IGOR HORTA CORREA

UMA FERRAMENTA DE *E-COMMERCE* PARA ELABORAÇÃO E COMPRA DE COMPUTADORES COM *HARDWARE* CUSTOMIZADO

LAVRAS - MG

2015

IGOR HORTA CORREA

UMA FERRAMENTA DE *E-COMMERCE* PARA ELABORAÇÃO E COMPRA DE COMPUTADORES COM *HARDWARE* CUSTOMIZADO

Monografia apresentada ao Colegiado do Curso de Sistemas de Informação, para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador

MSc. Renato Resende Ribeiro de Oliveira

Coorientador

Dr. Denilson Alves Pereira

LAVRAS - MG

2015

Ficha Catalográfica Elaborada pela Coordenadoria de Produtos e Serviços da Biblioteca Universitária da UFLA

Correa, Igor Horta.

Uma ferramenta de *e-commerce* para elaboração e compra de computadores com *hardware* customizado/ Igor Horta Correa. – Lavras : UFLA, 2015.

71 p. : il.

Monografia (graduação) – Universidade Federal de Lavras, 2016.
Orientador: Renato R. R. de Oliveira.
Bibliografia.

1. Extração de dados da web. 2. E-Commerce. 3. Web. 4. Banco de dados. I. Universidade Federal de Lavras. I. Título.

CDD – 004.6

IGOR HORTA CORREA

UMA FERRAMENTA DE *E-COMMERCE* PARA ELABORAÇÃO E COMPRA DE COMPUTADORES COM *HARDWARE* CUSTOMIZADO

Monografia apresentada ao Colegiado do Curso de Sistemas de Informação, para a obtenção do título de Bacharel em Sistemas de Informação.

APROVADA em 24 de janeiro de 2016.

Dr. Denilson A. Pereira	UFLA
Dr. Ahmed A. A. Esmin	UFLA
Dr. Raphael W. de Bettio	UFLA

MSc. Renato Resende Ribeiro de Oliveira

Orientador

LAVRAS - MG

2015

O presente trabalho

AGRADECIMENTOS

Agradeço a.

Agradeço também a.

RESUMO

Resumo da monografia

Palavras-chave: E-Commerce. Tecnologia da Informação. Extração de Dados. Mineração de Dados.

ABSTRACT

The english translation.

Keywords: E-Commerce. Information Technology. Data Extraction. Data Mining.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

HTML	Hypertext Markup Language
JS	JavaScript
JVM	Java Virtual Machine
SGBD	Sistema Gerenciador de Banco de Dados
MVC	Model View Controller
XML	extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização e Motivação

Com o aumento dos requisitos mínimos para se utilizar jogos e programas atuais e com o aumento de possíveis combinações de peças de computadores, a tarefa de customizar seu próprio computador para executar tarefas especificas se torna cada vez mais difícil. Visando facilitar esta tarefa, este trabalho propõe um sistema que utilize extração de dados para obter preços de diferentes peças de computadores em diferentes lojas de e-commerce e que então os utilize para auxiliar o usuário montar sua própria máquina visando o melhor custo benefício.

1.2 Objetivos Gerais e Específicos

O objetivo geral deste trabalho é uma aplicação web que realize a comparação de preços de peças de computadores em diferentes sites de e-commerce e que auxilie o usuário a comprar seu próprio desktop.

Visando atingir este objetivo geral, os objetivos específicos são definido a seguir:

- Desenvolver um algoritmo eficiente que seja capaz de coletar informações de produtos em diferentes sites de e-commerce.
- Desenvolver uma interface para aplicação.
- Modelar o banco de dados da aplicação.

1.3 Metodologia

Breve descrição da metodologia que foi utilizada...

1.4 Organização do Trabalho

O trabalho está organizado da seguinte forma, na Seção 2 será apresentado um referencial teórico acerca dos conceitos utilizados. Na Seção 3 será apresentada a metodologia do trabalho desenvolvido. Na Seção 4 serão apresentadas as simulações computacionais realizadas e os resultados obtidos. As conclusões e os trabalhos futuros serão apresentadas na Seção 5.

2 REFERENCIAL TEÓRICO

2.1 Hypertext Markup Language (HTML)

Linguagem de marcação é um conjunto de *tags*(palavras reservadas da linguagem) e regras que ao ser aplicados em um arquivo texto provem informações sobre o documento, tais informações podem ser por exemplo, de como o documento deve ser editado, formatado, exibido e impresso.

A *World Wide Web* sempre teve como linguagem de marcação o HTML. O HTML foi inicialmente desenvolvido para suprir a necessidade de descrever semanticamente documentos acadêmicos, porem, ao passar dos anos ele sofreu significativas adaptações e melhorias para suportar diversos novos tipos de documentos.

Em HTML as *tags* normalmente devem vir em duplas, onde a primeira indica o inicio do texto que sera modificado e a última o fim, como por exemplo:

`<title>` Olá mundo `</title>`

Ao executar este trecho de código, o navegador ira interpretar que o título da página deve ser ”Olá mundo”.

A figura 1, mostra um exemplo de um código em HTML, segundo W3C (2012) o HTML é dividido em 2 partes principais, *head*(cabeçalho) e *body*(corpo) o título do documento juntamente com demais informações sobre o mesmo ficam dentro da *tag* `<head>`.Dentro da *tag* `<body>` se encontra o conteúdo do documento. Já as *tags* `<h1>`,`<h2>`,`<h3>` representam tópicos no documento, onde `<h1>` é um elemento mais importante que `<h2>` e que `<h3>`. Navegadores normalmente renderizam os tópicos mais importantes com fontes maiores do que tópicos menos importantes.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <title>
            Titulo da Pagina
        </title>
    </head>
    <body>
        <div class='container'>
            <h1>Titulo grande</h1>
            <h2>Titulo medio</h2>
            <h3>Titutlo pequeno</h3>
        </div>
    </body>
</html >
```

Figura 1 Exemplo de documento em HTML

A figura 2, apresenta a interpretação do código anterior pelo navegador *Google Chrome*

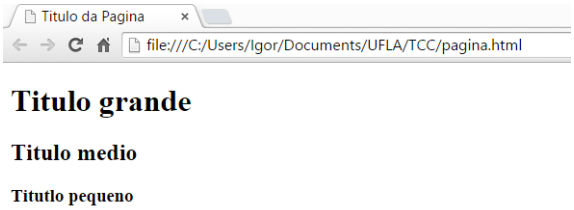

Figura 2 Interpretação de código HTML pelo browser Google Chrome.

2.2 Javascript

Segundo (ECMA International, 2011) linguagem de *scpript*(roteiro) é uma linguagem de programação usada para manipular customizar e otimizar recursos já existentes em um programa. Em tais sistemas, funcionalidades úteis já são disponíveis através de uma interface para o usuário, a linguagem de script é um mecanismo para expor tal funcionalidade para o controle da aplicação.

Na programação orientada a objetos implementa-se um conjunto de

classes que definem os objetos presentes na aplicação, cada classe determina o comportamento(definido no métodos) e estados possíveis(atributos) de seus objetos, assim como o relacionamento com outros objetos.

Javascript De acordo com (FLANAGAN, 2002) é uma linguagem de *script* com suporte a orientação a objetos, quando utilizada como *client-side*(lado do cliente) permite que conteúdo executável seja adicionado a páginas *web*,o que possibilita que estas interajam com o usuário, controlem o navegador e criem conteúdo HTML dinamicamente.

A figura 3, apresenta um documento em HTML que possui um trecho de código em Javascript, note que este trecho se encontra dentro da *tag* `<script>`, esta informa ao navegador que o trecho a seguir trata-se de um código em Javascript. Na linha 11 do código apresentado, podemos observar a chamada da função *write* do objeto *document*, esta função tem como objetivo adicionar texto a página que será exibida pelo navegador.

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4      <head>
5          <title>
6              Titulo da Pagina
7          </title>
8      </head>
9      <body>
10         <script type="text/javascript">
11                 document.write("Hello World from JavaScript!");
12         </script>
13     </body>
14 </html >
```

Figura 3 Código em HTML com trecho em Javascript

2.3 PhantomJS

Phatomjs é um navegador *opensource webkit*, programável em Javascript que possui suporte para diversos padrões web como: manipulação da DOM, seletores CSS, JSON, Canvas e SVG. Com o Phatomjs se é possível:

- Realizar testes unitários para aplicações web.
- Carregar, analisar e renderizar páginas web.
- Obter screenshots de websites.

O código apresentado na figura 4, associa uma instância da classe webpage do PhantomJS a uma variável chamada page, a partir de então a função open é chamada, esta função faz uma requisição ao site passado como parâmetro, no caso 'http://phatomjs.org/',se a requisição for um sucesso uma cópia da imagem renderizada pelo navegador PhantomJS e criada.

```
1  var page = require('webpage').create();
2  page.open('http://phantomjs.org/', function(status) {
3    console.log("Status: " + status);
4    if(status === "success") {
5      page.render('example.png');
6    }
7    phantom.exit();
8  });
```

Figura 4 Exemplo de código em Javascrpit que sera executado pelo phatomjs

A figura a seguir é a imagem gerada pelo código em Javascript

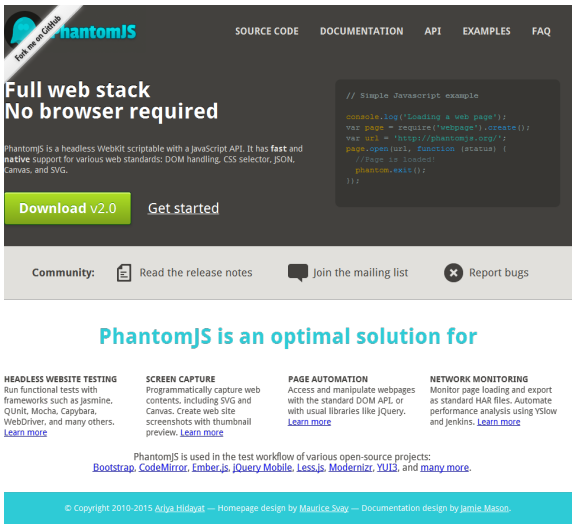

Figura 5 Imagem gerada após a execução do código em Javascript

2.4 Java

A linguagem de programação Java começou a ser desenvolvida em 1991 com a finalidade de ser utilizada em dispositivos eletrônicos inteligentes, porem com o crescimento exponencial da *World Wide Web* em 1993, os engenheiros da *Sun Microsystems* notaram o potencial da linguagem para aprimorar a funcionalidade de servidores *web* e a adaptou para estas funções.

A linguagem Java é orientada a objetos e possui o diferencial de ser interpretada, ou seja, diferentemente de linguagens como c e c++ em que seus códigos são diretamente transformados em códigos de maquinas quando compiladas, o java possui seu código transformado em uma linguagem intermediaria, esta é então interpretada em tempo de execução por um programa chamado *Java Virtual Machine*.Como resultado, Java pode muitas vezes atingir um nível de eficiência que é inatingível com intérpretes tradicionais.

Segundo (ROBERTS, 2013) As características principais do Java incluem o fato dela ser fácil de ser programada sem a necessidade de treinamento extensivo do desenvolvedor. Em Java estes desenvolvedores tem acesso a diversas bibliotecas que oferecem uma série de recursos já implementados tais como operações de entrada e saída, *interface* para redes e recursos gráficos. Por ser uma linguagem interpretada, uma outra vantagem é o fato dela possuir uma verificação de erros em tempo de execução.

2.5 MySQL

SGBD ou (sistema gerenciador de banco de dados), é um conjunto de programas que tem como objetivo armazenar, alterar, apagar e proteger informações de uma base de dados. Estes são bastante uteis em aplicações

web, por permitirem que as informações sejam armazenadas e utilizadas automaticamente.

Banco de dados armazenam informações sobre objetos distintos, entidades, associações ou ate mesmo sobre o relacionamento entre estas entidades. Como por exemplo, um banco para vendas pode guardar informações sobre produtos, clientes e vendas. Neste caso produtos e clientes são entidades enquanto vendas é um relacionamento entre cliente e produto.

No modelo de dados relacional os dados são percebidos pelos usuários como tabelas, onde cada coluna armazena um tipo de dados que podem ser de diversos tipos diferentes como numéricos, reais, texto e etc. Cada linha desta tabela representa uma instância.

Na figura 6: podemos ver como os dados são representados em um modelo relacional, a coluna ClienteID representa o identificador do cliente este número deve ser único, a segunda coluna representa o nome e a terceira o telefone celular de cada uma das linhas correspondentes.

ClienteID	Nome	Celular
1	Regina Ribeiro	35-97412145
2	Felipe Alvarenga	35-91421002

Figura 6 Exemplo de uma tabela de uma banco de dados relacional.

O MySQL segundo (TAHAGHOGHI; WILLIAMS, 2006) é um popular SGBD relacional *openSource*, que se destaca dos demais concorrentes por uma série de motivos, dentre elas podemos ressaltar: seu tamanho e velocidade, o MySQL funciona bem ate mesmo nos hardwares mais modestos, a velocidade com que ele consegue recuperar informações foi fator fundamental para torna-lo o SGBD favorito em diversas organizações. Outro fator muito importante para sua aceitação é a sua fácil instalação que pode ser realizada sem grandes dificuldades e com pouca configuração

2.6 Hibernate

Segundo (IRELAND et al., 2009), enquanto o paradigma relacional é um modelo extremamente comum em populares banco de dados do mercado, a programação orientada a objetos também é extremamente comum em aplicações atuais, logo a combinação entre estas tecnologias de diferentes paradigmas se torna inevitável. Enquanto o modelo relacional trata os dados como tabelas, o modelo orientado a objetos os representa como um conjunto de objetos, tamanha diferença em como os dados são representados geram uma série de problemas em aplicações que utilizam SGBDS relacionais e linguagens orientadas a objeto.

Como uma alternativa para este problema, algumas ferramentas de mapeamento objeto/relacional(MOR) foram criadas, dentre essas destaca-se o Hibernate

O Hibernate é um *framework*(ferramenta) de alta performance para mapeamento objeto/relacional em java. Este *framework* associa classes em java a tabelas do banco de dados. Ele também permite abstrair código sql em objetos java. Etre as diversas vantagens do Hibernate podemos ressaltar a facilidade em alterar formatados das tabelas, simplesmente alterando as classes java que as representam.

2.7 VRaptor

Model View Controller(Modelo Visão controlador) é um padrão de arquitetura de *software*, que separa a aplicação em três partes. O modelo (*model*) representa os dados da aplicação e suas respectivas regras de negócios. A visão (*view*) pode ser qualquer saída de representação dos dados.O controlador(*controller*) faz a mediação da entrada, preparando-a em coman-

dos para o modelo ou visão. O objetivo principal do MVC é trazer para aplicação reusabilidade de código e separação de conceitos.

O Vraptor é um *framework MVC* de código livre focado em gerar alta produtividade, foi designado para dar suporte ao desenvolvimento de *web-sites* dinâmicos bem como aplicações *web* que fazem uso do modelo *MVC*. Tem como objetivo poupar trabalho do desenvolvedor, mantendo-o focado nas suas atividades principais. De acordo com (CAVALCANTI, 2014), com o Vraptor é fácil aplicar as melhores práticas de orientação a objetos, já que este não impõe restrições no design das suas classes. É também um dos frameworks mais extensíveis em java, pois praticamente todos seus comportamentos podem ser sobrescritos com pouco código, sem a necessidade de configurações em *XML*.Além disso, é um *framework* que deixa seus usuários totalmente livres para escolher sua camada de visualização. Dentre as diversas vantagens citadas acima, podemos destacar o fato do Vraptor ser um *framework* nacional com ampla documentação em português.

3 METODOLOGIA

4 RESULTADOS E DISCUSSÃO

5 CONCLUSÃO E TRABALHOS FUTUROS

REFERÊNCIAS

CAVALCANTI, L. Vraptor, desenvolvimento ágil para web com java. *2ª Edição-São Paulo-Editora Casa do Código*, 2014.

ECMA International. *Standard ECMA-262 - ECMAScript Language Specification*. 5.1. ed. [s.n.], 2011. Disponível em: <http://www.ecma-international.org/publications/standards/Ecma-262.htm>.

FLANAGAN, D. *JavaScript: the definitive guide*. [S.l.]: "O'Reilly Media, Inc.", 2002.

IRELAND, C.; BOWERS, D.; NEWTON, M.; WAUGH, K. A classification of object-relational impedance mismatch. In: *Advances in Databases, Knowledge, and Data Applications, 2009. DBKDA '09. First International Conference on*. [S.l.: s.n.], 2009. p. 36–43.

ROBERTS, E. *Art and Science of Java*. [S.l.]: Pearson Education Limited, 2013.

TAHAGHOGHI, S. M.; WILLIAMS, H. E. *Learning MySQL*. [S.l.]: "O'Reilly Media, Inc.", 2006.

W3C. *HTML5 : a vocabulary and associated APIs for HTML and XHTML*. 2012. Acesso em 20/05/2015. Disponível em: <http://www.w3.org/TR/html5/>.