.5. Mensagens do sistema

Nesta sessão serão apresentados os dois tipos de testes realizados com o sistema. O primeiro, utilizando a ferramenta Device Mode, e o segundo avaliando os usuários que utilizaram o sistema por um período de tempo.

As mensagens do sistema são formas de avisar ao usuário sobre uma ação executada dentro do sistema. Esta mensagem pode informar se uma determinada ação foi bem-sucedida, mostrando uma mensagem de sucesso como ilustra a figura 39:

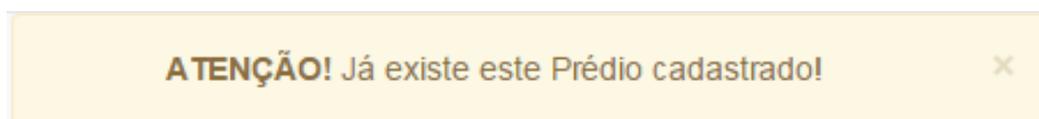
Figura 39 - Padrões do sistema: mensagem de sucesso



Fonte: Do Autor

Também pode ser mostrada uma mensagem de alerta informando que aquela ação não pode ser concluída descrevendo o problema encontrado, como está mensagem de exemplo abaixo:

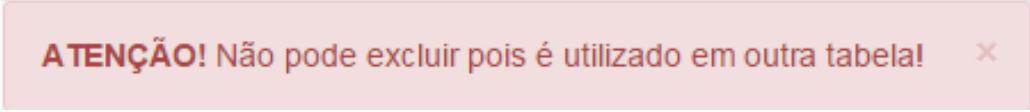
Figura 40 - mensagem de advertência



Fonte: Do Autor

A figura 40 acima traz uma mensagem de alerta informando o problema encontrado que no caso, já possui um dado cadastrado com a mesma descrição. Ao tentar excluir um item, o sistema verifica se aquele item não está relacionado com outro dado, caso afirmativo uma mensagem é disparada, conforme a figura 41:

Figura 41 - mensagem de erro



ATENÇÃO! Não pode excluir pois é utilizado em outra tabela! ✕

Fonte: Do Autor

Existe também uma mensagem que é disparada para o professor quando ele acessa o sistema e vai fazer uma retirada. A mensagem informa a disciplina que ele vai lecionar, a turma e o local de aula.

Figura 42 - mensagem informativa do professor



Olá **Professor 1!** ✕
Hoje é Sexta-Feira
Você ministrará a disciplina **FMC**, na turma **TSPI - 1M1**, na sala **512**

Fonte: Do Auto

4.1.6. Teste com o Device Mode do Google Chrome

O primeiro teste foi com o Device Mode do Google Chrome onde, foi simulado o uso do sistema em diferentes tipos de plataformas e uso de rede. Este teste mostrou-se satisfatório quanto a adaptação do sistema em dispositivos com resoluções de telas menores.

Figura 43 - Teste do sistema no Mode Device do Google Chrome

The screenshot displays the Google Chrome DevTools interface in Device Mode. The main window shows a simulated iPhone 4 screen with a login form titled "Sys {Ret}". The form includes a barcode, a "LOGIN" button, a "DEVOLVER" button, and a "NOVA SENHA" button. Below these are input fields for "Matricula" and a password, a "Lembrar-me" checkbox, and an "Entrar" button. The network panel on the right shows a list of requests with columns for Name Path, Initiator, Size, and Time Latency. A table of network requests is provided below:

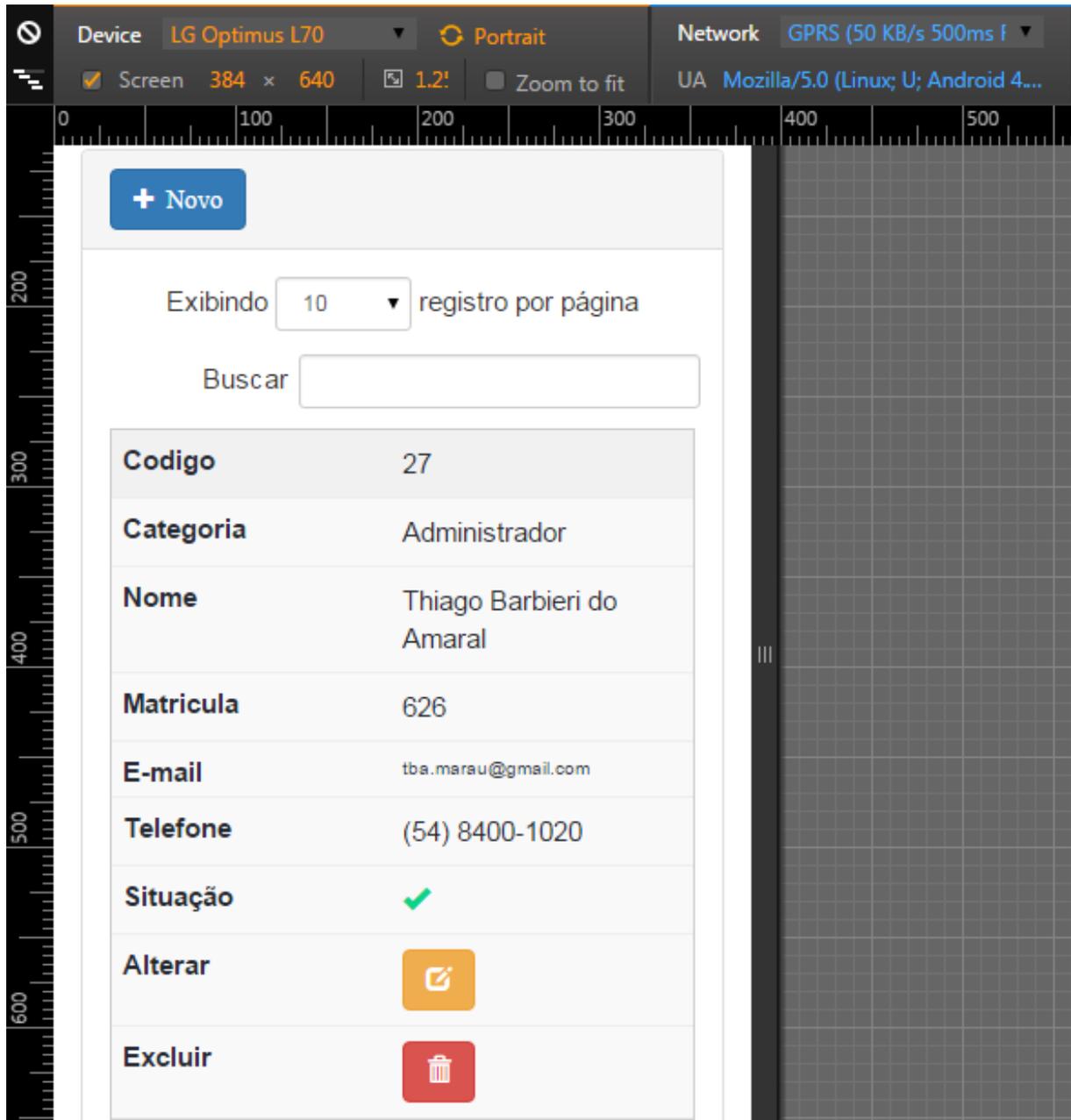
Name Path	Initiator	Size	Time Latency
jq...	[index]:10	2...	1.18 s
/b...	Parser	9...	1.18 s
ea...	[index]:11	2...	1.41 s
/b...	Parser	5...	1.41 s
cs...	[index]:36	1...	1.88 s
fo...	Parser	4...	1.88 s
lo...	[index]:94	2...	1.64 s
/b...	Parser	7...	1.64 s
to...	[index]:101	2...	704 ms
/b...	Parser	3...	703 ms
to...	[index]:110	2...	704 ms
/b...	Parser	3...	703 ms
to...	[index]:119	2...	938 ms
/b...	Parser	3...	937 ms

Summary statistics at the bottom of the network panel: 15 requests | 13.3 KB transferred | Finish: 5.48 s | DOM

Fonte: Do Autor

Neste exemplo da figura 43 mostra as opções de testes que a ferramenta disponibiliza como a escolha do aparelho para simulação. No menu *device* é feita a escolha do aparelho podendo, também, inverter a posição. Na opção *network* do menu ao lado do *device*, pode-se escolher o tipo de rede de internet para o teste. Este teste de escolha de rede é visualizado na aba da direita da ferramenta, clicando na opção *network*. Esta opção mostra uma linha com o tempo de requisição do sistema e, abaixo, os arquivos carregados.

Figura 44 - exemplo de listagem de pessoas



Fonte: Do Autor

A ilustração da figura 44 mostra um exemplo de avaliação aplicada no sistema utilizando um dispositivo com largura de 384 *pixel*. Neste exemplo, foi avaliado a listagem de dados da tabela pessoa onde, as linhas da tabela alinharam uma abaixo da outra, acompanhada com o cabeçalho de cada informação na coluna à esquerda. Isto foi um resultado obtido com a elaboração deste projeto, pois, o Bootstrap não oferece este mecanismo que foi aplicado. Portanto, foi criada uma *media query* afim de solucionar este problema, como mostra o quadro abaixo:

Figura 45 - media querie desenvolvida para a listagem dos dados da tabela de pessoas

```

1 <style type="text/css">
2 @media only screen and (max-width: 760px),(min-device-width: 768px) and (max-device-width: 1024px) {
3
4 table, thead, tbody, th, td, tr { display: block;}
5
6 thead tfoot tr { position: absolute; top: -9999px; left: -9999px;}
7
8 td {border: none; border-bottom: 1px solid #eee; position: relative;padding-left: 50% !important; }
9
10 td:before {position: absolute; top: 6px; left: 6px; width: 45%; padding-right: 10px; white-space: nowrap;}
11 td:nth-of-type(1):before { content: "Codigo"; }
12 td:nth-of-type(2):before { content: "Categoria"; }
13 td:nth-of-type(3):before { content: "Nome"; }
14 td:nth-of-type(4):before { content: "Matricula"; }
15 td:nth-of-type(5):before { content: "E-mail"; }
16 td:nth-of-type(6):before { content: "Telefone"; }
17 td:nth-of-type(7):before { content: "Situação"; }
18 td:nth-of-type(8):before { content: "Alterar"; }
19 td:nth-of-type(9):before { content: "Excluir"; }
20 td:before {font-size: 15px;font-weight: bold;}
21 #colEmail {font-size: 10px;}
22 }
23 </style>

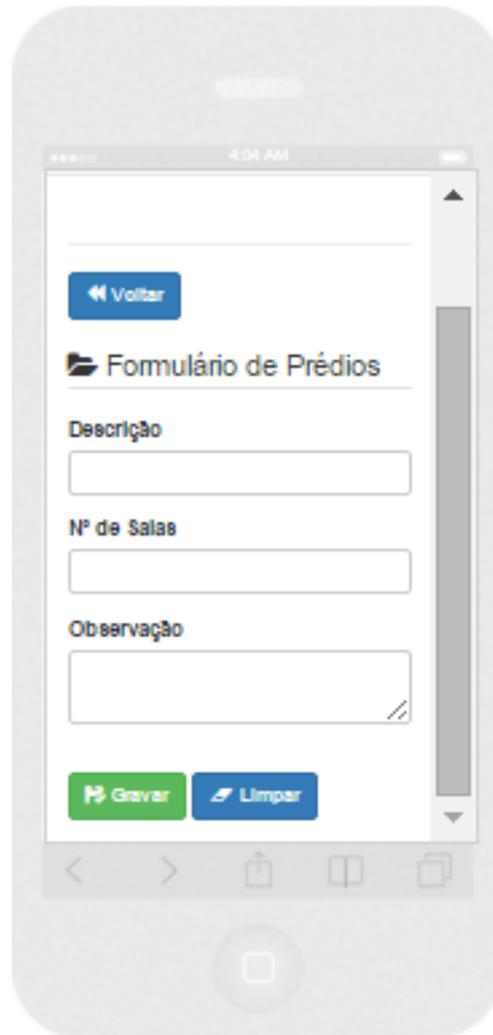
```

Fonte: Do Autor

Primeiramente, foi testado o *script* verificando se, o tamanho da resolução da tela do dispositivo é menor do que o tamanho real desta tabela, 1024 *pixels*. Caso seja menor, é aplicado um novo padrão de estilo na tabela. Então os elementos da tabela são quebrados em blocos, usando o valor da propriedade `display: block` dentro dos seletores das *tags* `table`, `thead`, `tbody`, `th`, `td` e `tr`. Na linha 8, o código esconde as propriedades que correspondem a cabeçalho da tabela e o rodapé. Então a partir da linha 10 até 19 é aplicado valores que apareceram no lado esquerdo de cada listagem estaticamente.

Os formulários obtiveram resultados positivos uma vez que, foi desenvolvido seguindo os padrões do Bootstrap. A figura 46 demonstra a ilustração do exemplo de teste realizado:

Figura 46 - formulário de prédios

A smartphone mockup displaying a mobile application interface. At the top, there is a status bar with signal strength, the time '4:04 AM', and battery level. Below the status bar is a blue button with a left-pointing arrow and the text 'Voltar'. The main title of the form is 'Formulário de Prédios' with a folder icon to its left. There are three input fields: 'Descrição' (a large text area), 'Nº de Salas' (a single-line text field), and 'Observação' (a large text area with a small icon in the bottom right corner). At the bottom of the form, there are two buttons: a green one with a save icon and the text 'Gravar', and a blue one with an eraser icon and the text 'Limpar'. The bottom of the screen shows a standard Android navigation bar with back, home, and recent apps icons.

Fonte: Do Autor

Para realização deste teste foi utilizado a ferramenta Responsinator essencialmente para ilustração do exemplo. Como pode ser observado, o formulário se adapta corretamente com a largura da tela do dispositivo. A largura da tela do dispositivo móvel utilizado é de 320 *pixel*.

Os testes são realizados durante o desenvolvimento do projeto a fim de encontrar erros na execução do sistema quanto, a visualização da aplicação em diferentes tipos de resoluções de telas. Assim, as correções são feitas de maneira a obter resultados satisfatórios.

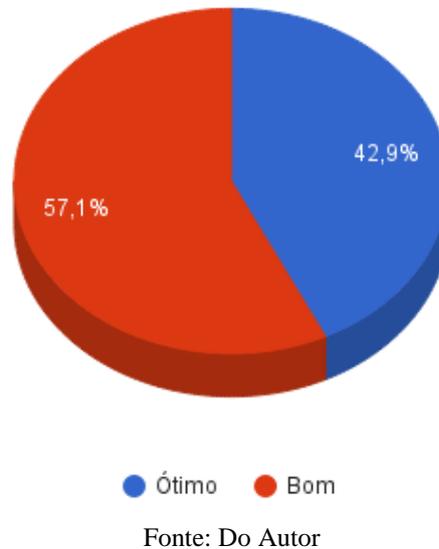
4.1.7. Teste com os usuários

O sistema foi implantado no IFSUL campus Passo Fundo e ficou disponível alguns dias para que, fossem feitos testes diretamente com alguns usuários selecionados. Estes

usuários foram convidados a utilizar o sistema e, depois foi aplicado um questionário com algumas perguntas.

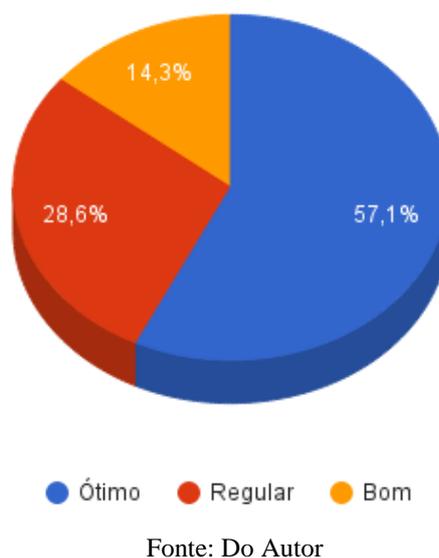
A primeira pergunta foi em relação as funcionalidades do sistema onde obteve as seguintes respostas como mostra o gráfico abaixo:

Figura 47 - Gráfico sobre as funcionalidades do sistema



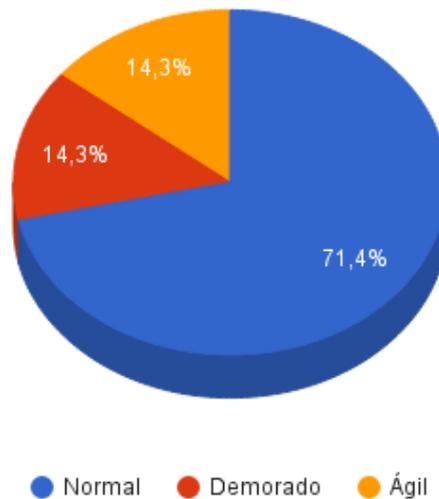
Mais de 57 por cento dos que avaliaram o sistema consideraram as funcionalidades do sistema ótimo e, o restante, consideraram bom. Outras três respostas, regular, ruim ou péssimo não obtiveram pontuação. Foi questionado também, a agilidade do sistema e como os usuários avaliavam este item obtendo as seguintes considerações:

Figura 48 - Gráfico sobre a agilidade do sistema desenvolvido



A maioria cerca de 57 por cento consideraram as funcionalidades do sistema ótimo sendo, o restante, 28,6% regular e 14,3% bom. Também nesta pergunta, outras respostas: ruim e péssimo não pontuaram. Em relação ao tempo de acesso ao sistema para efetuar a retirada do material, a maioria dos usuários avaliaram com normal, como mostrado a seguir:

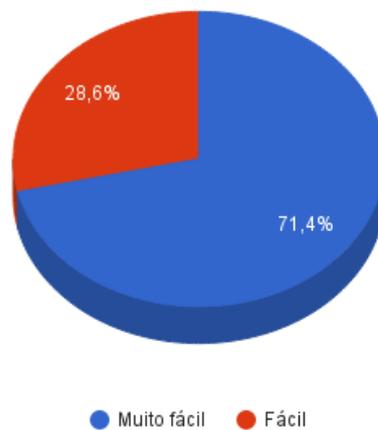
Figura 49 - Gráfico sobre o tempo de acesso ao sistema



Fonte: Do Autor

A figura 49 mostra que duas repostas obtiveram a mesma pontuação cerca de 14,3 por cento cada uma e tendo uma opção de resposta: muito demorado não obteve pontuação. Está pergunta foi feito com intuito de verificar se o sistema agilizaria o processo de empréstimos, que até então, é feita de forma manual.

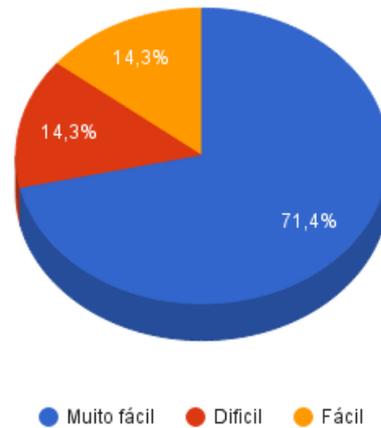
Figura 50 - Gráfico sobre a utilização do sistema



Fonte: Do Autor

A figura 50 ilustra o gráfico com as respostas quando perguntado se na opinião deles o sistema era fácil aprender a utilizar. Mais de 70 por cento consideraram muito fácil e, 28,6% fácil. Esta pergunta também teve opções que não pontuaram. Quando perguntado sobre a navegação:

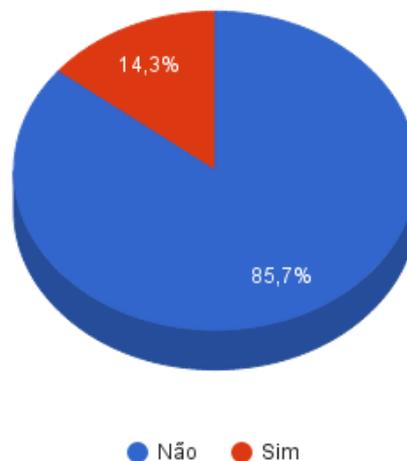
Figura 51 - Gráfico sobre a navegação no sistema



Fonte: Do Autor

Conforme a figura 51 a maioria cerca de 71,4% consideram a navegação muito fácil e, empatados com 14,3% consideram fácil ou difícil. Por ser um sistema responsivo, foi questionado se eles acessaram o sistema por algum tipo de dispositivo móvel, e obteve as seguintes respostas como mostra a figura 52:

Figura 52 - Gráfico sobre o acesso ao sistema utilizando dispositivo móvel



Fonte: Do Autor

Apenas 14,3% fez o acesso meio de um dispositivo móvel. Estas foram as perguntas de múltiplas escolhas onde o usuário assinalava uma opção por pergunta.

A última pergunta foi para que deixassem uma opinião/sugestão em relação ao sistema. Alguns usuários sugeriram que o processo de retirada fosse mais fácil, que apresentasse a página de retirada logo que o usuário faz a autenticação no sistema. A maioria considerou o processo de devolução simples, pois não precisa fazer a autenticação no sistema.

O questionário foi enviado para 13 pessoas tendo 6 abstenções. As perguntas com as respostas podem ser visualizadas na tabela 2 abaixo:

Tabela 2 - Perguntas com as repostas do questionário aplicado aos usuários

Perguntas	Opção	Resposta
1 - Em relação as funcionalidades do sistema, você considera:	Ótimo	3
	Bom	4
	Regular	0
	Ruim	0
	Péssimo	0
2 – Como você avalia a agilidade do sistema:	Ótimo	4
	Bom	1
	Regular	2
	Ruim	0
	Péssimo	0
3 - Como você avalia o tempo de acesso ao sistema para efetuar a retirada/empréstimo de material	Muito demorado	0
	Demorado	1
	Normal	5
	Ágil	1
4 - Na sua opinião, foi fácil aprender a utilizar o sistema? Faça a classificação de 1 – 5, sendo 1 muito difícil e 4 muito fácil.	1	0
	2	0
	3	0
	4	2
	5	5

5 - A navegação no sistema é:	1	0
Faça a classificação de 1 - 5, sendo 1 muito difícil e 4 muito fácil.	2	0
	3	1
	4	1
	5	5
6 - Você fez alguma consulta do sistema usando algum tipo de dispositivo móvel?	Sim	6
	Não	1

Fonte: Do Autor

5. CONSIDERAÇÕES FINAIS

Desenvolver um sistema para o ambiente web e que seja acessível independentemente do navegador ou resolução de tela é desafiador. Todavia o repto é superado com a disponibilização de frameworks voltados para a geração de aplicações responsivas, que ajudam a aligeirar o processo de desenvolvimento. Para a realização de testes, também, pode-se contar com ferramentas que simulam vários tipos de aparelhos dentro de sua interface.

As dificuldades deste trabalho foram em projetar a aplicação de uma maneira que, disponibilizasse os mesmos recursos independentemente do dispositivo que a estivesse acessando. A listagem de dados resultou em um estudo fora dos recursos disponíveis no framework utilizado, no caso, o Bootstrap. Conseguir fazer com que as tabelas ficassem fluidas foi um dado obtido graças a esse trabalho, pois, nenhum dos frameworks analisados aqui, nas versões apresentadas, possuem tabelas fluidas, todos eles, criam uma barra de navegação abaixo da tabela onde, o usuário precisa deslizar lateralmente para acompanhar as informações obtidas.

Vale ressaltar que os objetivos deste trabalho foram todos alcançados. Foi concretizado uma pesquisa bibliográfica com o tema em foco: web design responsivo. Na medida em que a pesquisa abordou inicialmente o usuário móvel, pode-se perceber a importância no desenvolvimento de aplicações que atendam este público, uma vez que, as pessoas estão cada vez mais conectadas com a internet por diferentes tipos de dispositivos, e passam cada vez mais tempo acessando à internet com os dispositivos moveis. A pesquisa abordou as linguagens utilizadas no desenvolvimento da aplicação.

O sistema foi modelado de acordo com os requisitos levantados através de entrevistas com os agentes envolvidos no processo de empréstimos de materiais do *Campus*. Isto possibilitou o desenvolvimento do projeto e, posteriormente, implantar o sistema por um período de teste. Os testes serviram para avaliar como este estudo pode melhorar o processo, que até então, é feito manualmente sem o auxílio tecnológico. O sistema ajuda a gerenciar o controle empréstimos e diminui os gastos com folhas de ofício.

Finalizando, este trabalho pode ser continuado realizando trabalhos futuros como:

- Criar um mecanismo para gerar relatório específicos de acordo com o item selecionado;

- Permitir que o usuário possa fazer uma reserva de local/material;
- Facilitar o processo de retirada;
- Importar os horários dos professores direto do sistema acadêmico;
- Realizar um estudo a partir dos resultados obtidos.

6. REFERÊNCIAS

AMADOR, João Gabriel. Estudo mostra crescimento no uso de dispositivos móveis e domínio Android. Disponível em: <http://www.correiobraziliense.com.br/app/noticia/tecnologia/2015/01/17/interna_tecnologia,466691/estudo-mostra-crescimento-no-uso-de-dispositivos-moveis-e-dominio-andr.shtml>.

Acesso em: 22. mar. 2015.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. Rio de Janeiro, RJ: Campus, 2012.

BOUVIÉR, Wilian. **Sistema web mobile para controle de eventos**. Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense – IFSUL, Passo Fundo, 2012.

CASTRO, Elizabeth; HYSLOP, Bruce. **HTML5 e CSS3**. 7. ed. Rio de Janeiro, RJ: Alta Books, 2013.

DATAFOLHA. **43 milhões de brasileiros acessam internet por dispositivos móveis**. Disponível em: <<http://datafolha.folha.uol.com.br/mercado/2014/01/1400618-43-milhoes-de-brasileiros-acessam-internet-por-dispositivos-moveis.shtml>>. Acesso em: 27. out. 2014.

DRAKE, Joshua; WORSLEY, John. **Practical PostgreSQL**. Sebastopol: O’Reilly Media, Inc., 2011. Disponível em: . Acesso em: 15 mai. 2015.

EIS, Diego. **OOCSS ou CSS do jeito certo**. Disponível em: <<http://tableless.com.br/oocss-ou-css-do-jeito-certo/>>. Acesso em: 26. out. 2014.

EIS, Diego. **Introdução sobre media queries**. Disponível em: <<http://tableless.com.br/introducao-sobre-media-queries/>>. Acesso em: 26. out. 2014.

FERREIRA, Elcio.; EIS, Diego. **HTML5: Curso W3C escritório Brasil**. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 08. mar. 2015.

FLANAGAN, David. **JavaScript: O guia definitivo**. 6. ed. Porto Alegre, RS: Bookman, 2013.

GARDNER, Lysa Danger.; GRIGSBY, Jason. **Use a cabeça: mobile web**. Rio de Janeiro, RJ: Alta Books, 2013.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo, SP: Atlas, 2002.

GUEDES, Gilleanes T. A. **UML2: Uma abordagem prática**. 2. ed. São Paulo, SP: Novatec, 2011.

HOPMA, Tim. **The real world guide to responsive design**. Disponível em: <<http://www.zazzlemedia.co.uk/blog/responsive-content-design/>>. Acessado em: 04. set. 2014.

MALCHER, José Stélio Jr. **Guia de referência do PHP**. Disponível em: <<https://goo.gl/XjXsvz>>. Acesso em: 04. abr. 2015.

NETO, Álvaro Pereira. **PostgreSQL: Técnicas avançadas: Versões open source 7.x e 8.x: Soluções para o desenvolvimento e administração de banco de dados**. 4. ed. São Paulo, SP: Érica, 2007.

RAMOS, Ricardo, SILVA, Joel e AFONSO, Ricardo. **PHP para profissionais**. São Paulo, SP: Digerati Books, 2007.

SALES, Robson. **IBGE: Mais de 50% usam celular e tablet para acessar a internet**. Disponível em: <<http://www.valor.com.br/brasil/4027294/ibge-mais-de-50-usam-celular-e-tablet-para-acessar-internet>>. Acesso em: 14. set. 2015.

SILVA, Maurício Samy. **Foundation for sites**: Aprenda a criar páginas web com uso do framework responsivo mais avançado do mundo. São Paulo, SP: Novatec, 2014.

SILVA, Maurício Samy. **jQuery Mobile**: Desenvolva aplicações web para dispositivos móveis com HTML5, CSS3, AJAX, jQuery e jQuery UI. 2. ed. São Paulo, SP: Novatec, 2013.

SILVA, Maurício Samy. **HTML5**: A linguagem de marcação que revolucionou a web. São Paulo, SP: Novatec, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo, SP: Person Addison-Wesley, 2007.

SILVA, Mauricio Samy. **Linha do gradiente**. Disponível em: < <http://www.maujor.com/tutorial/css3-gradientes-lineares.php> >. Acesso em: 10. set. 2014.

SILVA, Mauricio Samy. **jQuery**: A biblioteca do programador JavaScript. 2. ed. São Paulo, SP: Novatec Editora, 2010.

WELLING, Luke.; THOMSON, Laura. **PHP e MySQL** desenvolvimento web. 3. ed. Rio de Janeiro, RJ: Elsevier, 2005.

ZEMEL, Tércio. **Web Design Responsivo**: Páginas Adaptáveis para Todos os Dispositivos. São Paulo, SP: Casa do Código, 2014.

Ações do Ator	Ações do Sistema
1 – Selecionar na página inicial a opção Login	
	2 – Apresentar a tela de login
3 – Informar login e senha	
	4 – Abrir a área correspondente da Secretária no Sistema
Restrições/Validações	1 – Autenticar login e senha, caso esteja errado, voltar para o item 3 em Ações do Ator

Nome do Caso de Uso	Manter Tipos de Materiais
Ator principal	Secretaria
Atores secundários	
Resumo	Este modelo detalha os passos para que a Secretaria possa adicionar um novo tipo de matéria (projektor, notebook, caixa de som, etc), e também, editar e excluir
Pré-Condições	A Secretaria deve estar cadastrada na base de dados para realizar as ações no sistema
Pós-Condições	
Fluxo Principal – Adicionar Tipo de Material	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Tipo	
	2 – Carregar a tela de listagem com os Tipos, se existir alguma cadastrada, com as opções de: novo, editar e excluir
3 – Selecionar a opção novo	
	4 – Carregar a tela com o formulário de cadastro com o campo a ser preenchido: permissão e descrição
5 – Preencher os campos	
6 – Confirmar pressionando o botão gravar	

Restrições/Validações	7 – Fazer a persistência dos dados na base de dados
	1 – Os campos permissão e descrição são do tipo obrigatório
	2 – O campo código não é editável e, é gerado sequencialmente pela base de dados
Fluxo Alternativo I – Editar Tipo de Material	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Tipo, a linha correspondente que deseja editar e pressionar o botão editar	
	2 – Carregar o formulário com o campo desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Fazer a alteração	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer o update na base de dados
	7 – Retornar para a tela de listagem de Categorias
Restrições/Validações	As restrições de validações do Fluxo Alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Tipo de Material	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Tipo a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela

	de Material, caso esteja, mostrar uma mensagem para a Secretaria informando a situação
--	--

Nome do Caso de Uso	Manter Materiais
Caso de Uso Geral	
Ator principal	Secretária
Atores Secundários	
Resumo	Este modelo representa as ações referentes aos Materiais onde, a Secretária poderá: adicionar, editar ou excluir
Pré-Condições	A Secretária deve estar cadastrada na base de dados
Pós-Condições	
Fluxo Principal – Adicionar Material	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Material	
	2 – Mostra a tela de Materiais listando-os junto com os botões de opções: adicionar, editar, excluir e cancelar
3 – Selecionar a opção adicionar	
	4 - Mostrar a tela com os campos do Material: Patrimônio, Observação e Tipo
5 – Preencher os campos	
6 – Confirmar pressionando o botão ok	
	7 – Fazer a persistência dos dados na base de dados
Restrições/Validações	1 – Os campos: Patrimônio e Tipo são obrigatórios
	2 – Verificar se o campo patrimônio foi digitado apenas números
	3 – O campo Código não é editável e, é

	gerado sequencialmente pela base de dados
	4 – Se o botão cancelar for selecionado, carregar a tela inicial desta manutenção
Fluxo Alternativo I – Editar Material	
Ações do Ator	Ações do Sistema
1 - Selecionar na listagem do Material, a linha correspondente que deseja editar e pressionar o botão editar	
	2 – Carregar o formulário com os campos desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Altera os dados necessários	
4 – Confirmar a edição pressionando o botão salvar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer um update na base de dados
	7 – Carregar a listagem da manutenção correspondente
Restrições/Validações	1 – As restrições de validações do fluxo alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Material	
Ações do Ator	Ações do Sistema
1 - Selecionar na listagem do Material, a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela retiradas, caso esteja, mostrar uma mensagem para o usuário informando a situação

Nome do Caso de Uso	Manter Categorias
Ator principal	Secretaria
Atores secundários	
Resumo	Este modelo representa as ações referentes as categorias dos tipos de pessoas (secretaria, professor, coordenador, funcionário administrativo, etc), onde, a Secretaria poderá: adicionar uma nova categoria, editar ou excluir a categoria selecionada
Pré-Condições	A Secretaria deve estar cadastrada na base de dados para realizar as ações no sistema
Pós-Condições	
Fluxo Principal – Adicionar Categoria	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção categoria	
	2 – Carregar a tela de listagem com as categorias, se existir alguma cadastrada, com as opções de: novo, editar e excluir
3 – Selecionar a opção novo	
	4 – Carregar a tela com o formulário de cadastro com o campo a ser preenchido: descrição
5 – Preencher o campo	
6 – Confirmar pressionando o botão gravar	
Restrições/Validações	7 – Fazer a persistência dos dados na base de dados
	1 – O campo descrição é do tipo obrigatório
	2 – O campo código não é editável e, é gerado sequencialmente pela base de dados
Fluxo Alternativo I – Editar Categoria	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Categoria, a linha correspondente que deseja editar e	

pressionar o botão editar	
	2 – Carregar o formulário com o campo desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Fazer a alteração	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer o update na base de dados
	7 – Retornar para a tela de listagem de Categorias
Restrições/Validações	As restrições de validações do Fluxo Alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Categoria	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Categoria a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela de Pessoas, caso esteja, mostrar uma mensagem para a Secretaria informando a situação

Nome do Caso de Uso	Manter Pessoas
Caso de Uso Geral	
Ator Principal	Secretária

Atores Secundários	Professor, Coordenador,
Resumo	Este caso de uso detalha os passos para que a Secretária possa incluir, editar, excluir ou emitir relatório das Pessoas
Pré-Condições	A Secretária precisa estar registrada na base de dados
Pós-Condições	
Fluxo Principal - Adicionar Pessoa	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Pessoa	
	2 – Carregar todos os dados da Pessoa com as opções: adicionar, editar, excluir e cancelar
3 – Selecionar a opção novo	
	4 – Mostrar a tela com os campos: código, nome, matrícula, e-mail, telefone, categoria, situação e senha
5 – Preencher todos campos	
6 – Confirmar pressionando o botão ok	
	7 – Fazer a persistência na base de dados
Restrições/Validações	1 – Os campos: nome, matrícula, e-mail, telefone, categoria, situação e senha são obrigatórios
	2 – O campo nome só pode ter letras
	3 – Os campos: matrícula e telefone só pode ter dados numéricos
	4 – O campo código é não editável sendo gerado pela base de dados
	5 – O campo senha tem que ser criptografado antes de ser persistido na base de dados
Fluxo Alternativo II – Editar Pessoa	
Ações do Ator	Ações do Sistema

1 – Selecionar na listagem de Pessoa a linha correspondente que deseja editar	
	2 – Carregar na tela os campos com os dados selecionado para edição
3 – Alterar os dados necessários	
4 – Confirmar a edição pressionando o botão salvar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer um update na base de dados
	7 – Carregar a listagem da manutenção correspondente
Restrições/Validações	1 - As restrições e validações são os mesmos do fluxo principal.
	2 – O campo senha não poderá ser editado na área da Secretária, somente na área correspondente do usuário
Fluxo Alternativo III – Excluir Pessoa	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem de pessoas a linha correspondente que deseja excluir	
2 – Confirmar a exclusão	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se o código deste item não está sendo utilizados nas tabelas: Retiradas e Horários, caso esteja, mostrar uma mensagem para o usuário informando a situação

Nome do Caso de Uso	Manter Prédios
Ator principal	Secretaria
Atores secundários	

Resumo	Este modelo detalha os passos para a Secretaria possa adicionar um novo Prédio, também, editar e excluir
Pré-Condições	A Secretaria deve estar cadastrada na base de dados para realizar as ações no sistema
Pós-Condições	
Fluxo Principal – Adicionar Categoria	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção categoria	
	2 – Carregar a tela de listagem com os prédios, se existir alguma cadastrada, com as opções de: novo, editar e excluir
3 – Selecionar a opção novo	
	4 – Carregar a tela com o formulário de cadastro com o campo a ser preenchido: descrição, número de salas e observação
5 – Preencher os campos	
6 – Confirmar pressionando o botão gravar	
Restrições/Validações	7 – Fazer a persistência dos dados na base de dados
	1 – Os campos descrição e número de salas é do tipo obrigatório
	2 – O campo código não é editável e, é gerado sequencialmente pela base de dados
Fluxo Alternativo I – Editar Prédio	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem de Prédio, a linha correspondente que deseja editar e pressionar o botão editar	
	2 – Carregar o formulário com o campo desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição

3 – Fazer a alteração	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer o update na base de dados
	7 – Retornar para a tela de listagem de Categorias
Restrições/Validações	As restrições de validações do Fluxo Alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Prédio	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem de Prédios a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela de locais, caso esteja, mostrar uma mensagem para o usuário informando a situação

Nome do Caso de Uso	Manter Locais
Caso de Uso Geral	
Ator Principal	Secretária
Atores Secundários	
Resumo	Este caso de uso detalha os passos para que a Secretária possa incluir, editar, excluir.
Pré-Condições	A Secretária deve estar cadastrada na base de dados
Pós-Condições	

Fluxo Principal – Adicionar Local	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Local	
	2 – Carregar todos os dados do Local com as opções: adicionar, editar, excluir e cancelar
3 – Selecionar a opção novo	
	4 – Mostrar a tela com os campos: Código, Sala, Capacidade e Prédio
5 – Preencher todos os campos	
6 – Confirmar pressionando o botão gravar	
	7 – Fazer a persistência na base de dados
Restrições/Validações	1 – Os campos: Sala, Capacidade e Prédio são obrigatórios
	2 – O campo Capacidade só poderá ser digitado apenas números
	3 – O campo código não é editável
Fluxo Alternativo I – Editar Local	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem de Local a linha correspondente que deseja editar	
	2 - Carregar o formulário com os campos desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Alterar os dados necessários	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer update na base de dados
	7 – Carregar a listagem da manutenção correspondente

Restrições/Validações	1 – As restrições de validações do fluxo alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Local	
Ações do Ator	Ações do Sistema
1 - Selecionar na listagem do Local, a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado nas tabelas: retiradas e horários, caso esteja, mostrar uma mensagem para o usuário informando a situação

Apêndice B – Documentação do Caso de Uso: Professor

Nome do Caso de Uso	Fazer Login
Caso de Uso Geral	
Ator Principal	Professor
Atores Secundários	
Resumo	Descreve as etapas necessárias para realização do login no Sistema
Pré-Condições	Acessar a página do Sistema
Pós-Condições	Precisa estar cadastrado na base de dados
Fluxo Principal	
Ações do Ator	Ações do Sistema
Idênticos ao Caso de Uso – Secretária (Fazer Login), exceto que, o usuário tem que selecionar o ícone: Professor	Idênticos ao Caso de Uso – Secretária (Fazer Login), exceto que, quando o usuário Professor logar no Sistema, apresentar a área correspondente ao mesmo
Restrições/Validações	

Nome do Caso de Uso	Efetuar Retirada/Devolução
Caso de Uso Geral	
Ator Principal	Professor
Atores Secundários	
Resumo	Descreve as etapas necessárias para a retirada de materiais
Pré-Condições	Fazer login no Sistema Ter permissão
Pós-Condições	
Fluxo Principal – Fazer retiradas	
1 – Selecionar a opção retirada	
	2 – Carregar a tela de retiradas com os campos: data, hora, motivo e observação
3 – Preencher os dados do formulário	
4 – Confirmar pressionando em ok	
	5 – Mostrar uma mensagem para o usuário de confirmação
	6 – Persistir as informações na base de dados
Restrições/Validações	1 – Os campos: data, hora e motivo são obrigatórios
	2 – O campo código é gerado sequencialmente pela base de dados e não pode ser editável
Fluxo alternativo – Fazer devolução	
Ações do ator	Ações do sistema
1 – Passar o leitor de códigos de barras sobre o código de barras do material devolvido	
	3 - Mostrar uma mensagem de confirmação de devolução com o nome do usuário data e hora devolvidos

	2 – Persistir as informações: data devolução e hora da devolução na base de dados
Restrições/Validações	1 – Precisa informar o código do material

Nome do Caso de Uso	Manter Meus Dados
Caso de Uso Geral	
Ator Principal	Professor
Atores Secundários	Secretaria
Resumo	
Pré-Condições	O Professor deve estar cadastrado na base de dados
Pós-Condições	Dados editados
Fluxo Principal – Editar	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem Meus Dados a opção editar	
	2 – Carregar o formulário com os desta manutenção preenchidos com os valores correspondentes ao registro possibilitando o registro
3 – Alterar os dados necessários	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer update na base de dados
	7 – Carregar a listagem da manutenção correspondente
Restrições/Validações	1 – Todos os campos são obrigatórios
	2 - O campo telefone só pode ser digitado número
	3 – O campo e-mail tem que ser correspondente ao formato

Apêndice C – Documentação do Caso de Uso: Coordenador

Nome do Caso de Uso	Fazer Login
Caso de Uso Geral	
Ator Principal	Coordenador
Atores Secundários	
Resumo	Descreve as etapas necessárias para realização do login no Sistema
Pré-Condições	Acessar a página do Sistema
Pós-Condições	Precisa estar cadastrado na base de dados
Fluxo Principal	
Ações do Ator	Ações do Sistema
Idênticos ao Caso de Uso – Secretária (Fazer Login), exceto que, o usuário tem que selecionar o ícone: Coordenador	Idênticos ao Caso de Uso – Secretária (Fazer Login), exceto que, quando o usuário Coordenador logar no Sistema, apresentar a área correspondente ao mesmo
Restrições/Validações	

Nome do Caso de Uso	Manter Disciplinas
Ator principal	Coordenador
Atores secundários	Professor
Resumo	Este caso de uso detalha os passos para que o Coordenador possa adicionar uma nova disciplina e, também, editar e excluir
Pré-Condições	O Coordenador deve estar cadastrada na base de dados para realizar as ações no sistema
Pós-Condições	Adicionado, editado ou excluído a disciplina
Fluxo Principal – Adicionar Disciplina	
Ações do Ator	Ações do Sistema

1 – Selecionar a opção disciplina	
	2 – Carregar a tela de listagem com as disciplinas com as opções de: novo, editar e excluir
3 – Selecionar a opção novo	
	4 – Carregar a tela com o formulário de cadastro com o campo a ser preenchido: descrição e abreviatura
5 – Preencher os campos	
6 – Confirmar pressionando o botão gravar	
Restrições/Validações	7 – Fazer a persistência dos dados na base de dados
	1 – O campos descrição e abreviatura são do tipo obrigatório
	2 – O campo código não é editável e, é gerado sequencialmente pela base de dados
Fluxo Alternativo I – Editar Disciplina	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Disciplina, a linha correspondente que deseja editar e pressionar o botão editar	
	2 – Carregar o formulário com o campo desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Fazer a alteração	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer o update na base de dados
	7 – Retornar para a tela de listagem de Disciplinas

Restrições/Validações	As restrições de validações do Fluxo Alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Disciplina	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Disciplina a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela de Horários, caso esteja, mostrar uma mensagem para a Secretaria informando a situação

Nome do Caso de Uso	Manter Turmas
Ator principal	Coordenador
Atores secundários	Professor
Resumo	Este caso de uso detalha os passos para que, o Coordenador possa adicionar uma nova turma e, também, editar e excluir
Pré-Condições	O Coordenador deve estar cadastrada na base de dados para realizar as ações no sistema
Pós-Condições	Adicionado, editado ou excluído a turma
Fluxo Principal – Adicionar Turma	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Turma	
	2 – Carregar a tela de listagem com as Turmas com as opções de: novo, editar e excluir
3 – Selecionar a opção novo	
	4 – Carregar a tela com o formulário de

	cadastro com o campo a ser preenchido: descrição
5 – Preencher o campo	
6 – Confirmar pressionando o botão gravar	
Restrições/Validações	7 – Fazer a persistência dos dados na base de dados
	1 – O campo descrição é do tipo obrigatório
	2 – O campo código não é editável e, é gerado sequencialmente pela base de dados
Fluxo Alternativo I – Editar Turma	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Turma, a linha correspondente que deseja editar e pressionar o botão editar	
	2 – Carregar o formulário com o campo desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Fazer a alteração	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer o update na base de dados
	7 – Retornar para a tela de listagem de Categorias
Restrições/Validações	As restrições de validações do Fluxo Alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Turma	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem da Turma a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	

	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela de Horários, caso esteja, mostrar uma mensagem para a Secretaria informando a situação

Nome do Caso de Uso	Manter Cursos
Ator principal	Coordenador
Atores secundários	Professor
Resumo	Este caso de uso detalha os passos para que, o Coordenador possa adicionar um novo Curso e, também, editar e excluir
Pré-Condições	O Coordenador deve estar cadastrada na base de dados para realizar as ações no sistema
Pós-Condições	Adicionado, editado ou excluído o Curso
Fluxo Principal – Adicionar Curso	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Curso	
	2 – Carregar a tela de listagem com as opções de: novo, editar e excluir
3 – Selecionar a opção novo	
	4 – Carregar a tela com o formulário de cadastro com o campo a ser preenchido: descrição
5 – Preencher o campo	
6 – Confirmar pressionando o botão gravar	
Restrições/Validações	7 – Fazer a persistência dos dados na base de dados
	1 – O campo descrição é do tipo obrigatório
	2 – O campo código não é editável e, é

	gerado sequencialmente pela base de dados
Fluxo Alternativo I – Editar Curso	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem do Curso, a linha correspondente que deseja editar e pressionar o botão editar	
	2 – Carregar o formulário com o campo desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Fazer a alteração	
4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação
	6 – Fazer o update na base de dados
	7 – Retornar para a tela de listagem do Curso
Restrições/Validações	As restrições de validações do Fluxo Alternativo I é o mesmo do fluxo principal
Fluxo Alternativo II – Excluir Curso	
Ações do Ator	Ações do Sistema
1 – Selecionar na listagem do Curso a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	1 – O sistema deverá verificar se, o código deste item não está sendo utilizado na tabela de Horários, caso esteja, mostrar uma mensagem para a Secretaria informando a situação

Nome do Caso de Uso	Manter Horários
Ator Principal	Coordenador
Atores Secundários	Professor
Resumo	Descreve as etapas para que o Coordenador do curso possa cadastrar os horários de cada professor
Pré-Condições	Precisa estar logado no Sistema
Pós-Condições	
Fluxo Principal – Adicionar horário	
Ações do Ator	Ações do Sistema
1 – Selecionar a opção Cadastrar Horário	
	2 – Carregar a tela com os campos: código, disciplina, dia da semana, hora, local e professor
2 – Preencher os campos	
3 – Confirmar pressionando o botão salvar	
	4 - Mostrar uma mensagem de confirmação para o usuário
	5 – Persistir as informações na base de dados
Restrições/Validações	1 – Todos os campos são obrigatórios
	2 – O campo código é gerado sequencialmente pela base de dados e não pode ser editável
Fluxo alternativo I – Editar horário	
Ações do ator	Ações do sistema
1 – Selecionar na listagem de Horários a linha correspondente que deseja editar	
	2 – Carregar o formulário com os campos desta manutenção preenchidos com os valores correspondentes ao registro selecionado possibilitando a edição
3 – Fazer a alteração dos dados	

4 – Confirmar a edição pressionando o botão gravar	
	5 – Apresentar uma mensagem de confirmação da alteração realizada
	6 – Fazer update na base de dados
	7 – Retornar para a listagem de Horários
Restrições/Validações	1 – As restrições de validações do fluxo alternativo I é o mesmo do fluxo principal
Fluxo alternativo II – Excluir horário	
Ações do ator	Ações do sistema
1 – Selecionar na listagem do Horário a linha correspondente que deseja excluir	
2 – Confirmar a exclusão dos dados	
	3 – Deletar os dados correspondentes da base de dados
Restrições/Validações	

Apêndice D - Documentação do Caso de Uso: usuário

Nome do Caso de Uso	Fazer Login
Caso de Uso Geral	
Ator Principal	Usuário
Atores Secundários	
Resumo	Descreve as etapas necessárias para realização do login no Sistema
Pré-Condições	Acessar a página do Sistema
Pós-Condições	Precisa estar cadastrado na base de dados
Fluxo Principal	
Ações do Ator	Ações do Sistema
Idênticos ao Caso de Uso – Secretária	Idênticos ao Caso de Uso – Secretária
Restrições/Validações	

Nome do Caso de Uso	Manter meus Dados
Caso de Uso Geral	
Ator Principal	Usuário
Atores Secundários	Secretaria
Resumo	Descreve as etapas necessárias para editar os dados pessoais no sistema
Pré-Condições	Acessar a página do Sistema
Pós-Condições	Dados editados
Fluxo Principal	
Ações do Ator	Ações do Sistema
Idênticos ao Caso de Uso – Professor/Coordenador	Idênticos ao Caso de Uso – Professor/Coordenador
Restrições/Validações	