

UNIVERSIDADE FEDERAL DO ABC
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO DO ESTÁGIO SUPERVISIONADO I

**ESTÁGIO EM DESENVOLVIMENTO DE SISTEMA PARA
USO INTERNO NA AGÊNCIA CADARIS**

RAFAEL CARDOSO DA SILVA

Orientador: Prof. Dr. Daniel Morgato Martin

Santo André, 2017

RAFAEL CARDOSO DA SILVA

**ESTÁGIO EM DESENVOLVIMENTO DE SISTEMA PARA
USO INTERNO NA AGÊNCIA CADARIS**

Relatório do Estágio apresentado ao Curso de
Bacharelado em Ciência da Computação como
requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Prof. Dr. Daniel Morgato Martin

Santo André, 2017

DEDICATÓRIA

Dedico todo projeto realizado aos meus amigos e familiares que depositaram grande confiança em mim desde o início.

AGRADECIMENTOS

Meus agradecimentos são voltados à Agência Cadaris, principalmente a Diretoria Comercial, que propôs essa oportunidade de amadurecer meus conhecimentos obtidos na Universidade Federal do ABC, além do crescimento pessoal.

????????????????????

A maioria das pessoas pensa no
sucesso e no fracasso como
opostos, mas eles são ambos
produtos do mesmo processo

Roger Von Oech

RESUMO

???????????????? DEPOIS ??????????????????

ABSTRACT

???????????????? DEPOIS ?????????????????

Sumário

1	INTRODUÇÃO	1
1.1	CARACTERÍSTICAS DO ESTÁGIO	1
1.2	CARACTERIZAÇÃO E ANÁLISE DA EMPRESA	2
1.3	VISÃO GERAL DO ESTÁGIO	2
1.4	ORGANIZAÇÃO DO TRABALHO	3
2	ATIVIDADES DESENVOLVIDAS	3
2.1	APRESENTAÇÃO DAS REGRAS DE NEGÓCIO	5
2.2	ESTUDO DAS TECNOLOGIAS DE DESENVOLVIMENTO WEB	5
2.3	IMPLEMENTAÇÃO	5
2.3.1	CADASTROS	5
2.3.2	COMERCIAL	5
2.3.3	RELATÓRIOS E CONTROLE FINANCEIRO	5
2.4	INTERFACE E EXPERIÊNCIA DE USUÁRIO	5
2.5	HOMOLOGAÇÃO, E AJUSTE FINAIS	5
2.6	OUTRAS ÁREAS DA EMPRESA	5
2.6.1	Resultados	5
2.6.2	Conclusão	5
2.7	DIAGNÓSTICO DOS PRINCIPAIS PROBLEMAS OBSERVADOS E SUGESTÕES DE MELHORIA	5
3	FUNDAMENTAÇÃO TEÓRICA	5
3.1	PLANEJAMENTO DE PROJETOS	6
3.2	DESENVOLVIMENTO WEB	7
3.2.1	HTML	7
3.2.2	JAVASCRIPT	8
3.2.3	ANGULARJS	9

3.2.4	CSS	10
3.3	LÓGICA DE PROGRAMAÇÃO	11
3.3.1	CONEXÃO COM BANCO DE DADOS	11
3.3.2	EXPRESSÕES REGULARES	12
3.4	BANCO DE DADOS	13
4	CONSIDERAÇÕES FINAIS	14
4.1	CONTRIBUIÇÕES PARA A FORMAÇÃO	14
4.2	DIFICULDADES ENCONTRADAS	15
4.3	SUGESTÕES DE TRABALHOS FUTUROS	15
5	REFERÊNCIAS BIBLIOGRAFICAS	16

1 INTRODUÇÃO

O estágio é uma atividade fundamental para a formação do aluno que está matriculado em um curso de nível superior. Nessa atividade o aluno tem a possibilidade de aplicar o conhecimento adquirido durante o período de graduação, além de poder desenvolver novos conhecimentos e habilidades ao enfrentar novos desafios e buscar soluções para problemas corriqueiros do trabalho realizado dentro de uma empresa.

Também é de muita importância para o desenvolvimento pessoal e profissional do aluno, visto que é o primeiro contato com o mercado de trabalho. Isso permite que o aluno desenvolva mais responsabilidades e aprenda a trabalhar em equipe, além de aprender a lidar com problemas reais dentro de uma empresa.

O estágio é uma tarefa supervisionada por um orientador e é gerado um relatório que é utilizado para se fazer a avaliação do aluno durante a atividade do estágio.

Este relatório tem por objetivo descrever as atividades e os produtos obtidos durante a disciplina de Estágio Supervisionado I. Nesta seção, apresenta-se a caracterização do estágio, da empresa, uma visão geral do estágio e a organização geral deste documento.

1.1 CARACTERÍSTICAS DO ESTÁGIO

A modalidade deste estágio é em *Home Office*¹, que também pode ser definido com um trabalho remoto ou teletrabalho, para a Agência Cadaris [ref:Cadaris]. A jornada de trabalho é de segunda à sexta-feira, desempenhando 6 horas diárias, totalizando 30 horas semanais, com a gratificação de uma bolsa de R\$ 1.100,00 por mês.

A função principal deste estágio é o desenvolvimento de um projeto de uso interno para a empresa. E por se tratar de um *Home Office*, proporcionando ao aluno um horário mais flexível de trabalho, como melhor se adapte a sua agenda acadêmica

¹Escritório em casa, em uma tradução livre do inglês, trabalho que é realizado em espaço alternativo ao escritório de uma empresa.

e pessoal. E também mantem a comunicação direta com a empresa por meio de e-mails e ligações via telefone e *Skype*.

1.2 CARACTERIZAÇÃO E ANÁLISE DA EMPRESA

O estágio foi iniciado no dia 01/08/2016, contratado pela Agência Cadaris, CNPJ 01.556.009/0001-46, que é uma agência de publicidade e está localizada na R. Dr. Thirso Martins, 100, cj 303, CEP: 04120-050, Vila Mariana, São Paulo-SP, telefone: (11) 5571-9142.

Cadaris foi fundada.....

A Agência Cadaris é uma agência de publicidade com atuação nas áreas de marketing, editorial e comunicação. Trabalhando com ética e transparência, de forma responsável e comprometida. Produzindo com qualidade e criatividade, sempre com respeito aos prazos acordados.

A empresa é formada por...

Os principais clientes da Cadaris estão: Colgate-Palmolive, Biotropic, ABO, TAM e Mattel, entre outras empresas.

1.3 VISÃO GERAL DO ESTÁGIO

Inicialmente, durante o primeiras mês, o estagiário foi introduzido ao funcionamento da agência, com a familiarização das regras de negócio e as ferramentas atuais utilizadas para apoiar a administração da empresa.

Após este período inicial, foi iniciada o estudo das tecnologias que serão utilizadas para o desenvolvimento deste sistema, que foi completamente discutido no período anterior.

A seguir é iniciado a implementação do sistema após todos os estudos e os planejamentos deste, com foco no *Back-end*. Primeiramente é desenvolvido o Cadastro

de recursos e algumas rotinas administrativas. Em seguida o processo Comercial. E por ultimo a geração de Relatórios e controle Financeiro.

Ao final é desenvolvido um novo *Front-end* para o sistema, junto ao time de *Front-end* da Cadaris, com recursos necessários para uma melhor experiência do usuário.

1.4 ORGANIZAÇÃO DO TRABALHO

As próximas seções estão organizadas como segue:

A seção 2 apresenta as informações detalhadas de todas as atividades realizadas pelo estagiário. A seção está dividida em quatro partes, uma para cada atividade desenvolvida.

A seção 3 descreve as considerações finais em relação a todo o estágio, como: contribuições para a formação, dificuldades encontradas e sugestões para trabalhos futuros.

A seção 4 apresenta os principais problemas observados pelo estagiário em cada etapa e sugestões de melhorias para cada uma delas.

A seção 5 detalha a relação entre as disciplinas cursadas pelo estagiário e as atividades realizadas durante os projetos.

A seção 6 apresenta a conclusão sobre o estágio desenvolvido.

2 ATIVIDADES DESENVOLVIDAS

Esta seção tem por objetivo detalhar as atividades desenvolvidas durante o estágio supervisionado. Inicialmente são apresentadas as atividades de aprendizagem e em seguida a de desenvolvimento.

Lista de trabalhos realizados:

ESTÁGIO 1

- Reunião sobre as regras de negócio da empresa (DER)
- Estudo das tecnologias a serem utilizados (Laravel, bootstrap)
- Dev o sistema para substituir a antes utilizada (antigo: planilhas)

ESTÁGIO 2

- Dev o sistema conforme as regras de negocio atual da empresa (antigo: VBD)
- Dev o sistema para a geração de relatórios e controle financeiro

ESTÁGIO 3

- Front-end (VueJS)
- teste, homologacao, e ajuste finais
- Outras áreas da empresa

2.1 APRESENTAÇÃO DAS REGRAS DE NEGÓCIO

2.2 ESTUDO DAS TECNOLOGIAS DE DESENVOLVIMENTO WEB

2.3 IMPLEMENTAÇÃO

2.3.1 CADASTROS

2.3.2 COMERCIAL

2.3.3 RELATÓRIOS E CONTROLE FINANCEIRO

2.4 INTERFACE E EXPERIÊNCIA DE USUÁRIO

2.5 HOMOLOGAÇÃO, E AJUSTE FINAIS

2.6 OUTRAS ÁREAS DA EMPRESA

2.6.1 Resultados

2.6.2 Conclusão

2.7 DIAGNÓSTICO DOS PRINCIPAIS PROBLEMAS OBSERVADOS E SUGESTÕES DE MELHORIA

3 FUNDAMENTAÇÃO TEÓRICA

As disciplinas cursadas contribuíram de várias formas, direta ou indiretamente, nas atividades realizadas no período de estágio. A Tabela 3.1 apresenta as disciplinas cursadas pelo aluno e que foram necessárias para o desenvolvimento das atividades.

Tabela 3.1: Disciplinas desenvolvidas durante o estágio com os suas respectivas abordagens.

Algoritmo e estrutura de dados	Ordenação
Análise de algoritmos	Custo de operações
Banco de dados	Banco de dados relacional; Diagrama Entidade-Relacionamento; Diagrama de classes; Consultas SQL.
Engenharia de software	Planejamento; Modelo Entidade-Relacionamento; Padrão MVC.
Linguagens formais e autômata	Expressões Regulares.
Lógica básica	Operadores Lógicos
Processamento da informação	Lógica de programação
Programação orientada a objetos	Paradigma orientado a objetos.
Programação para Dispositivos Móveis	HTML; Responsividade
Programação para Web	HTML; CSS; Javascript; Servlet API.
Segurança de Dados	Criptografia; Autenticação.

3.1 PLANEJAMENTO DE PROJETOS

Segundo os pontos citados na Engenharia de Software e nos livros de Sommerville [15] e Pressman [16], não basta apenas programar e desenvolver projetos, é necessário ter um planejamento completo antes de iniciar a programação em si, além de possuir desenvolvedores competentes na equipe. Todo sistema está sujeito a falhas, mas quando há bom planejamento, incluindo os imprevistos, dificilmente terá grandes problemas durante o período de utilização, nada que o comprometa seriamente. Tais fatos foram notados ao desenvolver os sistemas de Bolsa de Estudos PROMAIS e Sistema de Gestão de Atividades. O primeiro foi totalmente planejado, desde a estrutura das classes como quais ferramentas seriam utilizadas e o projeto foi desenvolvido do começo ao fim como o previsto. O segundo projeto havia sido planejado, mas durante o desenvolvimento foi notada muita repetição de códigos e a necessidade de uma ferramenta que tornasse a aplicação mais dinâmica e suprisse esse obstáculo, e assim foi pesquisado e encontrado o AngularJS [4].

3.2 DESENVOLVIMENTO WEB

3.2.1 HTML

Em **Programação para Web** foram desenvolvidas aplicações para web e em **Programação para Dispositivos Móveis** foram vistos conceitos semelhantes ao HTML. O HTML5 é uma linguagem que foi desenvolvida para exibir conteúdos em páginas web. A diferença para algumas outras linguagens de programação é que o HTML5 não necessita ser compilado, ele é apenas interpretado pelo navegador. A programação é realizada através de tags (<tag>) que podem possuir atributos e elementos filhos, desde que estejam dentro da tag de fechamento, que é representada por uma barra (</tag>). A Figura 3.1 demonstra a estrutura de um arquivo HTML com algumas tags novas que foram introduzidas na versão 5, como header, nav, section, article, aside e footer. Novos elementos foram introduzidos para dar semântica à linguagem, ou seja, ao ler um código é possível ter noção do conteúdo que será apresentado de acordo com as tags utilizadas, nas versões anteriores, todo o conteúdo possuía o mesmo nível de informação.

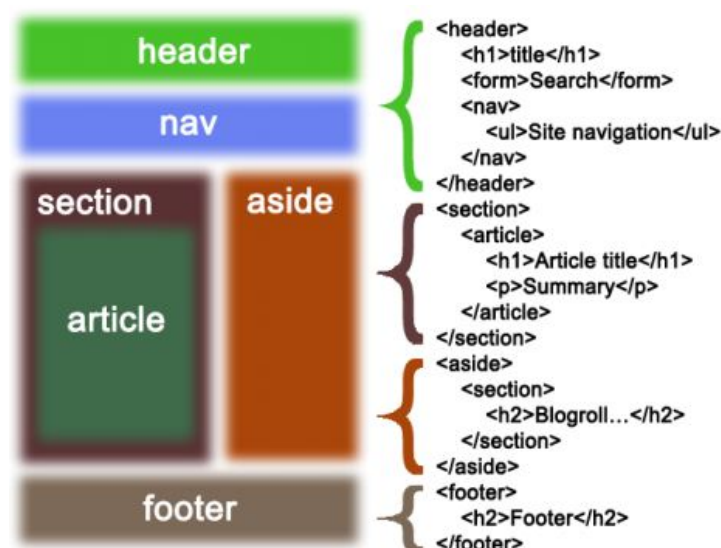


Figura 3.1: Estrutura do HTML com novas tags introduzidas na versão 5.

Com base na Figura 3.1, é demonstrada a herança de tags. As tags que são apresentadas dentro de uma tag de abertura e fechamento, são consideradas filhas, como h1, form e nav que são filhas da tag header.

O HTML5 é responsável pela exibição de conteúdo, sendo assim, não é capaz de criar novos elementos, conectar com um banco de dados ou com algum servidor, por exemplo. Essas outras atividades são realizadas por outras linguagens de programação que caminham em conjunto com o HTML5.

3.2.2 JAVASCRIPT

A linguagem de programação Javascript permite que haja interação entre o conteúdo HTML5 e o usuário, tornando a conteúdo da aplicação dinâmico. O DOM é uma multiplataforma que representa como as marcações em HTML, XHTML e XML são organizadas e lidas pelo navegador. Uma vez indexadas, estas marcações se transformam em elementos de uma árvore que você pode manipular via API, ou seja, o Javascript é capaz de manipular essas informações. Após o HTML ser carregado e interpretado pelo navegador, ele mesmo não tem poder para alterá-lo, mas o Javascript pode inserir, alterar ou remover nós em tempo de execução (Figura 3.2) [17].

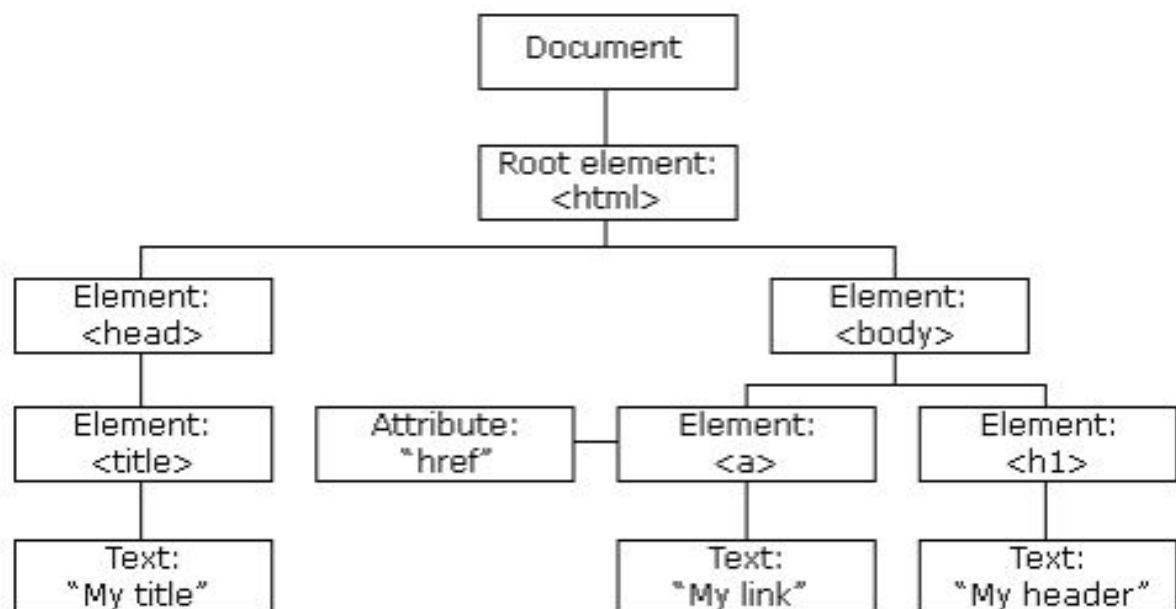
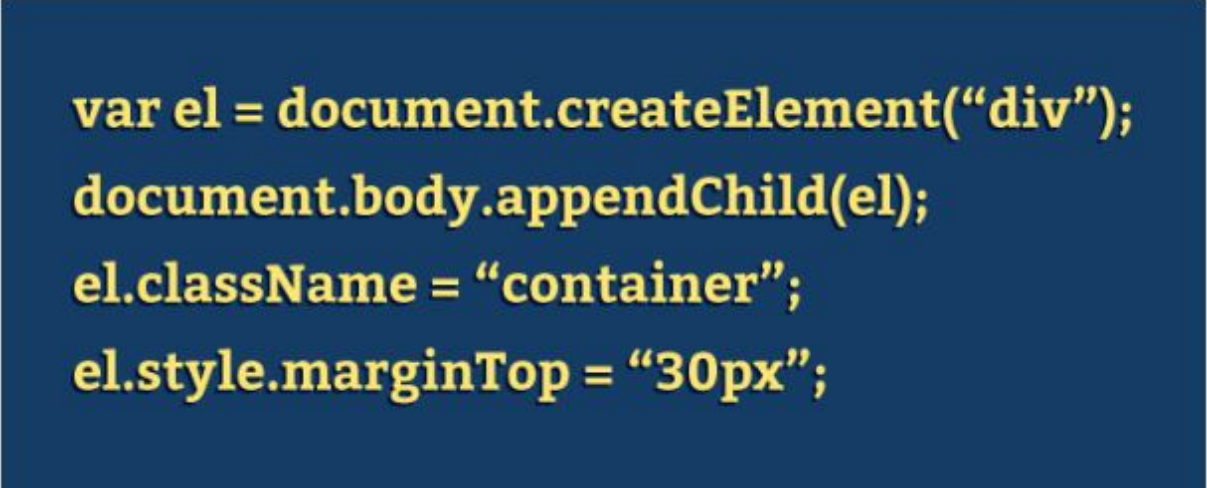


Figura 3.2: Estrutura de uma árvore DOM.

A Figura 3.3 apresenta um exemplo de manipulação de uma árvore DOM com o Javascript. A primeira linha cria uma tag do tipo “div” e salva na variável “el”. A segunda linha acrescenta a variável “el” ao elemento “body”. A terceira linha acrescenta uma classe CSS “container” ao elemento “el”. A quarta linha adiciona uma mar-

gem superior de 30px ao elemento. Se esse código fosse desenvolvido diretamente pelo HTML5, seria o mesmo que: `<body><div class="container" style="margin-top: 30px"></div></body>`

A dark blue rectangular box containing four lines of JavaScript code in a yellow, monospace-style font. The code demonstrates how to create a new div element, append it to the document body, and set its class and margin-top style.

```
var el = document.createElement("div");  
document.body.appendChild(el);  
el.className = "container";  
el.style.marginTop = "30px";
```

Figura 3.3: Exemplo de manipulação de DOM com Javascript.

3.2.3 ANGULARJS

O AngularJS [4] foi o grande encarregado por essa função de manipular o DOM nos sistemas de Bolsa de Estudos PROMAIS e o de Gestão de Atividades. A única necessidade do desenvolvedor é se preocupar com o JSON gerado e o AngularJS [4] o interpreta e cria o DOM de acordo com as informações que recebe. A Figura 3.4 é uma implementação básica em AngularJS [4] para a criação de um novo nó com o conteúdo desejado. O primeiro bloco é o código HTML em que a tag “li” possui uma instrução “ng-repeat” do AngularJS [4] para adicionar o conteúdo “text” da lista “opts” enquanto houver filhos. O código abaixo é o Javascript responsável por alimentar o JSON e adicionar o valor “x” ao clicar no botão “Add”. À direita está o resultado final da aplicação de exemplo. Esse mesmo código em Javascript puro seria um pouco mais complexo, mas nesse caso foi necessário acrescentar um novo valor no JSON e a aplicação se encarrega de acrescentar um novo nó automaticamente.



Figura 3.4: Exemplo de criação de nó com AngularJS.

3.2.4 CSS

O HTML5 possui atributos para personalizar as tags adicionadas, porém é um método depreciado e pouco eficiente com o surgimento do CSS3.

O CSS3 é uma linguagem de folha de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML5 e tem como papel principal separar o conteúdo de um documento e o seu formato.

A Figura 3.5 apresenta duas maneiras de como inserir personalizações CSS a um elemento: por classe e inline. A classe pode ser atribuída a vários elementos e assim todos terão apresentação em comum. Ao alterar uma classe, todos os elementos serão afetados e terão sua apresentação modificada. Os atributos da classe podem ser escritos no mesmo arquivo ou em um arquivo externo e importado. O método inline é adicionado com o “style”, é atribuído apenas ao elemento em questão e, caso haja alguma classe no elemento, sobrepõe os atributos coincidentes da classe.



Figura 3.5: Métodos de utilização do CSS <http://bit.ly/1G00zSp>.

3.3 LÓGICA DE PROGRAMAÇÃO

3.3.1 CONEXÃO COM BANCO DE DADOS

Todo o conteúdo até o momento está do lado do cliente em JSON, não foi enviado nada para o servidor e não foi feita nenhuma persistência de informações, portanto as informações são perdidas ao fechar a aplicação. Para a conexão com o banco de dados foi utilizada a linguagem C# da Microsoft, ela é uma linguagem orientada à objetos, fortemente tipada, foi baseada no C++ e possui similaridades com Java, linguagem desenvolvida em **Processamento da Informação, Programação Orientada a Objetos** e conceitos vistos em **Lógica Básica, Algoritmos e Estrutura de Dados e Análise de Algoritmos**.

A Figura 3.6 é um exemplo de conexão básica com o banco de dados na linguagem C# em que a conexão é declarada, em seguida são adicionados os parâmetros da conexão, a conexão é aberta, uma mensagem é exibida, a conexão é fechada e outra mensagem é exibida.

```

System.Data.SqlClient.SqlConnection con;

private void Form1_Load(object sender, EventArgs e)
{
    con = new System.Data.SqlClient.SqlConnection();

    con.ConnectionString = "Data Source=.\SQLEXPRESS;
AttachDbFilename=C:\\MyWorkers.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True";

    con.Open();

    MessageBox.Show("Database Open");

    con.Close();

    MessageBox.Show("Database Closed");
}

```

Figura 3.6: Exemplo de conexão básica com o bando de dados em C#.

3.3.2 EXPRESSÕES REGULARES

Foram utilizadas expressões regulares, conteúdo de **Linguagens Formais e Autômata**, no Sistema de Bolsa de Estudos PROMAIS para verificar se os valores inseridos pelos usuários estavam de acordo com as informações necessárias, por exemplo: número de telefone não pode aceitar caracteres ou caracteres especiais, mas pode aceitar hífen, parênteses e espaço. Sendo assim, a expressão regular que representa uma maneira correta para telefone é exibida na Figura 3.7.

Não há uma única expressão regular para alguns casos. Esse é um caso que depende da forma que o desenvolvedor decide para um melhor desempenho na sua aplicação. O exemplo apresentado permite receber telefones, obrigatoriamente, com DDD, mas sem obrigação dos parênteses. É permitido colocar espaço entre o DDD e o número ou preenchê-lo completamente sem espaço. No meio do telefone é aceito o sinal de hífen, ponto ou espaço. São aceitos telefones com oito ou nove números no total.

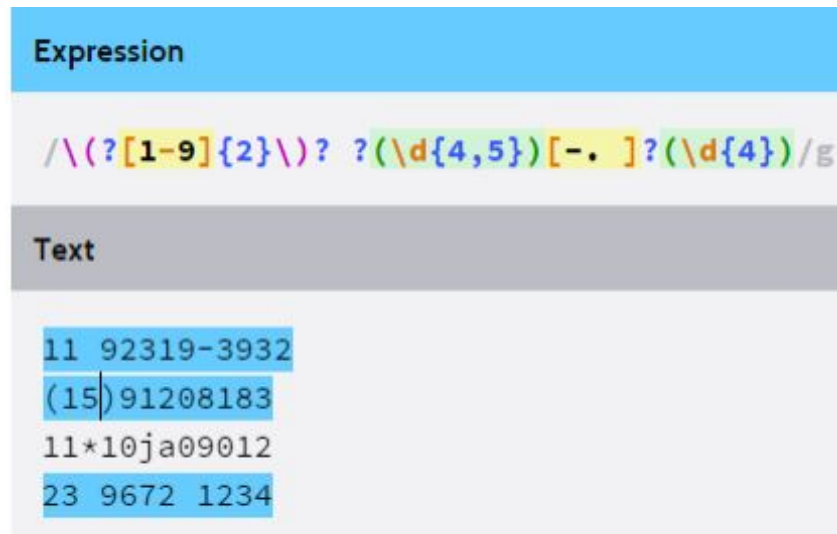


Figura 3.7: Expressão Regular para telefone.

3.4 BANCO DE DADOS

Os bancos de dados de todas as aplicações foram desenvolvidos em SQL Server da Microsoft e a disciplina **Banco de Dados** instruiu os primeiros passos para desenvolver uma estrutura sólida e coerente, de acordo com as formas normais. A disciplina de **Segurança de Dados** também contribuiu para que as informações fossem transferidas de forma criptografada e segura de usuários não autorizados.

Com banco de dados é possível criar consultas específicas para retornar qualquer tipo de informação necessária pelo usuário. Tabelas podem ser relacionadas através das chaves primárias e estrangeiras. As chaves primárias representam uma identificação única para a informação e a chave estrangeira é a representação de uma informação de outra tabela. A Figura 3.8 apresenta o fluxo de uma consulta e os elementos possíveis, como: colunas a serem exibidas, as tabelas a serem consultadas, as condições de filtros, agrupamentos e filtros após os resultados.

A modelagem dos bancos desenvolvidos pode ser vista em Figura 2.6 e Figura 2.14. Nelas é possível notar que estão nas formas normais, existem relacionamentos “um para um”, “um para n” e “n para n”.

Antes do desenvolvimento prático, foram feitas reuniões e modelagens conceituais para que fosse possível chegar ao modelo ideal sem que houvesse modificações

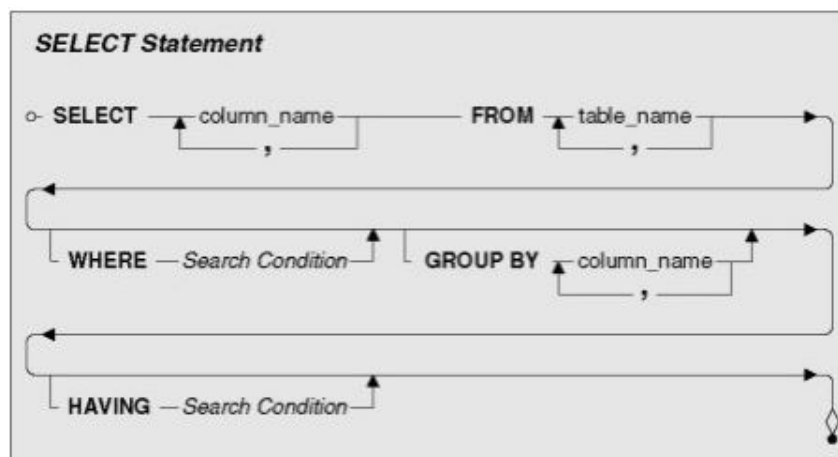


Figura 3.8: Modelo de consulta em banco de dados.

durante o desenvolvimento das aplicações.

4 CONSIDERAÇÕES FINAIS

Este capítulo tem por objetivo descrever as contribuições que o estágio proporcionou à formação acadêmica e profissional, as dificuldades encontradas durante o processo e sugestões de trabalhos futuros.

4.1 CONTRIBUIÇÕES PARA A FORMAÇÃO

O estágio exigiu conhecimento e aplicação de várias disciplinas como Processamento da Informação, Programação para Web, Programação Orientada a Objetos, Banco de Dados, Lógica Básica, Computadores, Ética e Sociedade, Linguagens Formais e Autômata, Segurança de Dados, Programação para Dispositivos Móveis, Análise de Algoritmos e Engenharia de Software. O estágio exigiu o conhecimento para criar lógicas que solucionassem problemas, operadores lógicos, expressões regulares, consultas em banco de dados, desenvolvimento em camadas (MVC) e planejamento.

As ferramentas ensinadas ao aluno durante os cursos de formação facilitaram as atividades e exigiram uma curva de aprendizado menor nos assuntos em que foi necessário aprofundamento, considerando que o aprendizado em um ambiente aca-

dêmico acontece em um cenário ideal e pequeno em relação ao mercado de trabalho.

O período total foi suficiente para perceber a diferença entre a teoria e a prática. Na teoria todo o assunto é passado ao aluno, os conceitos, maneiras de resoluções de problemas, os exercícios e projetos são grandes desafios e são os que põem a prova tudo o que foi aprendido nas aulas, porém, quando o aluno se depara com a realidade do mercado de trabalho, nota a diferença da dimensão de tudo o que foi visto antes e que nas aulas, apesar de parecer um projeto grande, é apenas um pequeno exemplo que pode acontecer na realidade. Essa distância entre a teoria e a prática é o que amadurece e faz com que o estagiário aplique tudo o que foi aprendido, contribuindo no crescimento profissional.

4.2 DIFICULDADES ENCONTRADAS

O desenvolvimento de aplicações no ambiente de produção exige muita disciplina do profissional. O código deve ser bem estruturado, pois será mantido por uma equipe. Deve haver um planejamento do projeto antes de iniciar qualquer programação para evitar erros e trabalhos desnecessários. O profissional necessita pesquisar muito sobre como realizar a tarefa, qual seria a melhor solução, quais seriam as melhores ferramentas, conversar com outros desenvolvedores para adquirir experiência e criar uma solução ideal para o problema.

Algumas ferramentas utilizadas no decorrer do estágio não foram ensinadas nos cursos de formação do aluno, dificultando inicialmente o desenvolvimento dos projetos. Porém, tal fato implicou em um maior empenho e dedicação em estudos e pesquisas, gerando ótimos resultados nos projetos e amadurecendo o aluno profissionalmente.

4.3 SUGESTÕES DE TRABALHOS FUTUROS

As disciplinas ministradas aos alunos devem possuir uma reciclagem periodicamente, pois a área de tecnologia possui novidades constantemente. Além da reciclagem nos assuntos de cada disciplina, também é preciso verificar a necessidade de

novas disciplinas que possam abranger as novidades de hardware e/ou software.

Os projetos desenvolvidos pelo aluno possuem tecnologias e ferramentas recentes e esse caminho deve ser mantido. Foram utilizadas ferramentas de código aberto, que facilita a atualização por parte dos desenvolvedores e da comunidade que participa com constantes contribuições. Outras ferramentas e recursos podem ser agregados aos sistemas para facilitar a experiência do usuário, mas com cuidado para que não haja exageros e conflitos com as ferramentas atuais.

O AngularJS demonstrou ser uma ferramenta com pequena curva aprendizado para tarefas mais comuns em um sistema. Ele deve ser mantido e aperfeiçoado nos sistemas atuais ecotado para o desenvolvimento de novos sistemas.

5 REFERÊNCIAS BIBLIOGRAFICAS

[1] Tripletech, <http://www.tripletech.com.br/>, acessado em 03/08/2015 às 06h50

[2] Faculdade de Direito de São Bernardo do Campo, <http://www.direitosbc.br/>, acessado em 03/08/2015 às 07h00.

[3] Faculdade de Direito de São Bernardo do Campo - Bolsas de Estudos, <http://www.direitosbc.br/a-faculdade-servicos-bolsa-de-estudo.aspx>, acessado em 04/08/2015 às 07h25.

[4] AngularJS by Google, <https://angularjs.org/>, acessado em 04/08/2015 às 07h40.

[5] JSON, <http://json.org/>, acessado em 04/08/2015 às 07h45.

[6] Bootstrap, <http://getbootstrap.com/>, acessado em 04/08/2015 às 08h15.

[7] Runrun.it, <http://runrun.it/>, acessado em 04/08/2015 às 13h20.

[8] Angular Material Design by Google, <https://material.angularjs.org/latest/>

#, acessado em 05/08/2015 às 13h50.

[9] Google, <http://google.com.br/>, acessado em 05/08/2015 às 13h55.

[10] Android 5.0 Lollipop,

https://www.android.com/intl/pt-BR_br/versions/lollipop-5-0/, acessado em 06/08/2015 às 07h35.

[11] Especificações Material Design by Google, <https://www.google.com/design/spec/material-design/introduction.html>, acessado em 06/08/2015 às 07h40.

[12] Google Inbox, <http://inbox.google.com.br/>, acessado em 06/08/2015 às 07h45.

[13] Google Music, <http://music.google.com.br/>, acessado em 06/08/2015 às 07h46.

[14] Google Keep, <http://keep.google.com.br/>, acessado em 06/08/2015 às 07h47.

[15] Sommerville, I. Engenharia de Software. 8.ed. - São Paulo : Addison-Wesley, 2007.

[16] Pressman, Roger S. Engenharia de Software. 6.ed. - Rio de Janeiro: McGraw-Hill, 2006

[17] O que é DOM, <http://tableless.com.br/tenha-o-dom/>, Acessado em 06/08/2015 às 13h35