UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA VISUAL PARA GERAÇÃO DE ARQUIVOS DE SCRIPT EM PHP

LEONARDO WALTRICK SOMMARIVA

BLUMENAU
2008

2008/1-24

LEONARDO WALTRICK SOMMARIVA

FERRAMENTA VISUAL PARA GERAÇÃO DE ARQUIVOS DE SCRIPT EM PHP

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Alexander Roberto Valdameri, Mestre – Orientador

BLUMENAU
2008

2008/1-24

FERRAMENTA VISUAL PARA GERAÇÃO DE ARQUIVOS DE SCRIPT EM PHP

Por

LEONARDO WALTRICK SOMMARIVA

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:	Prof. Alexander Roberto Valdameri, Mestre – Orientador, FURB
Membro:	Prof. Adilson Vahldick, Especialista – FURB
Membro:	Prof. Paulo Fernando da Silva, Mestre – FURB

Blumenau, 09 de julho de 2008

Dedico este trabalho aos meus pais, aos meus amigos e ao meu orientador que me ajudaram diretamente na realização deste.

AGRADECIMENTOS

A toda minha família, pelo apoio.

Ao meu pai e minha mãe, pelo patrocínio, quando não tinha condições financeiras para pagar os custos de meu curso.

Á minha namorada Vanessa, que me fez voltar a andar, pelo carinho, pela inspiração e por me apoiar sempre em tudo o que eu faço.

Aos meus amigos, pelos momentos felizes e festas durante o curso.

Ao meu sócio Felipe Fert pela compreensão nos momentos que precisei deixar a empresa de lado para me dedicar à faculdade.

Ao meu orientador, Alexander Roberto Valdameri, por ter acreditado na conclusão deste trabalho, e por todo apoio que recebi durante o curso.

Se você quer ser bem sucedido tem que ter dedicação total, buscar o seu último limite e dar o melhor de si mesmo.

Ayrton Senna da Silva

RESUMO

Com a crescente utilização da Internet é comum que mais pessoas e empresas queiram divulgar informações na rede. Neste contexto, é essencial que a divulgação deste conteúdo seja feita de forma fácil e rápida sem ter conhecimentos técnicos. Este trabalho apresenta uma ferramenta visual, executada na web, para criação de páginas com divulgação na Internet sem exigir do desenvolvedor qualquer contato com código fonte. A ferramenta permite criar layout, utilizar funções da linguagem PHP e permite interação das páginas com banco de dados MySQL.

Palavras-chave: Desenvolvimento web. Ambiente de desenvolvimento integrado. Programação visual.

ABSTRACT

Within the growing utilization of the internet it becomes more common the desire of people and companies to advertise information in it. In this context it´s essential that these contents may be advertised in an easy and quick way as well as it´s necessary to not require much technological knowledge. This essay presents a visual tool, executed in the web to create sites with advertises in the internet without requiring from the developer any contact with source this tool allows one to create layout, use functions in the PHP language and also provides an interaction with pages with MySQL database.

Key-words: Web development. Integrated development environment. Visual programming.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

API – *Application Programming Interface*

CASE - *Computer-Aided Software Engineering*

CSS - *Cascading Style Sheets*

CLX - *Component Library for Cross Platform*

DHTML - *Dynamic* HTML

DOM - *Document Object Model*

IDE - *Integred Development Environment*

JSP – *Java Server Pages*

PHP - PHP *Hypertext Preprocessor*

RAD - *Rapid Application Development*

VCL - *Visual Component Library*

W3C - *World Wide Web Consortium*

XHTML - *eXtensible HyperText Markup Language*

XML - *eXtensible Markup Language*

XSLT - *eXtensible Stylesheet Language for Transformation*

SUMÁRIO

1 INTRODUÇÃO

Com a criação e evolução da web, tornou-se comum que empresas queiram divulgar seus produtos e serviços na internet, motivadas principalmente pela alta disponibilidade e por não ter limites de fronteira. Uma página pode ser acessada na rede no mundo inteiro. Com isso, cresceu muito o número de desenvolvedores especializados em web e também aumentou a complexidade para o desenvolvimento destas aplicações.

Segundo Soares (2004, p. 11), há poucos anos as páginas dinâmicas eram minoria no mundo virtual, com poucas linguagens oferecendo um suporte à plena utilização na internet. Todavia, hoje a realidade é outra, o usuário não se interessa mais por páginas estáticas e as empresas buscam divulgar seus produtos e serviços de forma interativa através de páginas dinâmicas.

A construção de uma aplicação utilizando os melhores recursos existentes atualmente exige dos desenvolvedores diversos conhecimentos como linguagens de marcação como *eXtensible HyperText Markup Language* (XHTML)[1], linguagens de estilo como *Cascading Style Sheets* (CSS), linguagens de programação para web como PHP: *Hypertext Preprocessor* (PHP), *Java Server Pages* (JSP), Java Script e conhecimento na administração de banco de dados.

As aplicações na web rodam através de um navegador e o usuário não necessita geralmente de mais nenhum outro software para que este aplicativo funcione corretamente. Os desenvolvedores web não precisam se preocupar com a compatibilidade entre seus aplicativos e os Sistemas Operacionais, proporcionando a vantagem dos aplicativos serem multiplataforma.

Diante do exposto, foi desenvolvida uma ferramenta gráfica executada na web e que possibilita aos usuários sem nenhum contato com código fonte, criar interfaces de páginas para internet através de uma ferramenta de programação visual. A ferramenta gera dinamismo nas páginas utilizando recursos da linguagem PHP e operações com banco de dados MySQL. O usuário pode manipular componentes na tela com *mouse*, clicando e arrastando estes componentes para modificar e acessar suas propriedades.

---

[1] Segundo Rogério (2006), o XHTML é uma evolução do *HyperText Markup Language* (HTML) sendo criado dentro do conceito do *eXtensible Markup Language* (XML).

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi criar uma *Integred Development Environment* (IDE) executada via web e que permite criar interfaces para páginas que serão publicadas na Internet.

Os objetivos específicos do trabalho são:

a) permitir ao usuário criar páginas dinâmicas utilizando a linguagem PHP com interação ao banco de dados MySQL;
b) criar uma biblioteca em *Dynamic* HTML (DHTML) que permita a manipulação com *mouse* e teclado de objetos dentro de uma página na web.

1.2 ESTRUTURA DO TRABALHO

O texto está estruturado em quatro capítulos. No segundo capítulo é apresentada fundamentação teórica com conceitos gerais sobre DHTML, AJAX, Programação Visual, e Ambiente de Desenvolvimento Integrado.

No terceiro capítulo é apresentado o desenvolvimento da ferramenta, incluindo a especificação dos requisitos e dos casos de uso, a modelagem estrutural das classes, o mapa de funcionalidades, a implementação e a operacionalidade da ferramenta proposta.

No último capítulo são apresentadas as conclusões e extensões com idéias para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nas seções seguintes são apresentados alguns aspectos teóricos relacionados ao trabalho, tais como: DHTML, AJAX, Programação Visual e Ambiente de Desenvolvimento Integrado.

2.1 DHTML

DHTML é uma junção entre três tecnologias diferentes, HTML, CSS e Java Script. A utilização do DHTML numa página significa que ela pode reagir ao usuário sem retornar continuamente ao servidor para obter mais dados.

Segundo Teague (2001, p. 168), o DHTML nada mais é que um termo de marketing criado pelo Netscape e pela Microsoft para descrever uma série de tecnologias introduzidas nas versões 4.0 dos seus navegadores, melhorando suas capacidades dinâmicas.

Embora não exista nenhuma definição oficial ou padronizada de DHTML, algumas considerações devem ser feitas como, por exemplo: A DHTML deve utilizar *tags* da HTML e das linguagens de criação de scripts sem exigir *plug-ins* ou qualquer outro software além do navegador. Além disso, a DHTML deve melhorar a interatividade e o apelo visual das páginas na internet.

2.1.1 Vantagens e desvantagens do uso do DHTML

Para Ferreira (2004), o DHTML não é acessível para quem não tiver um *browser* moderno, e dependendo como uma página foi desenvolvida, sem se preocupar com a compatibilidade entre *browsers*, pode ter uma série de implicações indesejadas. Um exemplo é um deficiente visual usando um leitor de tela que pode não conseguir entrar no site se caso o desenvolvedor não apresente um subtítulo para um menu desenvolvido em DHTML por exemplo. Por isso, para se desenvolver alguma página utilizando DHTML, deve ser analisado o impacto sobre acessibilidade da página.

O DHTML possui diversas vantagens comparadas a outras tecnologias para tornar

páginas web dinâmicas como o Macromedia Flash. Algumas delas são: apesar de divergências é suportada por diferentes navegadores, pode ser criado em pequenos arquivos de texto da mesma forma que um arquivo HTML, nenhum *plug-in* é necessário ser instalado pelo usuário para a sua execução, e nenhuma tecnologia envolvida é proprietária. As desvantagens são incompatibilidades entre certas operações entre navegadores, porque a implementação do *Document Object Model*(DOM), Java Script e CSS são diferentes entre os navegadores.

2.1.2 DOM

A capacidade de alterar uma página da web dinamicamente com uma linguagem de criação de scripts é possível através do DOM, o qual pode conectar qualquer elemento definido com através de um `id`, com uma função do Java Script. O DOM é o endereço no qual é possível acessar qualquer objeto, como por exemplo, um botão, um campo de texto, numa página HTML.

No quadro 1 é apresentado um exemplo de um código HTML.

```
<html>
      <head>
             <title>Título do Documento.</title>
      </head>
      <body>
    <h1>Título</h1>
    <p>Texto, texto, texto, <b>texto</b>...</p>
       </body>
</html>
```

Fonte: adaptado de Ferreira (2004).

Quadro 1 – Exemplo de código HTML

Ao exibir esse documento o navegador vai criar na memória uma estrutura hierárquica, em formato de árvore, com representações de cada objeto exibido. Essa estrutura inicia-se com o objeto `window`, que representa a própria janela aberta do navegador. Cada navegador tem pequenas particularidades ao montar os detalhes dessa estrutura. O documento acima quando exibido pelo navegador deve gerar a estrutura apresentada no quadro 2.

OBJETO	DESCRIÇÃO
Window	a janela do navegador ou frame onde o documento é carregado.
Opener	se é uma janela *pop-up*, opener aponta para a janela que a abriu.
Parent	se é um frame, parent aponta para o frame ou janela pai, aquele que contém o frameset que carrega esta página.
Frames	aponta para uma coleção se houverem frames no documento.
Location	um objeto que gerencia o endereço da janela atual. Possui entre outros os métodos reload e replace.
History	um objeto que gerencia o histórico da janela atual. Possui entre outros os métodos back e forward.
Document	representa o documento HTML aberto.
Title	o título do documento.
Body	representa a *tag* body, considerada a *tag* base para o acesso ao conteúdo do documento.
childNode 0	o elemento h1 do código, primeiro filho do body.
childNone 1	o elemento p, segundo filho de body. Esse elemento também possui um filho.
childNode 0 dentro de childNode 0	o elemento b dentro do p.

Fonte: adaptado de Ferreira (2004).

Quadro 2 – Estrutura do documento HTML

2.1.3 O papel do JavaScript no DHTML

Ferreira (2004) afirma que há um método para se acessar elementos do HTML que é de longe o mais usado, e é também o mais simples de todos. Trata-se do método getElementById, disponível no objeto document. O método getElementById está disponível no Internet Explorer desde a versão 5.0.

Há outros métodos para acessar os elementos do HTML. Um exemplo é o método getElementsByTagName, do objeto document. Esse método recebe uma string com um nome de *tag* e retorna um array com todas as *tags* daquele tipo. No quadro 3 é apresentado um exemplo da utilização do getElementsByTagName.

```
paragrafos=document.getElementsByTagName("p")

for(var x=0;x<paragrafos.length;x++)
    paragrafos[x].style.border="1px solid red"
```

Fonte: adaptado de Ferreira (2004).

Quadro 3 – Exemplo da utilização do método getElementsByTagName

A primeira linha armazena na variável parágrafos um array com todas as *tags* p do documento. E dentro do for é colocado uma borda vermelha em todos os parágrafos do

documento.

Além do atributo `style` há vários outros que são padrões e podem ser utilizados que são demonstrados no quadro 4.

ATRIBUTO	DESCRIÇÃO
`attributes`	um `array` com todos os atributos HTML da *tag*. Cada elemento do `array` é um objeto com as propriedades `name` e `value`. O `array` pode ser acessado por índice numérico ou pelo nome do atributo. Assim: `elemento.attributes["align"].value` retorna o valor do atributo HTML `align` da *tag* selecionada.
`childNodes`	é uma coleção de todos os nós filhos do atual.
`className`	é o nome da classe CSS do elemento.
`offsetWidth e offsetHeight`	altura e largura real do elemento na tela.
`offsetTop e offsetLeft`	posição real do elemento na tela.
`Id`	uma espécie de apelido do elemento.
`innerHTML`	retorna o código HTML dentro do elemento.
`nodeName e tagName`	retornam o nome da *tag* do elemento.
`parentNode`	elemento pai, aquele que contém o elemento atual.
`title`	valor do atributo `title` do elemento.

Fonte: adaptado de Ferreira (2004).

Quadro 4 – Atributos do elemento

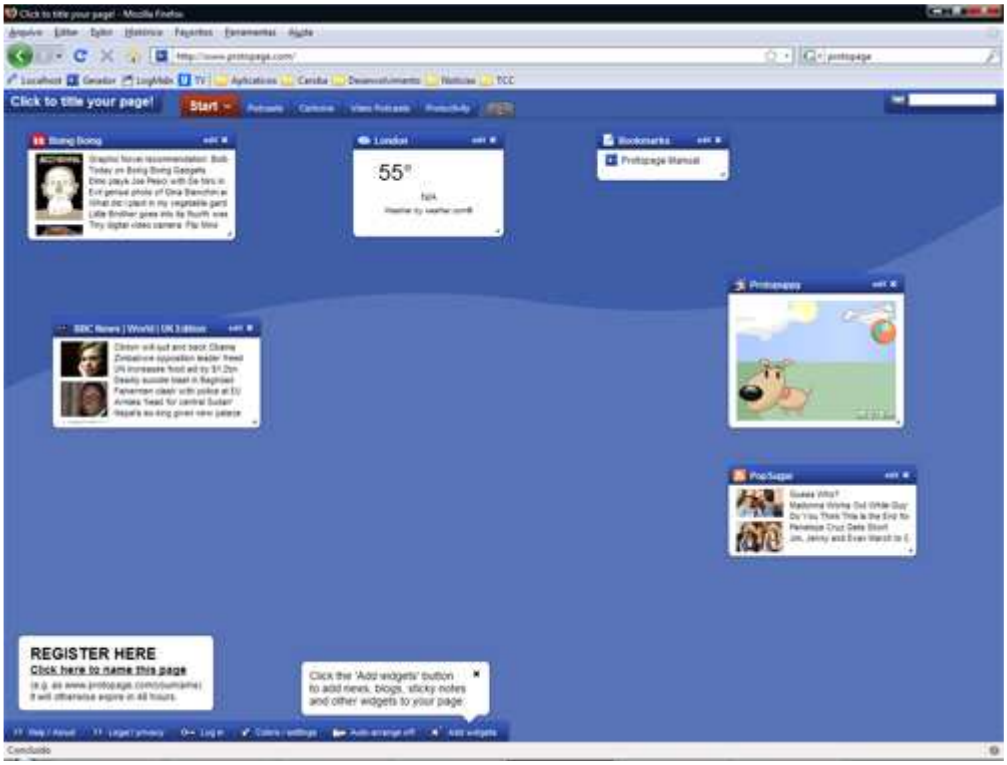
2.1.4 Exemplo de aplicações que utilizam DHTML

Um exemplo comum da utilização do DHTML é no site da Prefeitura do Município de São Bento do Sul. Há um menu principal com dez opções. No exemplo da figura 1, foi selecionado o *link* “Administração”. Após de selecionar o *link*, abriu um novo menu, com quatro opções. Selecionando “Secretarias”, abre mais um menu ao lado com a listagem de secretarias da prefeitura. Nesse caso a utilização do DHTML, evita que o usuário tenha que navegar em diversas páginas até encontrar uma secretaria.

Fonte: Prefeitura do Município de São Bento do Sul (2006).

Figura 1 – Site da Prefeitura do Município de São Bento do Sul

Na figura 2, é apresentado outro exemplo de uso do DHTML, neste caso, mais avançado, no site Protopage. Nele o usuário pode criar e personalizar páginas com quadros que podem ser arrastados, fechados, minimizados e manipulados através do DHTML. Os conteúdos dos quadros também podem ser alterados dinamicamente.



Fonte: Protopage (2007).

Figura 2 – *Site* Protopage

2.2 AJAX

Segundo Borba (2006, p. 20), o desenvolvedor WEB sempre encontrou limitações técnicas para atualizações de informações em páginas WEB. Para atualizar uma informação dentro de uma página era necessário que um *frame* ou uma página inteira fosse atualizado, muitas vezes para atualizar um número ou uma pequena informação na página.

Este problema foi resolvido com a chegada do ActiveX XMLHtttpRequest, criado pela Microsoft, que lançou no Internet Explorer 5.0 e sem seguida adaptado pelos demais navegadores concorrentes. Com esse objeto é possível enviar requisições assíncronas ao servidor que devolve os dados para o navegador que usa essas informações para atualização dinâmica para da página sem precisar atualizá-la inteira.

2.2.1 Tecnologias envolvidas

Para Borba (2006, p. 20) nenhuma tecnologia que o Ajax envolve é inovadora. Os seus

componentes tecnológicos já existem na web há um tempo. O que é notável no AJAX é a forma de como essas tecnologias trabalham em conjunto.

Em seguidas estão listadas as principais tecnologias do AJAX:

a) HTML, XHTML: formato básico de apresentação de conteúdo;
b) DOM: É a estrutura completa de uma página web disponível à linguagem Java Script por meio de objetos possíveis de manipulação. Essa combinação torna possível a alteração da interface e conteúdo de uma página web em tempo de execução;
c) CSS: Organiza a construção do visual da interface de forma padronizada, pode ser reutilizada e também padronizada. Se utilizada em conjunto com *eXtensible Stylesheet Language for Transformation*(XSLT) mais *eXtensible Markup Language*(XML), pode gerar páginas XHTML. Com CSS é possível definir padrões de aparência e comportamento dos elementos de uma página. As cores, parágrafos, margens espaçamentos, espessura ou qualquer outro elemento da página, pode ser definido em apenas um arquivo e, assim, a qualquer momento, pode-se mudar totalmente a forma de visualização de todas as páginas de um site, apenas alterando esse arquivo que contém as definições de CSS;
d) XML: formato padronizado de dados para manipulação e comunicação;
e) XSLT: Transformação do XML em XHTML combinado com estilos CSS;
f) XMLHttpRequest: O objeto XMLHttpRequest é responsável pela comunicação assíncrona, ou seja, é o agente principal pata implementação do AJAX que é acionado pelo código em JavaScript;
g) Java Script: É a linguagem principal do AJAX, voltada para executar operações do lado do cliente, faz o elo de ligação entre as demais tecnologias. Java Script não é uma linguagem compilada, ela é interpretada e é inserida junto com o código fonte HTML das páginas.

No servidor, o desenvolvedor tem a opção de optar por diversas tecnologias para atender e processar as requisições feitas com AJAX como o PHP, *Active Server Pages*(ASP), ASP.Net, entre outras.

Basicamente, a tecnologia AJAX é fortemente atrelado ao objeto XMLHttpRequest, o qual dá-se maior ênfase a seguir.

2.2.2 Objeto XMLHttpRequest

Segundo Limeira (2006), XMLHttpRequest é o objeto que faz a conexão assíncrona entre a página e o servidor de aplicações Web, ou seja, é a tecnologia principal do AJAX, sem ele, o AJAX não existiria. Trata-se de um objeto Java Script que pode ser usado para fazer requisições ao servidor Web, em segundo plano, sem congelar o navegador ou recarregar a página atual. Este objeto é hoje parte da especificação do DOM, nível 3. Ou seja, qualquer navegador que queira oferecer suporte aos padrões precisa implementar o objeto XMLHttpRequest.

A troca de informações com o servidor web, geralmente utiliza XML, embora possa utilizar qualquer outro formato de texto, como por exemplo, o HTML.

O XMLHttpRequest foi criado inicialmente no navegador Internet Explorer 5 como um componente do ActiveX. Depois disso, outros navegadores passaram a implementá-lo, mas o grande problema é que o XMLHttpRequest não é um padrão *World Wide Web Consortium*(W3C) e por isso cada navegador pode implementar de maneiras diferentes. Atualmente, o Mozilla Firefox, Konqueror, Safari e Opera implementam este objeto da mesma forma, ou seja, como um objeto JavaScript nativo. Já o Internet Explorer prefere implementar esse mesmo recurso utilizando outro objeto, o ActiveX.

O XMLHttpRequest possui métodos apresentados no quadro 5.

MÉTODOS	DESCRIÇÃO
open(método, url,Assíncrona, usuário,senha)	esse método relaciona o objeto à página web que se deseja conectar. O argumento de método pode ser GET, POST ou PUT. O endereço pode ser relativo ou absoluto. Os 3 últimos parâmetros são opcionais.
Send(content)	envia a solicitação para o servidor. Caso a conexão tenha sido aberta com o parâmetro Assinc igual a false (indicando que a conexão não é assíncrona), esse método aguarda a resposta do servidor; caso contrário, não há espera (o que deve ser o padrão para aplicações AJAX).
setRequestHeader	configura o cabeçalho http especificado com o valor fornecido.
getResponseHeader	retorna o valor da string do cabeçalho especificado.
getAllResponseHeaders	retorna uma string com todos os cabeçalhos http especificados.
Abort	interrompe o processamento atual do objeto XMLHttpRequest.

Fonte: adaptado de Asleson e Schutta (2006, p. 25).

Quadro 5 – Métodos do XMLHttpRequest

O XMLHttpRequest possui propriedades apresentados no quadro 6.

PROPRIEDADES	DESCRIÇÃO
status	contém o código de status enviado pelo servidor web.
statusText	a versão em texto do código de status http.
readyState	o estado da solicitação. Os cinco valores possíveis são 0- não inicializada, 1- carregando, 2- carregada, 3- interativa e 4- concluída.
responseText	a resposta do servidor na forma de uma string.
responseXML	a resposta do servidor em formato XML.
Onreadystatechange	o manipulador de eventos que é acionado a cada mudança de estado, normalmente, uma chamada a uma função Java Script.

Fonte: adaptado de Asleson e Schutta (2006, p. 25).

Quadro 6 – Propriedades do XMLHttpRequest

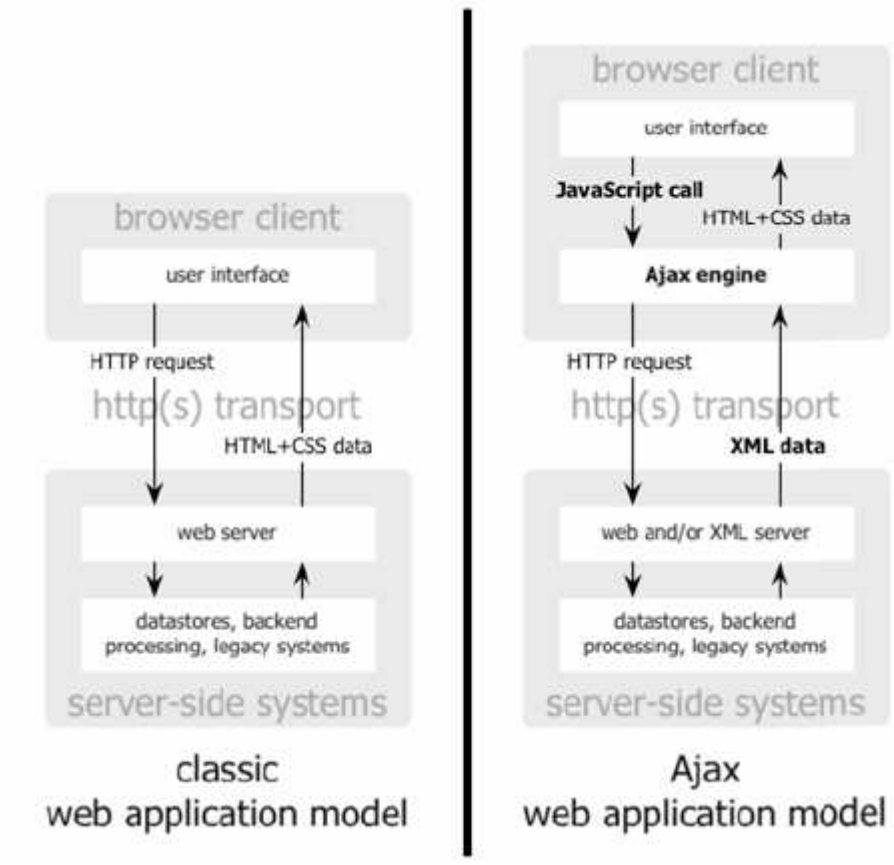
2.2.3 Vantagens do AJAX

Com o AJAX pode-se trafegar apenas os dados que realmente foram atualizados em uma página Web e assim ganhar em desempenho e economia de tráfego de dados. O seu uso é recomendado para vários casos, dentre eles, validação de dados de um formulário, atualização de informação em pouco tempo, como por exemplo, a cotação de ações da bolsa de valores em tempo real.

Além disso, é possível disponibilizar aplicações com alta capacidade de interatividade e usabilidade sem precisar de recursos pesados de programação ou necessidade de softwares proprietários.

2.2.4 Diferenças entre aplicações normais e com AJAX

Na figura 3, percebe-se bem a diferença entre os dois modelos e os módulos envolvidos. À esquerda, no modelo tradicional, tem-se a interface do usuário interagindo diretamente com servidor web para enviar a página inteira para o servidor e depois receber uma nova página inteira para ser mostrada ao usuário que fica aguardando o retorno sem conseguir fazer qualquer outra operação. À direita, no modelo AJAX, a interface do usuário interage com um mecanismo AJAX que administra as solicitações entre cliente e servidor, só passando as informações necessárias de ambos os lados, permitindo que o usuário, quando solicite algo da página Web, possa continuar trabalhando na página enquanto o resultado ainda não é mostrado.

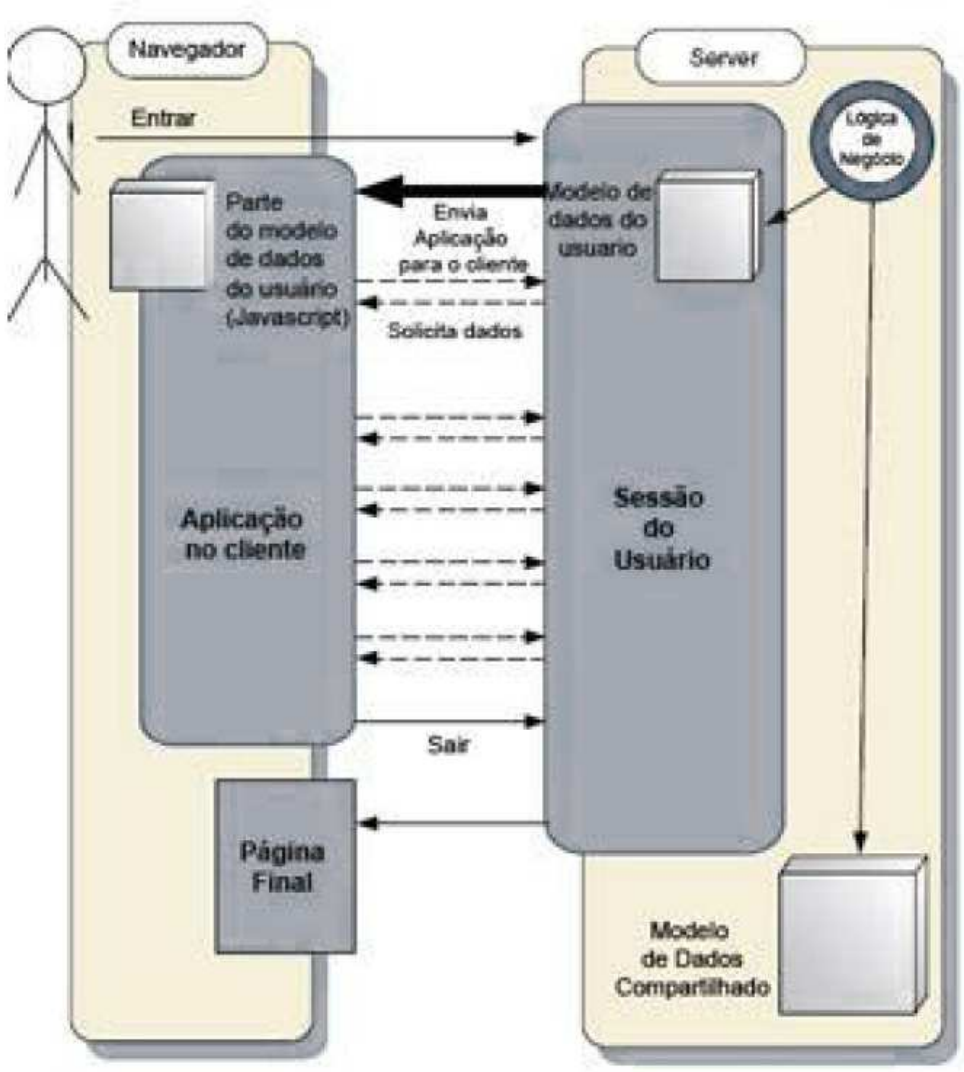


Fonte: Garret (2006).

Figura 3 – Comparativo entre aplicações normais e aplicações que utilizam AJAX

2.2.5 Estrutura do AJAX

Na figura 4, observa-se um fluxo de informações utilizando AJAX. Neste caso, o navegador interage com o servidor, passando apenas as informações solicitadas e apenas nos momentos que realmente são necessárias, evitando, assim, o tráfego de toda página a cada requisição.



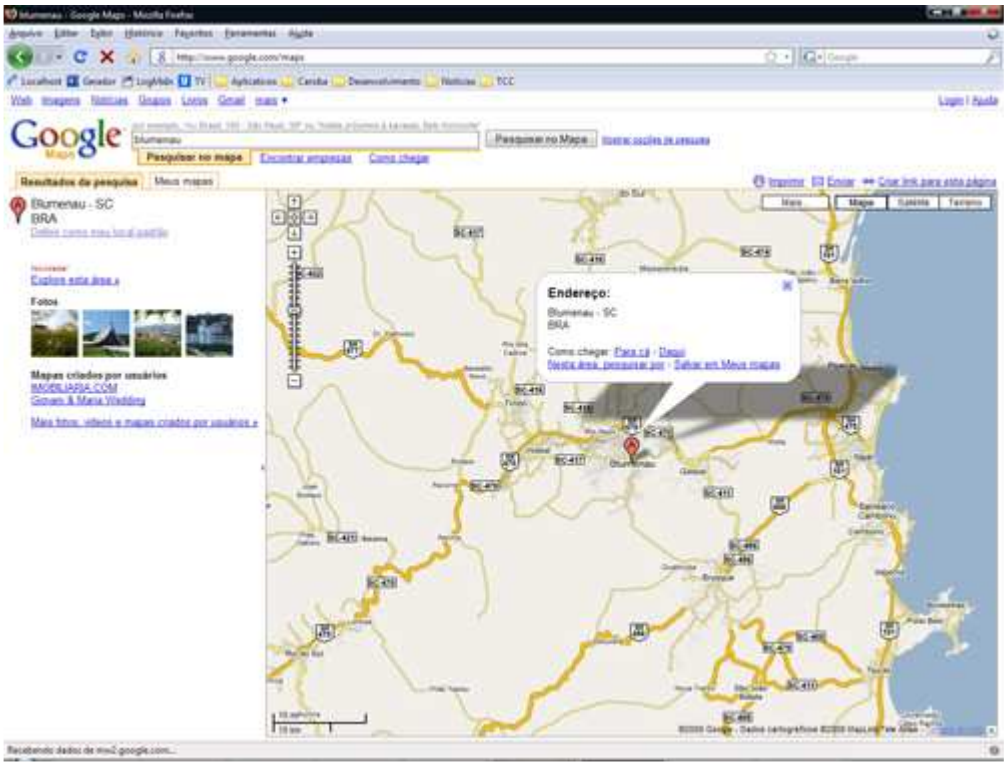
Fonte: Sousa(2006).

Figura 4 – Fluxo de informações de aplicações AJAX

2.2.6 Exemplos de aplicações com AJAX

O Google Maps é um cruzamento entre um visualizador de mapas e um sistema de pesquisa. O mapa pode ser consultado usando texto livre, permitindo afunilar a pesquisa para

endereços de ruas específicas ou até mesmo para nomes de hotéis e restaurantes. Na figura 5 mostra uma visão geral do Google Maps (GOOGLE INC, 2004).



Fonte: Google INC(2004).

Figura 5 – Visão geral do Google Maps

O recurso de pesquisa funciona como um aplicativo Web clássico, atualizando a página inteira, mas o mapa em si é alimentado pelo AJAX. Clicar em links individuais a partir da pesquisa de um hotel fará com que *pop-ups* adicionais sejam exibidos de modo instantâneo, possivelmente até mesmo rolando o mapa levemente para acomodá-los.

A rolagem do mapa é o recurso mais interessante do Google Maps. O usuário pode arrastar o mapa inteiro usando o mouse. O mapa é composto de pequenos pedaços e, se o usuário rolar pelo mapa até um ponto suficientemente distante para expor m novo pedaço, este será carregando assincronamente.

Às vezes há um retardo notável, com uma área em branco aparecendo inicialmente, que é preenchida depois pelo pedaço do mapa que é carregado. O usuário pode continuar a rolar, desencadeando solicitações atualizadas daquele pedaço, enquanto é feito *download*. Os pedaços do mapa são armazenados em *cache* pelo navegador até o final de uma sessão de usuário, tornando muito mais rápido retornar a uma parte do mapa já visitada.

2.3 PROGRAMAÇÃO VISUAL

É uma forma de programação onde os desenvolvedores constroem suas aplicações utilizando a manipulação gráfica de componentes referentes a comandos e recursos de uma linguagem de programação através de uma barra de ferramentas (RODRIGUES, 2006, p. 19).

Segundo Cantú (2003, p. 4), ao trabalhar com programação visual, o tempo do desenvolvedor é gasto em duas partes diferentes do aplicativo: nos projetistas visuais e no código fonte. Os projetistas visuais permitem que você selecione os componentes necessários e defina um valor inicial das propriedades dos componentes.

A maioria dos ambientes que utilizam os conceitos de programação visual possuem uma caixa de ferramentas com vários botões, onde cada botão possui uma imagem ou uma palavra correspondente a uma funcionalidade da linguagem trabalhada, e com a seta do *mouse* o usuário seleciona o componente e arrasta para a área de trabalho, podendo então inseri-lo no programa. Com o componente na área de trabalho, o programador pode alterá-lo ou apagá-lo sem escrever nenhuma linha diretamente no código.

As vantagens da programação visual são que o desenvolvedor tem menos contato com código fonte, acelerando então o desenvolvimento do sistema a ser construído e também exige menos conhecimento do programador.

Na figura 6 mostra a interface gráfica da ferramenta de programação visual Borland Delphi.

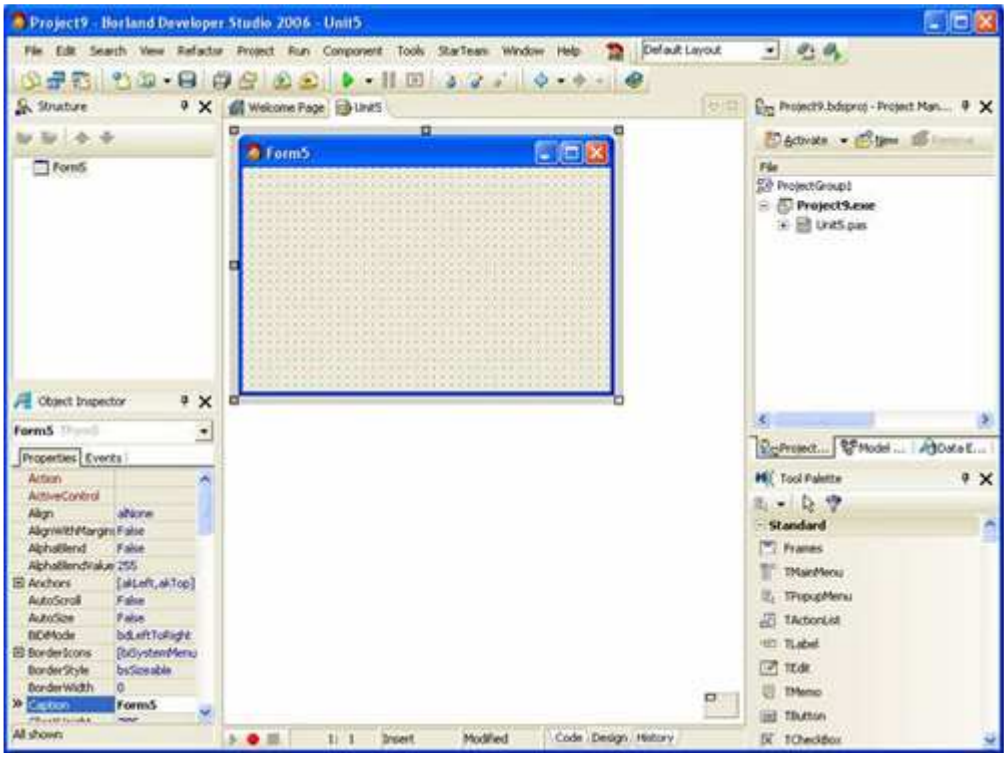


Figura 6 – Borland Delphi

Cantú (2003, p. 4), afirma que o Borland Deplhi, é uma ferramenta de programação visual, que possui uma IDE bastante intuitiva. A ferramenta possui duas bibliotecas visuais diferentes, a *Visual Component Library*(VCL), que é dependente de plataforma e a Component Library for Cross Platform(CLX), que é independente de plataforma.

2.4 AMBIENTE DE DESENVOLVIMENTO INTEGRADO

Conforme Bolonha, Hampshire e Leão (2004, p. 5), nos últimos anos, o surgimento dos sistemas operacionais com interface gráfica fez com que as tarefas relacionadas ao desenvolvimento de softwares comerciais sofressem transformações radicais.

Inicialmente, o desenvolvimento de sistema para ambiente Windows requeria a utilização da linguagem C, na qual estão implementadas as funções da *Application Programming Interface*(API) do Windows. O desenvolvimento de uma aplicação extremamente simples, que exibisse apenas uma janela com alguma mensagem estática, requeria dezenas de linhas de código em linguagem C. Em aplicações mais complexas e com interface gráfica o problema aumentava pelo excesso do número de linhas de código. Com isso surgiram as primeiras ferramentas com conceito *Rapid Application Development*(RAD).

Ferramentas com conceito RAD permitem associar, de maneiras simples e rápidas, um elemento de interface e código de aplicação. Uma das primeiras ferramentas a adotar o conceito RAD foi o Visual Basic for Windows, da Microsoft.

IDE é um software que provem facilidades para programadores no processo de desenvolvimento de software, utilizando o conceito de RAD. As IDEs são projetadas para auxiliar ao máximo a produtividade do programador com barras de componentes com ferramentas para geração e formatação de código, entre outros.

Geralmente, uma IDE é desenvolvida especificamente para uma linguagem de programação, proporcionando um conjunto de características que melhor corresponde a linguagem que está sendo trabalhada.

Os primeiros ambientes consistam num simples editor de código, com algumas funcionalidades que facilitavam algumas tarefas do programador, como por exemplo, atalhos de teclado para realizar compilação.

2.4.1 Principais funcionalidades

Nas IDEs mais modernas possuem diversas funcionalidades dentre as mais importantes:

a) editor: edita o código fonte do programa escrito nas linguagens suportadas pela IDE;
a) compilador: compila o código fonte do programa editado numa linguagem específica e transforma em linguagem de máquina;
b) *linker*: tem a função de unir vários programas já compilados de forma independente e unificá-los em um programa executável;
c) *debugger*: auxilia no processo de encontrar e corrigir erros no código fonte do programa;
d) modelagem: criação do modelo de classes, objetos, interfaces, associações e interações dos artefatos envolvidos no software com o objetivo de solucionar as necessidades-alvo do software final;
e) geração de código: característica mais explorada em ferramentas *Computer-Aided Software Engineering*(CASE). A geração de código também é encontrada em IDEs, contudo com um escopo mais direcionado a *templates* de código comumente utilizados para solucionar problemas rotineiros. Todavia, em conjunto com

ferramentas de modelagem, a geração pode gerar todo ou praticamente todo o código fonte do programa com base no modelo proposto, tornando muito mais rápido o processo de desenvolvimento de software;

f) *deploy*: auxilia no processo de criação do instalador de software;
g) testes automatizados: realiza testes no software de forma automatizada, com base em *scripts* ou programas de testes previamente especificados, gerando um relatório dos mesmos, assim auxiliando na análise do impacto das alterações no código fonte;
h) *refactoring*: consiste na melhoria constante no código fonte do software, seja na construção de código mais otimizado, mais limpo e com melhor entendimento.

Algumas IDEs possuem ferramentas que auxiliam o desenvolvedor na criação de interfaces gráficas.

2.4.2 Exemplos de Ambientes de Desenvolvimento Integrados

Um exemplo de ambiente integrado de desenvolvimento é o Eclipse IDE. Escrito em Java, na sua forma padrão é feita para desenvolvedores em Java. Os usuários podem estender suas funcionalidades instalando *plug-ins*, como por exemplo, *toolkits* para trabalhar com outras linguagens de programação. O desenvolvedor também pode desenvolver *plug-ins* próprios. A figura 7 apresenta a interface gráfica do Eclipse IDE:

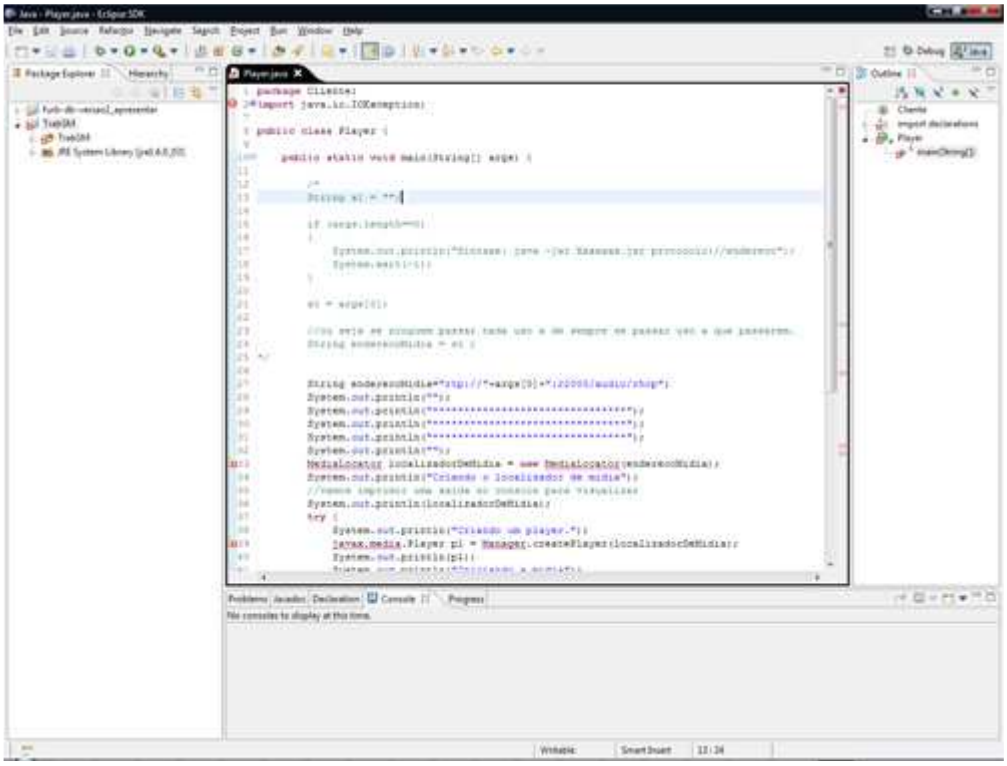


Figura 7 – Interface gráfica do Eclipse IDE

Outro exemplo de IDE é o Zend Studio, ambiente pago, escrita em Java e criada para desenvolvedores da linguagem PHP. Além de ser um editor de código, proporciona uma série de ajudas ao desenvolvedor como, criação e gestão de projetos e depuração do código. Na figura 8 é apresentada a interface gráfica do Zend Studio.

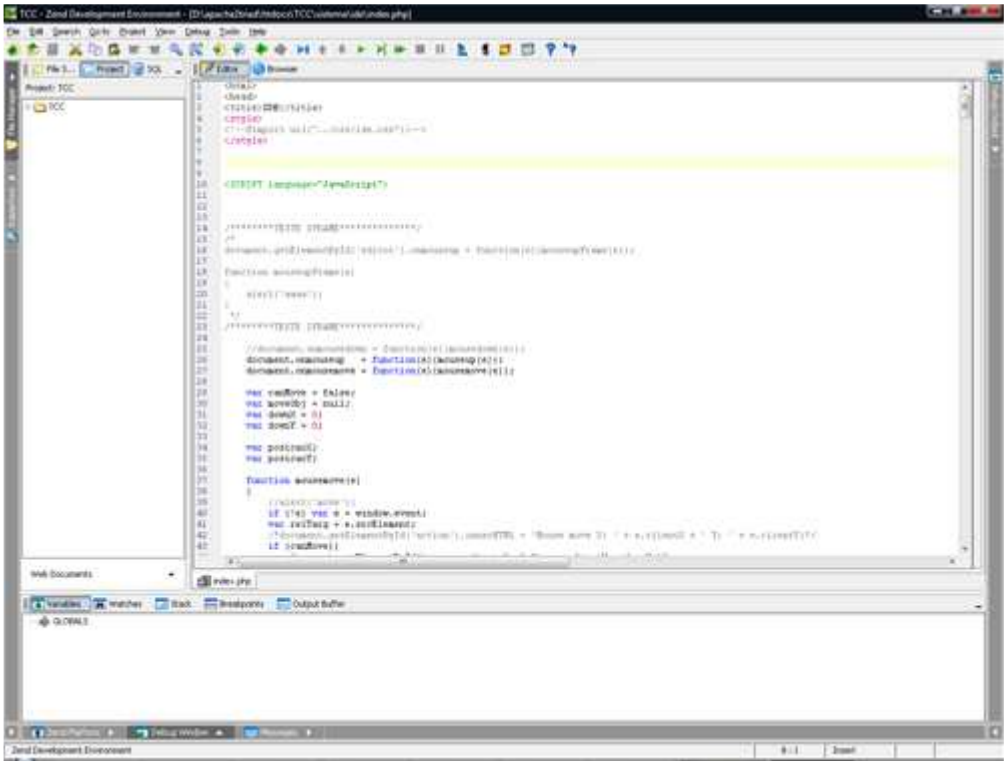


Figura 8 – Interface gráfica do Zend Studio

Na ferramenta desenvolvida é utilizado DHTML para permitir interações como arrastar componentes com mouse, abrir e fechar quadros dentro de uma página entre outros recursos. Também é utilizado AJAX no qual permite processar informações do servidor sem recarregar a página.

3 DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo apresenta o desenvolvimento da ferramenta W2PHP e os resultados obtidos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A atividade de levantamento de requisitos corresponde à etapa de compreensão do problema aplicada no desenvolvimento de software. O principal objetivo do levantamento de requisitos é de que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido.

Os requisitos funcionais determinam a funcionalidade do sistema. Esta ferramenta possui os seguintes requisitos funcionais:

a) permitir que o usuário crie interface para web através de uma IDE sem necessidade de codificação pelo usuário;
b) permitir que o usuário utilize funções (formatação de data, literal, números, etc) pré definidas da linguagem PHP através de uma ferramenta gráfica;
c) permitir que o usuário conecte os arquivos gerados em PHP com banco de dados MySQL e realize operações de inserção, modificação e exclusão de dados no banco de dados;
d) permitir a manutenção no metadados do banco de dados, como criar, editar e excluir tabelas e colunas;
e) ter uma ferramenta para manipulação de arquivos para criação, edição e exclusão de pastas no servidor;
f) permitir que o usuário faça *upload* de arquivos da máquina do usuário para o servidor;
g) permitir que o usuário manipule os componentes da ferramenta e os objetos criados dentro das páginas através da movimentação do *mouse*. O usuário deve conseguir arrastar componentes pela tela, acessar suas propriedades ao clicar e selecionar cada objeto com *mouse;*

Os requisitos não funcionais declaram as características de qualidade que o sistema

deve possuir e que estão relacionadas as suas funcionalidades. Esta ferramenta possui os seguintes requisitos não funcionais:

a) ser executada integralmente na web;
b) ser compatível com o navegador Mozilla Firefox;
c) desenvolver na linguagem PHP;
d) utilizar banco de dados MySQL.

3.2 ESPECIFICAÇÃO

A especificação da ferramenta foi feita através do Enterprise Architect para o desenvolvimento dos diagramas de classes, casos de uso e diagrama de atividades.

3.2.1 Diagrama de casos de uso

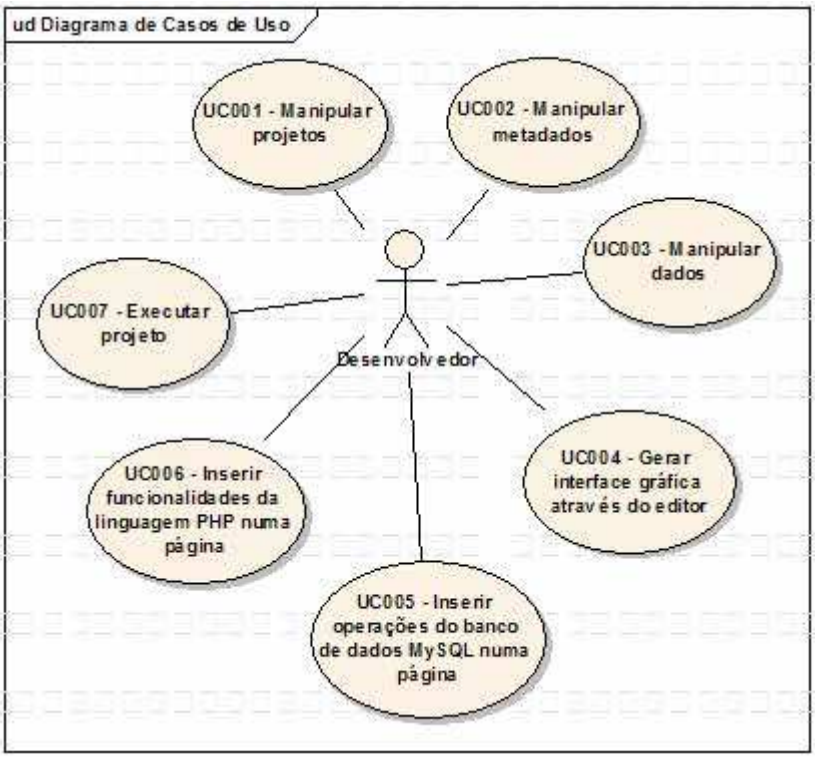


Figura 10 – Diagrama de casos de uso

Os quadros 7 a 13 apresentam a descrição dos casos de uso da ferramenta.

Manipular projetos
Sumário: O desenvolvedor efetua operações à definição do projeto, criado novos projetos, alterando e salvando suas configurações.
Ator primário: Desenvolvedor.
Fluxo principal: Criar novo projeto 1) O usuário acessa o menu correspondente ao novo projeto. 2) A ferramenta apresenta a tela correspondente às definições do novo projeto. 3) O usuário preenche o nome do projeto. 4) O usuário preenche o diretório do projeto. 5) O usuário clica no botão “Criar”. 6) A ferramenta verifica se todas as informações foram preenchidas corretamente e se o diretório digitado pelo usuário já não existe. 7) A ferramenta cria os diretórios correspondentes ao novo projeto. 8) A ferramenta cria o banco de dados correspondente ao projeto. 9) A ferramenta apresenta “Projeto criado com sucesso”. 10) Usuário clica em “Abrir Projeto”. 11) A ferramenta libera o editor para que o desenvolvedor possa criar e editar páginas.
Fluxo alternativo: Abrir projeto existente No passo 1, o usuário acessa o menu e seleciona a opção de abrir projeto. 1.1) O usuário acessa o menu correspondente a abrir projeto. 1.2) A ferramenta apresenta a lista de projetos já criados. 1.3) O usuário seleciona o projeto a ser aberto. 1.4) A ferramenta lista na aba “projeto” os arquivos do projeto. 1.5) Retorna ao passo 10.
Fluxo alternativo: Exclusão de projeto existente 1.3.1) O usuário clica no botão excluir ao lado do nome do projeto. 1.3.2) A ferramenta exclui os arquivos, diretórios e banco de dados correspondente ao projeto. 1.3.3) A ferramenta apresenta “Projeto excluído com sucesso”. 1.3.4) O usuário retorna ao passo 1.
Fluxo de exceção: Erro no preenchimento do formulário No passo 6, caso alguma informação esteja incorreta. 6.1) Apresenta o erro ao usuário informando erro em algum campo preenchido ou se o diretório digitado já existe. 6.2) Retorna ao passo 3.
Pós-condições: Os diretórios do projeto foram criados. O banco de dados do projeto foi criado.

Quadro 7 – Descrição do caso de uso UC001 – Manipular Projetos

Manipular metadados

Sumário:
Permite ao desenvolvedor realizar manutenção no metadados do banco de dados, como criar, editar e excluir tabelas e colunas.

Ator primário:
Desenvolvedor.

Fluxo Primário: Criar uma tabela

1) Usuário acessa o item “Banco de Dados” no menu.
2) A ferramenta apresenta uma tela onde são listadas as tabelas já criadas, caso existam, e um formulário para criação de uma nova tabela.
3) O usuário preenche o campo com o nome da tabela.
4) O usuário preenche o número de campos que a tabela deverá ter.
5) O usuário clica no botão “Criar”
6) A ferramenta verifica se o nome da tabela é válido, e se o número de campos é valido.
7) A ferramenta apresenta uma tela com o número de campos que o usuário escolheu para criar a tabela.
8) O usuário preenche um nome para cada campo a ser criado.
9) O usuário preenche o tipo de dado para cada campo a ser criado.
10) A ferramenta, quando o usuário preenche o tipo de dado a ser criado, preenche automaticamente o campo com o tamanho do campo a ser criado.
11) O usuário preenche o valor padrão do campo.
12) O usuário clica no botão “Criar”.
13) A ferramenta verifica se os nomes dos campos são validos.
14) A ferramenta verifica se os valores padrões correspondem ao tipo de dados selecionado pelo usuário.
15) A ferramenta apresenta mensagem “Tabela criada com sucesso”.

Fluxo alternativo: Editar uma tabela
No passo 2, após a ferramenta apresentar a tela onde são listadas as tabelas já criadas.
2.1) Em alguma tabela já criadas o usuário clica no botão “Editar tabela”.
2.2) A ferramenta irá listar os campos já criados pelo usuário e um formulário para a criação ou edição de um campo.
2.3) Retorna ao passo 3

Fluxo alternativo: Excluir uma tabela
No passo 2, após a ferramenta apresentar a tela onde são listadas as tabelas já criadas.
2.1) O usuário clica no botão “Excluir tabela” correspondente a tabela que o usuário deseja excluir.
2.2) A ferramenta exclui a tabela.
2.3) A ferramenta exibe a mensagem “Tabela excluída com sucesso”.
2.4) Retorna ao passo 2.

Fluxo alternativo: Limpar uma tabela
No passo 2, após a ferramenta apresentar a tela onde são listadas as tabelas já criadas.
2.1) O usuário clica no botão “Limpar tabela” correspondente a tabela que o usuário deseja limpar.
2.2) A ferramenta exclui todos os dados da tabela.
2.3) A ferramenta exibe a mensagem “Todos os dados da tabela foram excluídos com sucesso”.
2.4) Retorna ao passo 2.

Fluxo alternativo: Inserir um campo
No passo 2, após a ferramenta apresentar a tela onde são listadas as colunas já criadas na tabela.
2.2.1) O usuário preenche o formulário no fim da página com os dados do novo campo.
2.2.2) A ferramenta checa se todos os dados preenchidos estão corretos.
2.2.3) O usuário clica em “Inserir”.
2.2.4) A ferramenta cria o novo campo.
2.2.5) Retorna ao passo 2.2.

Fluxo alternativo: Editar um campo
No passo 2, após a ferramenta apresentar a tela onde são listadas as colunas já criadas na tabela.
2.2.1) O usuário clica no botão “Editar” ao lado do campo que deve ser editado.

2.2.2) A ferramenta preenche automaticamente o formulário no fim da página com os dados do campo. 2.2.3) O usuário modifica os dados do campo. 2.2.4) O usuário clica no botão “Editar”. 2.2.5) A ferramenta checa se todos os dados preenchidos estão corretos. 2.2.6) A ferramenta modifica o campo. 2.2.7) Retorna ao passo 2.2.
Fluxo alternativo: Excluir um campo No passo 2, após a ferramenta apresentar a tela onde são listadas as colunas já criadas na tabela. 2.3.1) O usuário clica no botão “Excluir campo” ao lado do campo que deve ser excluído. 2.3.2) A ferramenta exclui o campo. 2.3.3) Retorna ao passo 2.2.
Fluxo de exceção: No passo 7 caso algum dado digitado não corresponda ao tipo de dado da coluna. 7.1) A ferramenta apresenta a mensagem “Tipo de dado incorreto com o tipo de dado da coluna” 7.2) Retorna ao passo 5.
Fluxo de exceção: No passo 2.2.2 caso algum dado digitado não corresponda ao tipo de dado da coluna. 2.2.2.1) A ferramenta apresenta a mensagem “Tipo de dado incorreto com o tipo de dado da coluna” 2.2.2.2) Retorna ao passo 2.2.1.
Fluxo de exceção: No passo 2.2.2 caso algum dado digitado não corresponda ao tipo de dado da coluna. 2.2.2.1) A ferramenta apresenta a mensagem “Tipo de dado incorreto com o tipo de dado da coluna” 2.2.2.2) Retorna ao passo 2.2.1.
Pós-condições: O metadados do banco de dados foi alterado.

Quadro 8 - Descrição do caso de uso UC002 – Manipular Metadados

Manipular dados
Sumário: O desenvolvedor manipula dados nas tabelas criadas dentro do projeto.
Ator primário: Desenvolvedor
Pré-condições: Deve haver ao menos uma tabela criada.
Fluxo primário: Inserção de dados 1) Desenvolvedor acessa no menu a opção “Banco de dados”. 2) A ferramenta lista as tabelas criadas. 3) O usuário clica no botão “Dados” na tabela que deseja manipular dados. 4) A ferramenta lista os dados da tabela já cadastrados. 5) O desenvolvedor insere dados pelo formulário de inserção na tabela. 6) O desenvolvedor clica no botão “Inserir”. 7) A ferramenta verifica se os tipos de dados inseridos pelo usuário correspondem com os tipos de dados de cada campo inserido. 8) A ferramenta lista os dados da tabela já cadastrados com o novo dado inserido.
Fluxo alternativo: Alteração de dados 5.1) Num dado já inserido o usuário clica em “Alterar dado” . 5.2) Num formulário no fim da página a ferramenta carrega os dados correspondentes ao dado clicado pelo usuário. 5.3) O usuário altera os dados e clica no botão “Alterar”. 5.4) Retorna ao passo 7
Fluxo alternativo: Exclusão de dados 5.1) Num dado já inserido o usuário clica em “Excluir” . 5.2) Retorna ao passo 8
Fluxo de exceção: No passo 7 caso algum dado digitado não corresponda ao tipo de dado da coluna. 7.1) A ferramenta apresenta a mensagem “Tipo de dado incorreto com o tipo de dado da coluna” 7.2) Retorna ao passo 5.
Pós-condições: Os dados de uma tabela do banco de dados foram alterados.

Quadro 9 - Descrição do caso de uso UC003 – Manipular dados

Gerar interface gráfica através do editor
Sumário: Permite ao desenvolvedor criar e editar interfaces em HTML manipulando componentes.
Ator primário: Desenvolvedor
Pré-condições: Um projeto deve estar aberto.
Fluxo Primário: manipulando componentes 1) Desenvolvedor cria um novo arquivo ou abre algum já existente do projeto. 2) A ferramenta deixa o campo do editor livre pra edição. 3) O desenvolvedor utiliza as ferramentas de edição para construir a página. 4) O desenvolvedor salva o arquivo
Pós-condições: A interface gráfica foi gerada num arquivo PHP.

Quadro 10 - Descrição do caso de uso UC004 – Gerar interface gráfica com usuário

Inserir operações do banco de dados MySQL numa página
Sumário: O desenvolvedor arrasta elementos do quadro “MySQL” para dentro do editor.
Ator primário: Desenvolvedor
Pré-condições: Um projeto deve estar aberto. Deve haver ao menos uma tabela criada dentro do banco de dados do projeto.
Fluxo Primário: Inserir uma funcionalidade 1) O usuário seleciona um ícone de uma funcionalidade no quadro PHP e arrasta pro editor. 2) A ferramenta insere o ícone da funcionalidade no editor. 3) A ferramenta abre uma janela com as configurações da operação inserida. 4) A ferramenta salva as configurações da operação inserida.
Pós-condições: Foi inserido uma operação com banco de dados MySQL numa página representado por uma imagem no editor.

Quadro 11 - Descrição do caso de uso UC005 – Inserir operações do banco de dados mysql numa página.

Inserir funcionalidades da linguagem PHP numa página
Sumário: O desenvolvedor arrasta elementos do quadro “PHP” para dentro do editor.
Ator primário: Desenvolvedor
Pré-condições: Um projeto deve estar aberto.
Fluxo Primário: Inserir uma funcionalidade 1) O usuário seleciona um ícone de uma funcionalidade no quadro PHP e arrasta pro editor. 2) A ferramenta insere o ícone da funcionalidade no editor. 3) A ferramenta abre uma janela com as configurações da funcionalidade inserida. 4) A ferramenta salva as configurações da funcionalidade inserida.
Pós-condições: Foi inserida uma funcionalidade da linguagem PHP numa página representada por uma imagem no editor.

Quadro 12 - Descrição do caso de uso UC006 – Inserir funcionalidades da linguagem PHP numa página

Executar projeto
Sumário: Permite ao desenvolvedor executar o projeto criado pela ferramenta.
Ator primário: Desenvolvedor
Pré-condições: Um projeto deve estar aberto.
Fluxo Primário: Criar uma tabela 1) Usuário acessa o item “Executar” no menu. 2) A ferramenta verifica se existe algum arquivo chamado “index.php”. 3) A ferramenta apresenta mensagem “O projeto não possui nenhum erro”. 4) O desenvolvedor clica no botão “executar”. 5) A ferramenta abre uma nova janela executando o projeto do usuário.
Fluxo de exceção: Erro ao executar projeto 2.1) A ferramenta apresenta a mensagem “arquivo índex.php não encontrado”. 2.2) Retorna ao passo 1.
Pós-condições: Foi aberto uma nova janela com o projeto do usuário sendo executado.

Quadro 13 - Descrição do caso de uso UC007 – Executar projeto

3.2.2 Diagrama de classes

O diagrama de classes da ferramenta W2PHP está representado na figura 9.

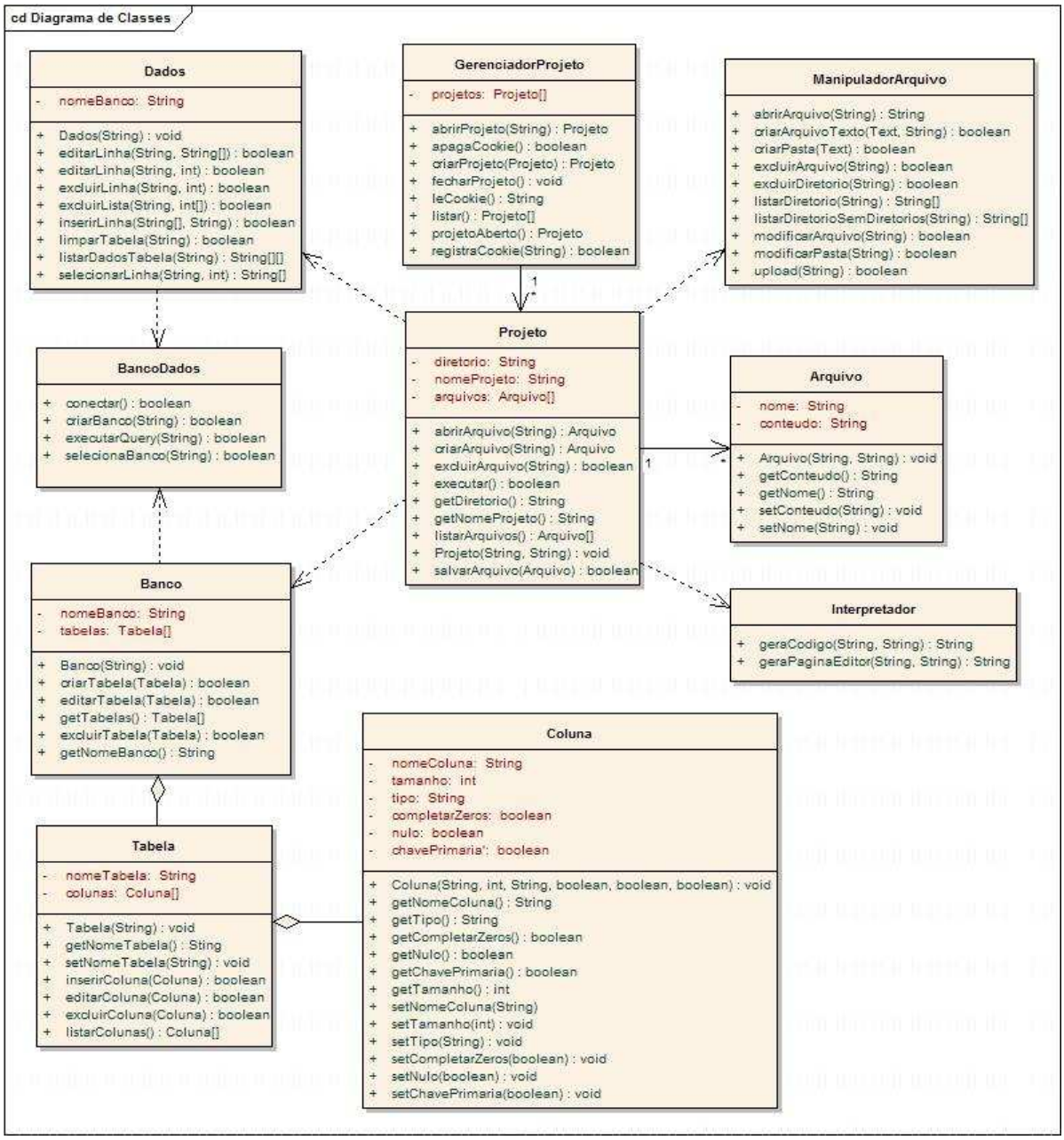


Figura 9 – Diagrama de classes da ferramenta W2PHP

Segue o detalhamento das classes relacionadas da figura 9, descrevendo o papel de cada uma delas:

a) `GerenciadorProjeto`: classe que gerencia os projetos. Permite a criação, listagem e exclusão de projetos e selecionar um projeto ser trabalhado.

b) `Projeto`: esta classe controla as configurações gerais do projeto como nome e diretório do projeto, listagem de arquivos;

c) `ManipuladorArquivo`: gerencia diretamente com funções de manipulação de arquivos do PHP permitindo a criação, edição e exclusão de arquivos, e também de diretórios.

d) `Arquivo`: armazena a estrutura de um arquivo, como o seu nome e seu conteúdo;

e) `Interpretador`: esta classe possui dois métodos, um responsável pela transformação do código recebido do editor para o código a ser executado, e outro que transforma o código a ser executado para o código do editor.
f) `BancoDados`: responsável pela execução de comandos SQL e conexão com banco de dados diretamente com funções do PHP;
g) `Banco`: guarda a estrutura de um banco de dados criado num projeto;
h) `Dados`: responsável pela execução de comandos que manipulem dados dentro de uma tabela;
i) `Tabela`: armazena a estrutura de uma tabela dentro de um banco;
j) `Coluna`: armazena toda a estrutura de uma determinada coluna de uma tabela.

3.2.3 Diagrama de atividades

Na figura 11, apresenta o diagrama de atividades da ferramenta.

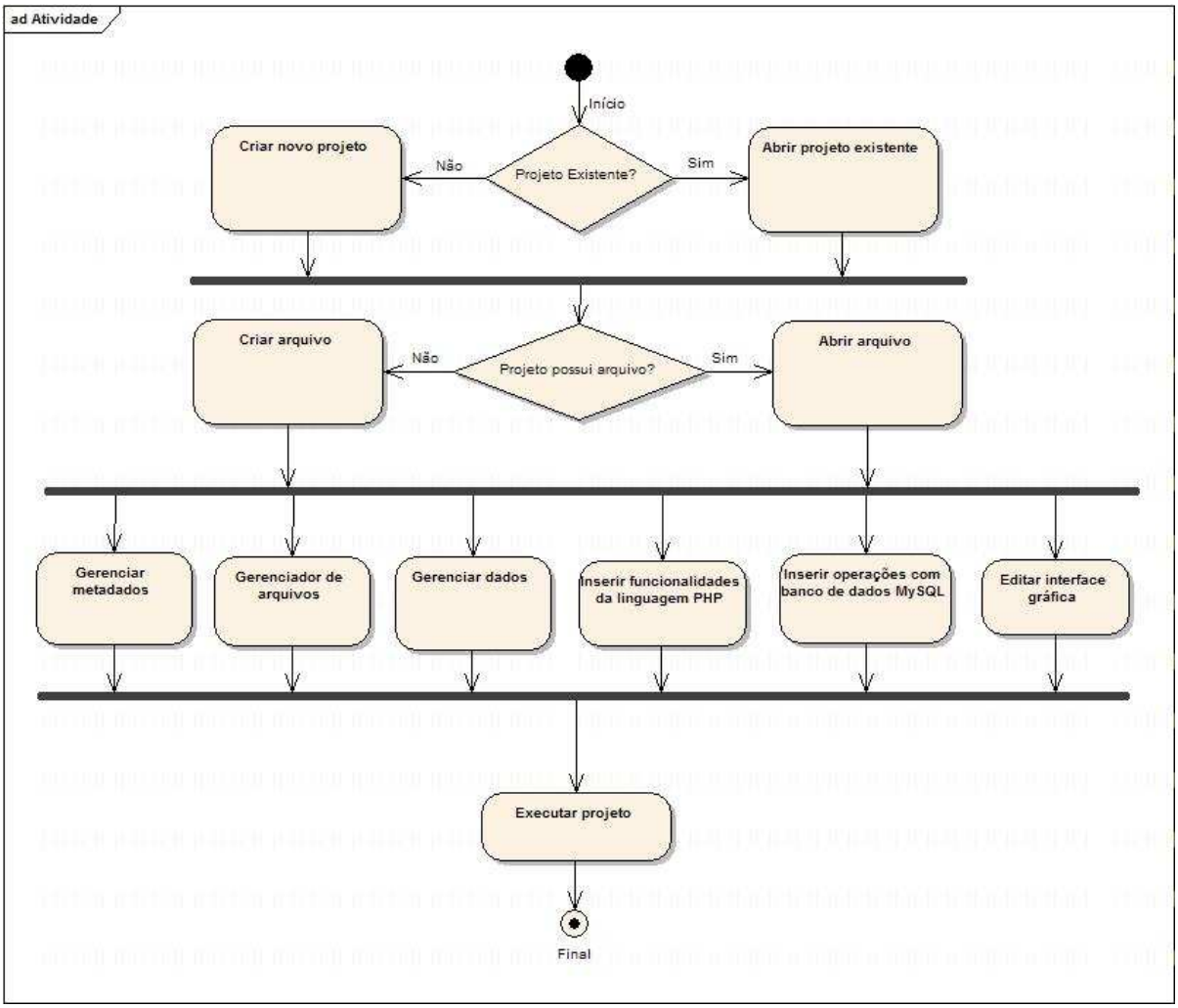


Figura 11 – Diagrama de atividades

Para poder criar páginas com interação com a linguagem PHP e banco de dados MySQL o usuário deverá criar ou abrir um projeto já existente. Após o projeto criado, o usuário poderá criar arquivos ou abrir arquivos já existentes. E nestes arquivos criar interface gráfica, inserir operações do banco de dados MySQL e funcionalidades da linguagem PHP.

O usuário poderá utilizar as ferramentas para manipular metadados e dados do banco de dados do projeto e utilizar o gerenciador de arquivos da ferramenta. E após a geração de um arquivo, poderá ser feito a execução do projeto.

3.3 IMPLEMENTAÇÃO

A ferramenta W2PHP foi desenvolvida na linguagem PHP 5.0 utilizando o ambiente de desenvolvimento Zend Studio na versão 5.5.0. No cliente o navegador utilizado para

executar a ferramenta é o Firefox 3.0 na sua versão beta. No servidor está sendo utilizado o Apache 2.0, com PHP 5.0 versão para Windows, e Mysql 5.0 também com versão para Windows.

Para a instalação, basta que os arquivos da ferramenta sejam copiados para a pasta onde ficam os arquivos a serem publicados no Apache. Dentro da pasta “include” existe o arquivo “conexao.php” que deve ser configurado o nome de usuário e senha do administrador do banco de dados MySQL. Esse administrador deve ter permissão de todas as operações com o banco de dados.

3.3.1 Implementação de manipulação de componentes

A ferramenta W2PHP, apesar de rodar integralmente via web, segue uma aparência de uma ferramenta *desktop*, para melhorar a usabilidade com o usuário. Para isso, foi desenvolvida uma biblioteca em DHTML que permite a manipulação de componentes dentro da tela, como arrastar e soltar componentes, abrir e fechar quadros.

Para tornar possível arrastar objetos na tela, foi utilizado funções do Java Script que permitem detectar quando o botão do *mouse* do usuário é acionado, quando ele é solto e quando ele é movido. O quadro 14 mostra o trecho de código três propriedades que detectam as ações do *mouse* por parte do usuário.

```
document.onmousedown = function(e){mousedown(e)};
document.onmouseup   = function(e){mouseup(e)};
document.onmousemove = function(e){mousemove(e)};
```

Quadro 14 – Manipulação de eventos de *mouse* através de Java Script

A propriedade `document.onmousedown` detecta quando o botão esquerdo do *mouse* do usuário foi apertado. Quando ocorre a ação é disparada a função `mousedown` na qual é verificado se o usuário clicou em algum objeto que deve ser movido. Caso sim, o objeto é liberado para poder ser arrastado na tela. Quando o mouse está sendo movido pela tela, a propriedade `document.onmousemove` dispara a função `mousemove` que calcula instantaneamente a posição do mouse na tela e também atualizando o valor de posição do objeto selecionado na tela, criando assim o efeito de arrastar o objeto pela tela. Quando o botão esquerdo do mouse do usuário é solto a propriedade `document.onmouseup` aciona a função `mouseup` que calcula a posição final do objeto e trava para ele não poder ser mais movido.

Para criar o efeito de abrir e fechar janela na própria página da ferramenta, sem usar *pop-ups*, ficando com toda a manipulação dentro da mesma janela, foi também utilizado DHTML utilizando funções do Java Script e propriedades do CSS.

A figura 12 mostra o exemplo quando a janela “Executar” é aberta.

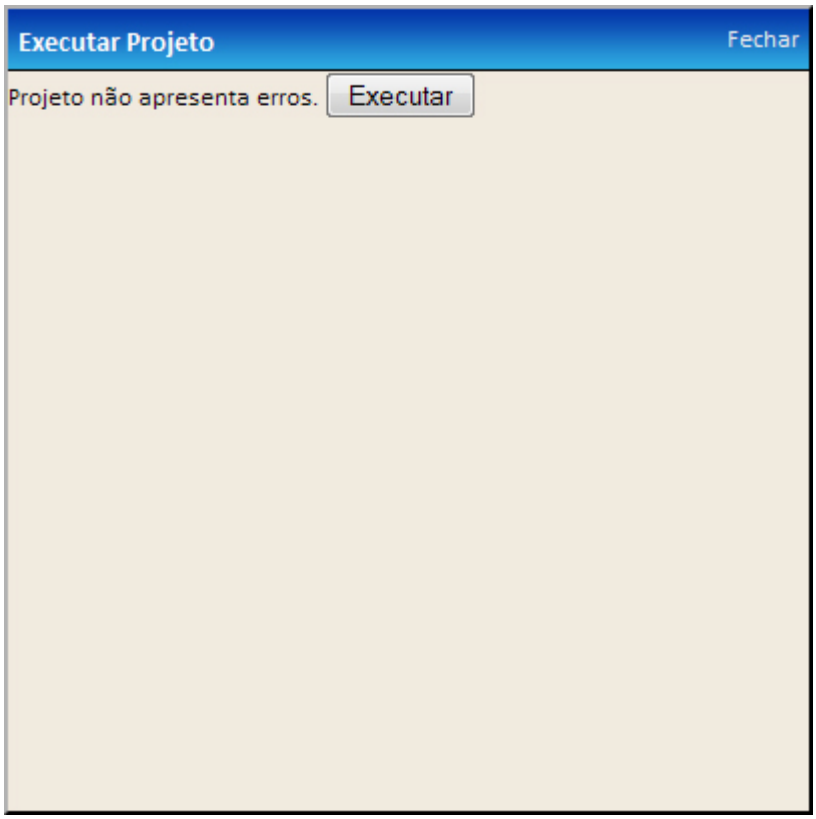


Figura 12 – Exemplo de janela aberta dentro da ferramenta.

Foi criado um `div` onde pelo CSS é definido para ficar invisível. Quando o usuário clica na opção “Executar” o Java Script modifica esta propriedade do quadro no CSS, definindo para ele ser visível. Assim a janela aparece para o usuário criando o efeito de abrir a janela. E para fechar, quando o usuário clica no *link* “Fechar”, o Java Script define no CSS, que a janela deve ficar invisível. O quadro 15 demonstra a função `abrirCaixa`, que permite a abertura de quadros dentro da ferramenta.

```
function abrirCaixa(posicaoX, posicaoY, tamanhoX, tamanhoY, qualDiv)
{
	if(document.getElementById(qualDiv).style.display=="none"                    ||
document.getElementById(qualDiv).style.display=="")
	{
		document.getElementById(qualDiv).style.display="block";
		document.getElementById(qualDiv).style.width=tamanhoX;
		document.getElementById(qualDiv).style.height=tamanhoY;
		document.getElementById(qualDiv).style.left=posicaoX;
		document.getElementById(qualDiv).style.top=posicaoY;
	}
	else
	{
		fecharCaixa(qualDiv);
	}
}
```

Quadro 15 – Código que permite que quadros sejam abertos dentro da ferramenta

A função `abrirCaixa` recebe como parâmetro as posição horizontais e verticais, a largura e a altura, qual div que deve ter estas propriedades modificadas e modifica para que caso o `div` esteja invisível na janela, muda a propriedade display no CSS do `div`, transformando em visível.

O quadro 16 mostra a função fecharCaixa que permite que um div aberto possa ser fechado.

```
function fecharCaixa(qualDiv)
{
	document.getElementById(qualDiv).style.display="none";
}
```

Quadro 16 – Código que permite que quadros sejam fechados dentro da ferramenta

A função `fecharCaixa` modifica a propriedades display no CSS do objeto, deixando invisível, criando assim o efeito de fechar janela.

Outra característica importante, é que alguns quadros abertos precisam processar e resgatar informações importantes como executar operações com banco de dados ou executar funções do PHP, sem poder atualizar a janela inteira da ferramenta. Para isto, em diversas situações foi utilizado AJAX, que será detalhado a seguir.

3.3.2 Utilização de AJAX na implementação da ferramenta

Para criar recursos utilizando AJAX deve ser criado objetos XMLHttpRequest que

permitem a atualização de um `div` assincronamente numa página. O quadro16 mostra quando um exemplo da instanciação do objeto.

```
try{
    xmlhttp = new XMLHttpRequest();
}catch(ee){
    try{
       xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    }catch(e){
       try{
           xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
      }catch(E){
         xmlhttp = false;
      }
   }
}
```

Quadro 16 – Instanciação do objeto XMLHttpRequest

Neste caso há uma tentativa de criar o objeto XMLHttpRequest na forma padrão em diversos navegadores como Mozilla Firefox, Konqueror, Safari e Opera. Mas o Microsoft Internet Explorer implementa através do ActiveX. Assim, no caso de falha, é feito uma tentativa de criá-lo compatível com o Internet Explorer. Podem ser criados mais de um objeto XMLHttpRequest numa página.

O quadro 17 demonstra a função para manipular quadros que necessitam utilizar AJAX.

```
function carregarAjax(objxmlhttp, qualDiv, arquivo)
{
      //Exibe o texto carregando no div conteúdo
      var conteudo=document.getElementById(qualDiv);

      conteudo.innerHTML='<font>Carregando janela</font>';
      objxmlhttp.open("GET", arquivo,true);

      //Executada quando o navegador obtiver o código
      objxmlhttp.onreadystatechange=function()
      {
            if (objxmlhttp.readyState==4)
            {
                  //Lê o texto
                  var texto=objxmlhttp.responseText;

                  //Desfaz o urlencode
                 // texto=texto.replace(/\+/g," ");
                  texto=unescape(texto);

                  //Exibe o texto no div conteúdo
                  var conteudo=document.getElementById(qualDiv);
                  conteudo.innerHTML=texto;
            }
     }
     objxmlhttp.send(null)
}
```

Quadro 17 – Função Java Script para manipular quadros com AJAX

A função `carregaAjax` recebe como parâmetro o objeto XmlHttpRequest criado, o `div` que deve ser atualizado, e o arquivo PHP que deve ser aberto neste quadro. O conteúdo do `div` é alterado para “Carregando janela”. Através da propriedade `readyState`, é controlado o status da solicitação. Quando o valor recebido é “4” significa concluída, e o conteúdo recebido é atualizado no `div`.

3.3.3 Implementação do editor de texto

Para que o usuário pudesse formatar textos, inserir imagens, tabelas, listas entre outros recursos do HTML, um `div` chamado editor foi definido para que pudesse ser editado. A forma de implementação para que um `div` possa ser modificado é diferente entre os navegadores. O quadro 18 demonstra como na ferramenta W2PHP, foi implementado.

```
function iniciarEditor()
{
	document.getElementById("editor").contentDocument.designMode="on";
}
```

Quadro 18 – Definição de um div para que possa ser modificado

Toda vez que o arquivo que contém a ferramenta é carregado, é executada função `iniciarEditor` que define o div chamado “editor” para que possa ter seu conteúdo modificado.

Para que o div editor possa ter as propriedades dos seus textos e conteúdos inseridos modificados foi utilizado o método `execCommand` do Java Script para permitir que num `div` fosse inserido comandos para edição de texto. A implementação do método também é diferente entre os navegadores, e na ferramenta W2PHP, foi utilizado a implementação padrão para Mozilla Firefox. O quadro 18 mostra um exemplo da utilização do método.

```
function comandoEditor(comando,flag,valor)
{
	document.getElementById('editor').contentDocument.execCommand(comando,
false, valor);
}
```

Quadro 19 – Utilização do método execCommand

Quando algum botão do editor como, por exemplo, o “negrito” for acionado, é executada a função `comandoEditor`, que tem como parâmetro o comando que no caso será `“bold”`, a *flag* que é utilizada por alguns comandos, neste exemplo é nulo, e o valor que também é utilizado para alguns comandos. O texto selecionado pelo usuário se transformará em negrito. O editor funciona da mesma forma que diversos editores de textos convencionais, mas sendo executado via web.

3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO

A figura 13 apresenta uma visão geral da ferramenta. No menu da ferramenta há opções para a de criação de projeto, arquivos, manipulação de banco de dados, gerenciador de arquivos, entre outros (1). Há uma barra de ícones com atalhos para algumas funcionalidades mais utilizadas (2). A aba das informações do projeto fica localizada na direita, com os arquivos que fazem parte do projeto aberto (3). Abaixo fica localizada a aba de funcionalidades da linguagem PHP que podem ser inseridas na página (4) e a aba de

operações MySQL que podem ser inseridos na página (5). O editor da página, onde é permitido que o usuário altere as fontes, tamanhos de letras, cores, cores de fundo, criar tabelas, títulos, listas, entre outros (7).

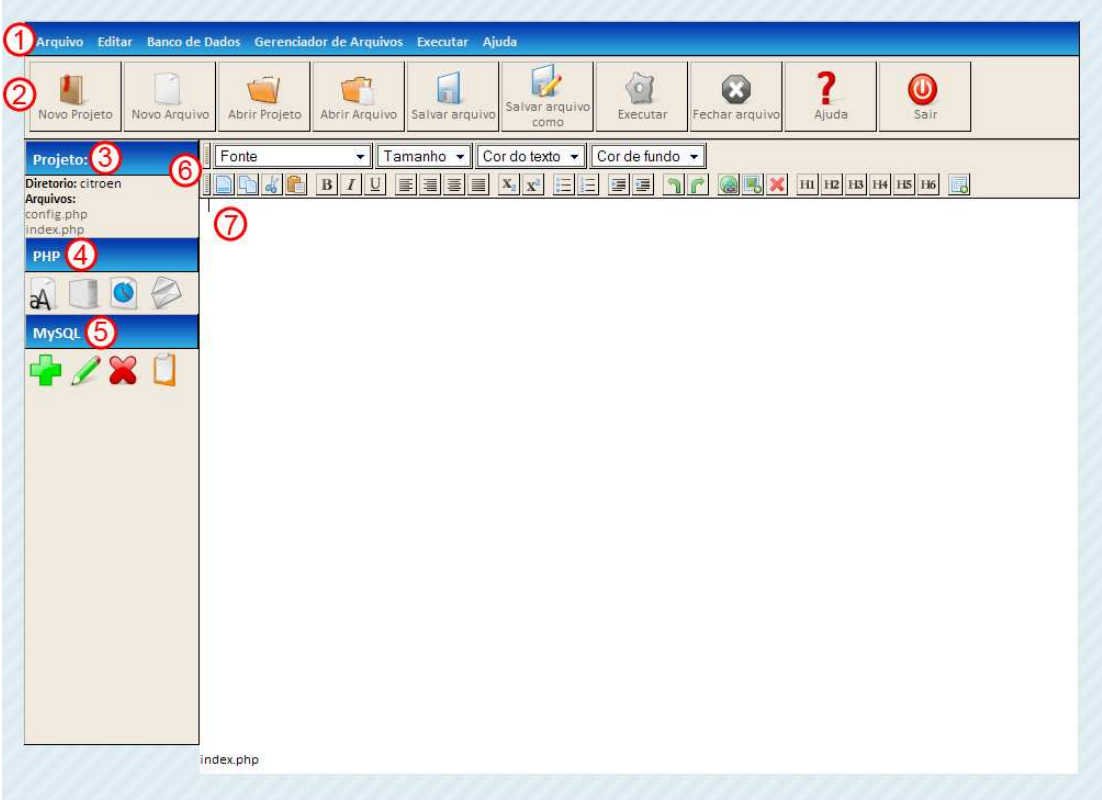


Figura 13 – Visão geral da ferramenta

Para demonstrar o funcionamento da ferramenta foi criado um caso, simulando um caso de uma concessionária de carros que deseja publicar seu estoque para a internet.

O usuário deve criar um projeto ou abrir um projeto caso já exista. Na figura 14 demonstra um novo projeto criado com nome “Citroen” e com um diretório chamado “citroen”.

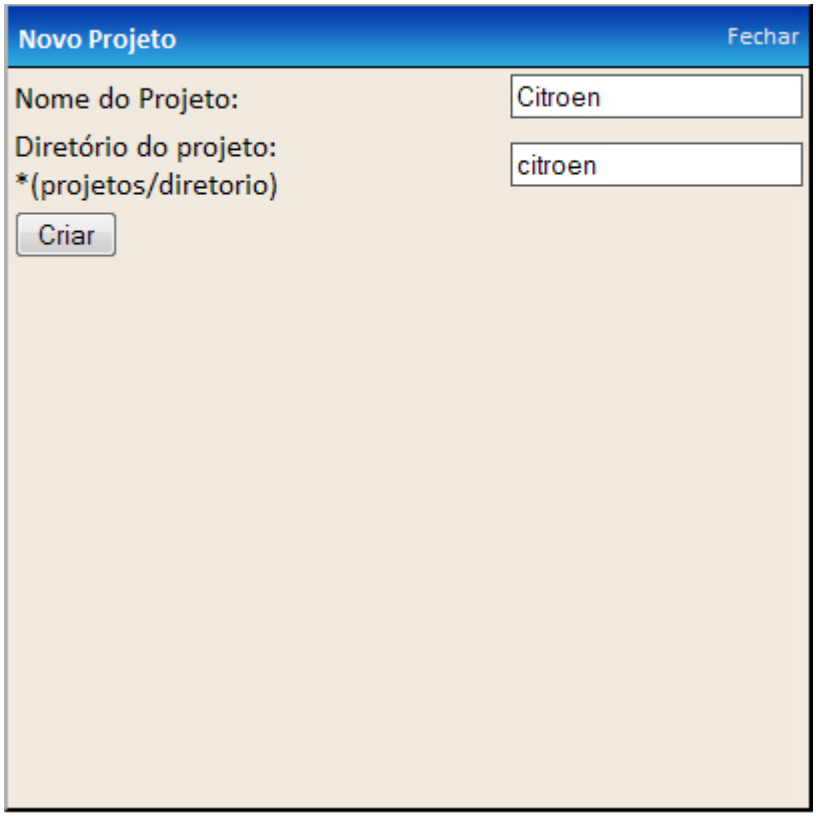


Figura 14 – Criação de um novo projeto

Após o projeto ser criado, o usuário poderá criar arquivos que satisfaçam suas necessidades. Na figura 15 apresenta o usuário criando um arquivo chamado “index.php”, que será a capa do projeto.

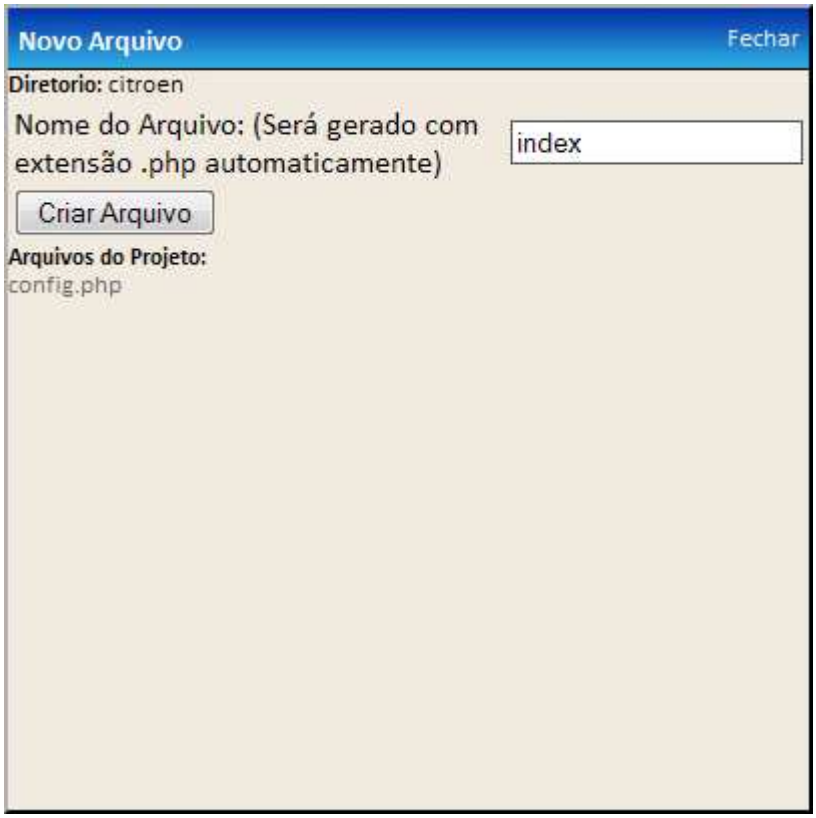


Figura 15 – Criação de um novo arquivo

Com o arquivo criado e aberto o usuário pode começar a editar o conteúdo da página. O editor possui diversos recursos para edição de texto, inserção de imagens, listas e tabelas. Para inserir imagens o usuário pode copiar imagens de páginas fora do projeto ou utilizar arquivos do projeto dentro do gerenciador de arquivos. O gerenciador permite que os arquivos sejam enviados da máquina do usuário até o servidor. O exemplo da figura 16 ilustra um *upload* de imagem da máquina do usuário até o servidor.

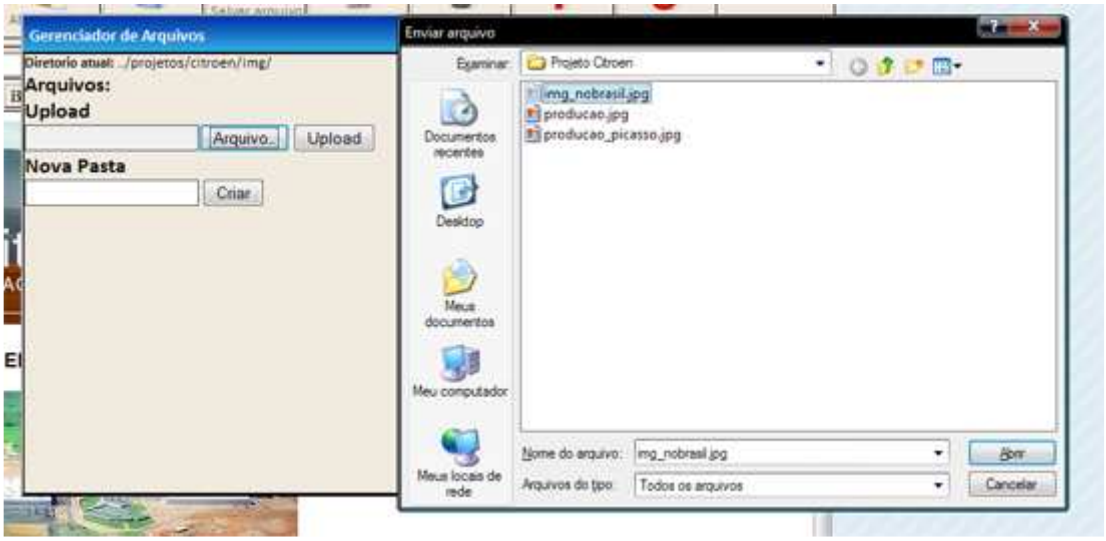


Figura 16 – Exemplo de envio de imagens para o servidor

Quando o usuário quiser inserir no editor alguma imagem já enviada ao gerenciador, o

usuário deve clicar em “Selecionar” para adicionar os arquivos no editor.

A figura 17 mostra a criação de uma nova pasta no gerenciador de arquivos.

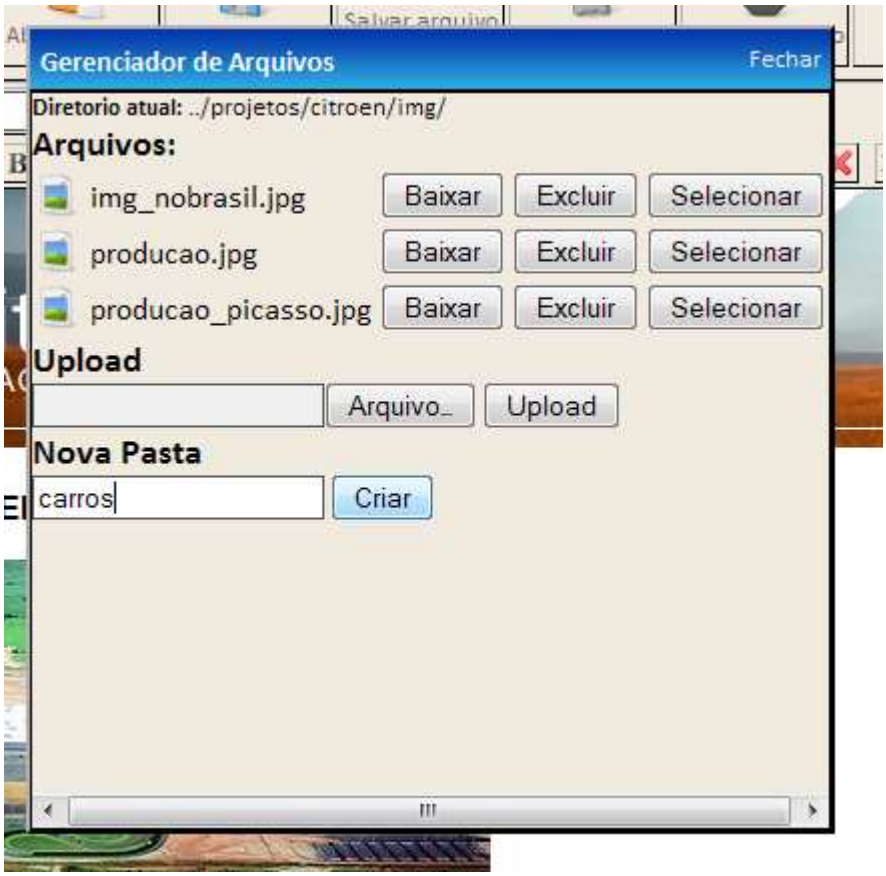


Figura 17 – Criação de nova pasta

Outro recurso importante e muito utilizado nas páginas web são as tabelas. As tabelas servem tanto para estruturar os conteúdos nas páginas, como também tabular informação. A figura 18 mostra o quadro de configuração de uma nova tabela dentro do editor.

Figura 18 – Criação de uma nova tabela dentro do editor.

A figura 19 mostra o editor após o usuário inserir algumas imagens e textos. A ferramenta permite que o usuário selecione fontes para a edição do texto (1). Na lista aparecem apenas fontes comuns em diversos sistemas operacionais, fazendo com que as páginas criadas pelos usuários abram iguais nos diferentes sistemas operacionais e navegadores.

Figura 19 – Edição de texto no editor da ferramenta

O usuário pode criar páginas internas e no exemplo da figura 20, mostra a lista de arquivos criados no projeto. O usuário para editar estes arquivos, basta clicar no nome do arquivo para que ele seja aberto no editor para poder realizar sua edição.

Figura 20 – Lista de arquivos do projeto

A ferramenta também possui a opção para administrar banco de dados. Nela o usuário pode realizar diversas operações. A figura 21 apresenta a criação de uma tabela que armazenará o estoque com sete campos definidos pelo usuário.

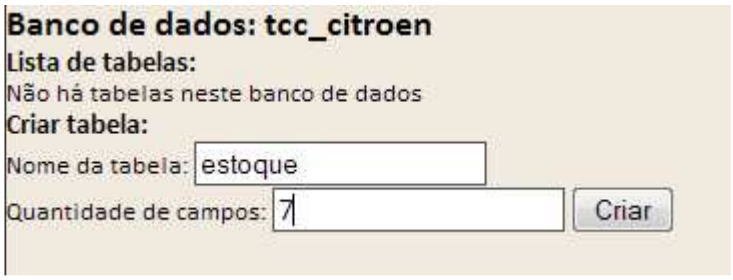


Figura 21 – Criação de tabela na ferramenta de administração de banco de dados

Na figura 22 mostra a tela com os campos preenchidos da nova tabela. O usuário

define o nome do campo, o tipo de dado, valor padrão, se é não nulo e se deve preencher zeros nos campos numéricos. O tamanho do dado é definido automaticamente quando o usuário seleciona o tipo do dado.

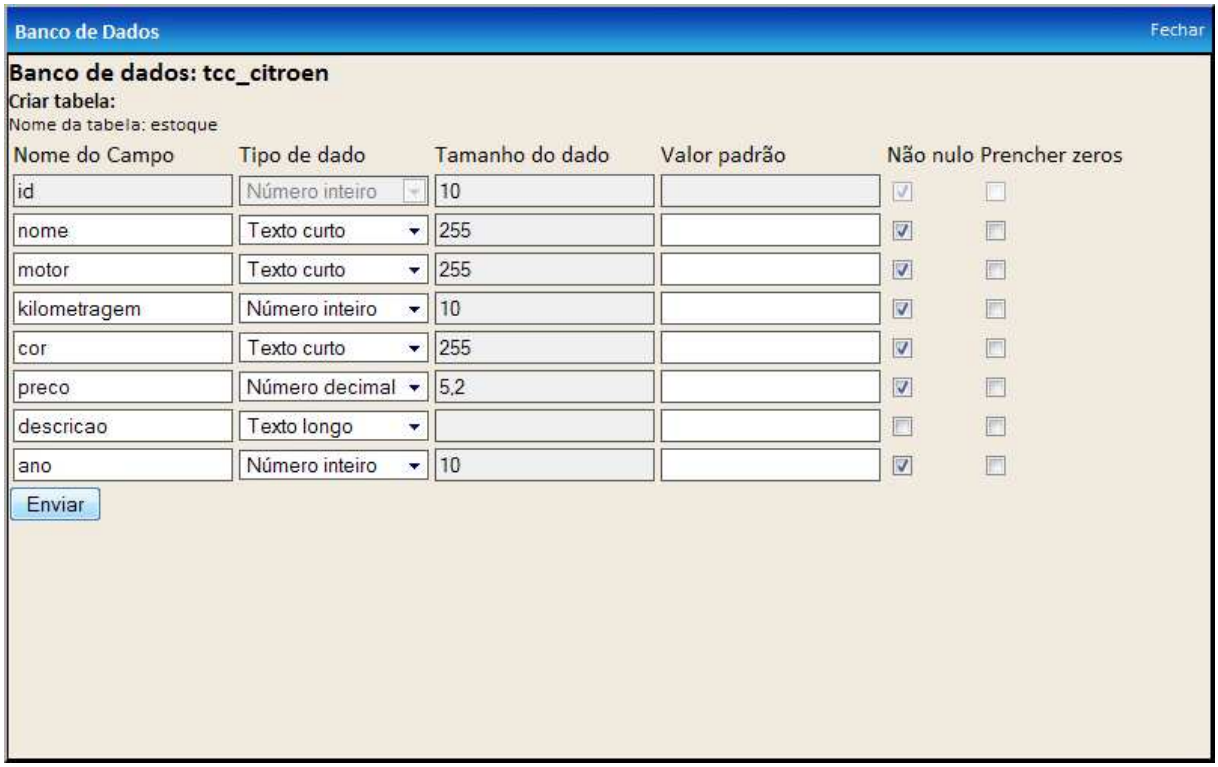


Figura 22 – Criação das colunas na tabela criada pelo usuário

O usuário pode escolher 8 tipos de dados diferente. O tipo “caracter” para colunas de texto com tamanho 1, “número inteiro” para colunas de número inteiro, “número decimal” para colunas de número decimal, “texto curto” para colunas com texto de até 255 caracteres, “texto longo” para textos grandes sem limitação, “data” para campos de somente data, “hora” para campos que possuem somente hora e “data e hora” para campos com data e hora juntos.

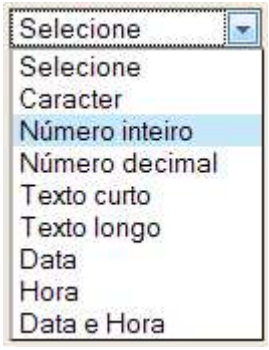


Figura 23 – Tipo de campos a serem criados nas tabelas do projeto

Após a tabela ser criada ela será aparecerá na ferramenta de administração de banco de dados como mostra na figura 24. O usuário poderá alterar suas colunas, administrar os dados da tabela, limpar os dados e excluir a tabela.

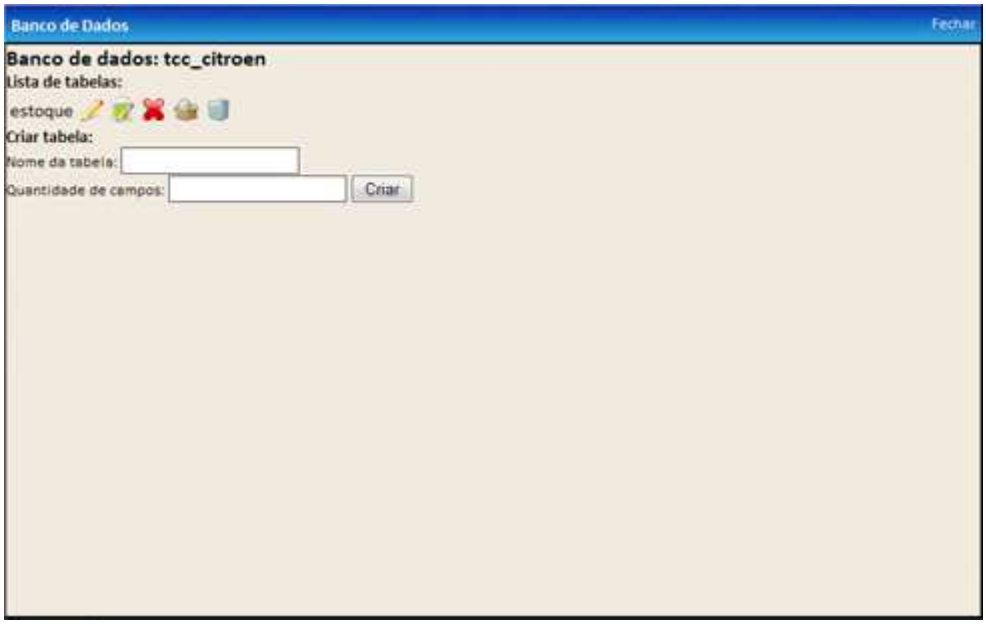


Figura 24 – Lista de tabelas criadas pelo usuário

Na figura 25 apresenta a estrutura da tabela, que pode ser editada. O usuário pode inserir uma nova coluna, ou ao clicar num “lápis” que é o ícone de edição da coluna que é carregada automaticamente no formulário os dados deste campo para que o usuário possa realizar alterações. E o campo pode ser excluído também.

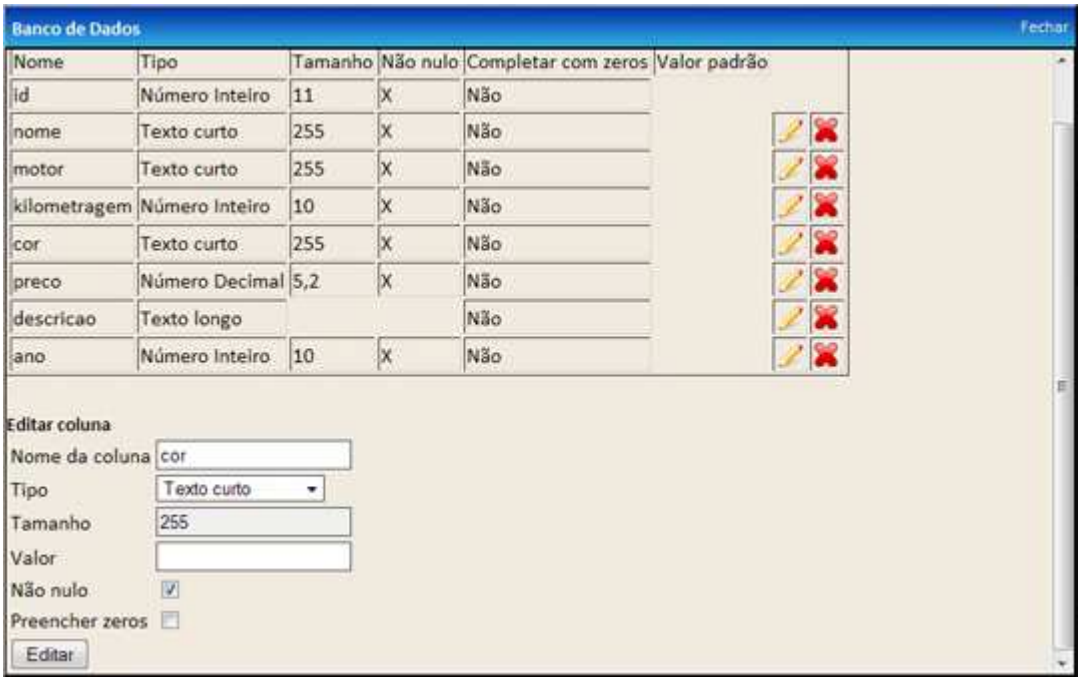


Figura 25 – Edição de tabelas já criadas

O usuário também pode inserir, editar e excluir dados dentro da tabela. Da mesma forma que a edição de colunas, o usuário no formulário pode inserir dados, ou clicando no símbolo demonstrado com um “lápis”, é carregado os dados da linha selecionada no formulário e o usuário poderá fazer a edição dos dados. Na figura 26 mostra a manipulação de

dados dentro da tabela.

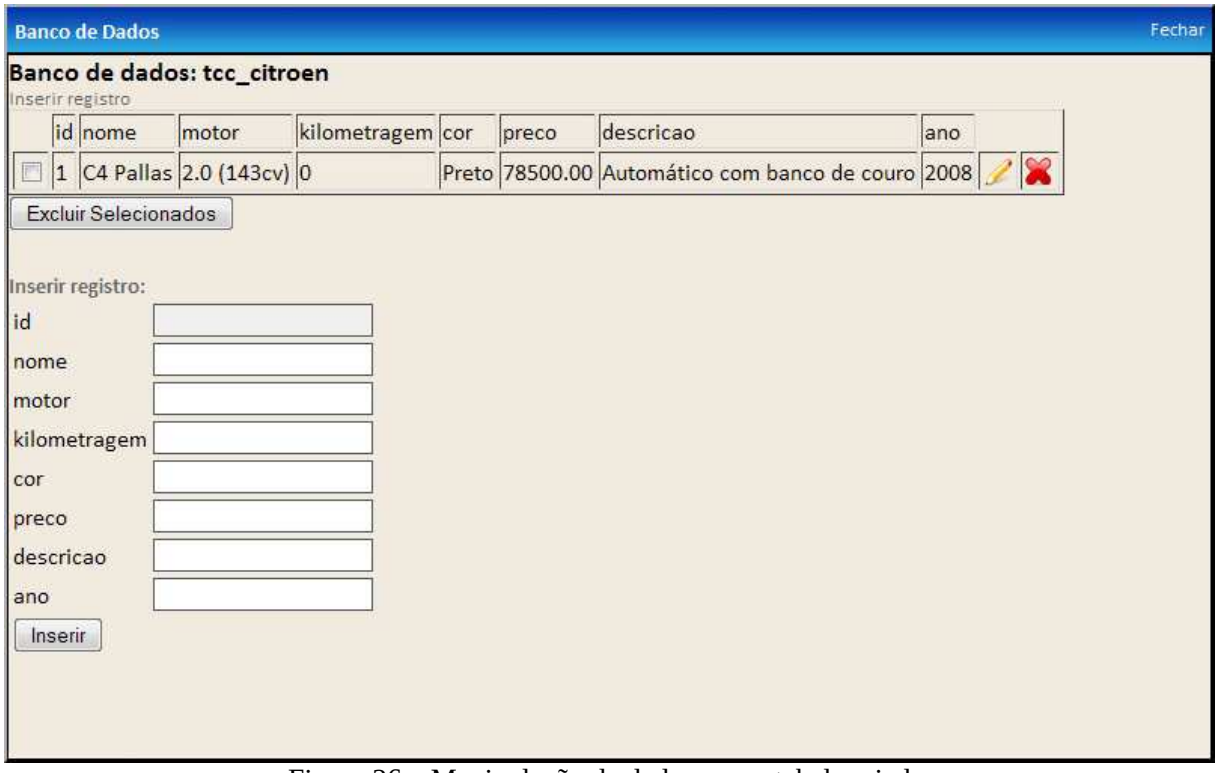


Figura 26 – Manipulação de dados numa tabela criada.

Na figura 27, mostra a página “estoque.php” que mostrará a listagem do estoque da concessionária. O usuário clica no ícone de listagem da aba de operações com MySQL (1), aparecerá um quadro com configurações da listagem representado na figura 28 e automaticamente será inserido no editor o ícone correspondente a listagem de dados (2).

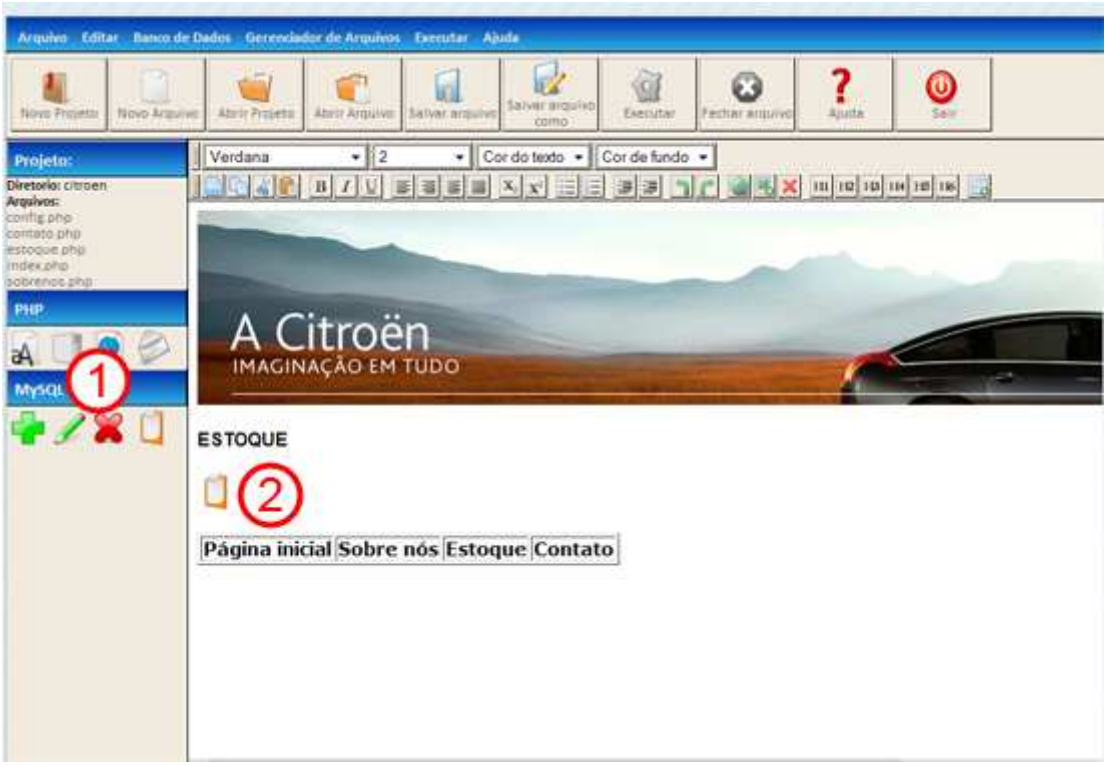


Figura 27 – Página que listará o estoque da concessionária

A figura 28 mostra a configuração da listagem do estoque da concessionária. O usuário escolhe qual tabela deseja listar e logo após quais campos deve aparecer na listagem.

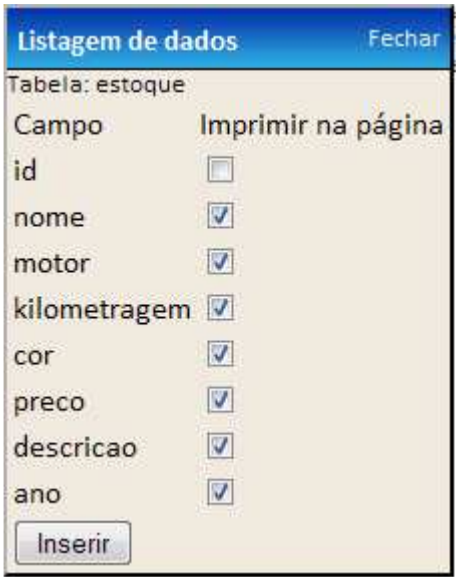


Figura 28 – Configurações de listagem de dados

O usuário para executar o projeto deve selecionar no menu a opção “Executar projeto”. Na figura 29 apresenta o quadro caso não exista nenhum erro no projeto a ferramenta apresentação a mensagem “Projeto não apresenta erros”, caso apresente, a ferramenta listará os erros. No exemplo não existe nenhum erro, e o usuário clicando em “Executar” abre uma nova janela com a execução do projeto.

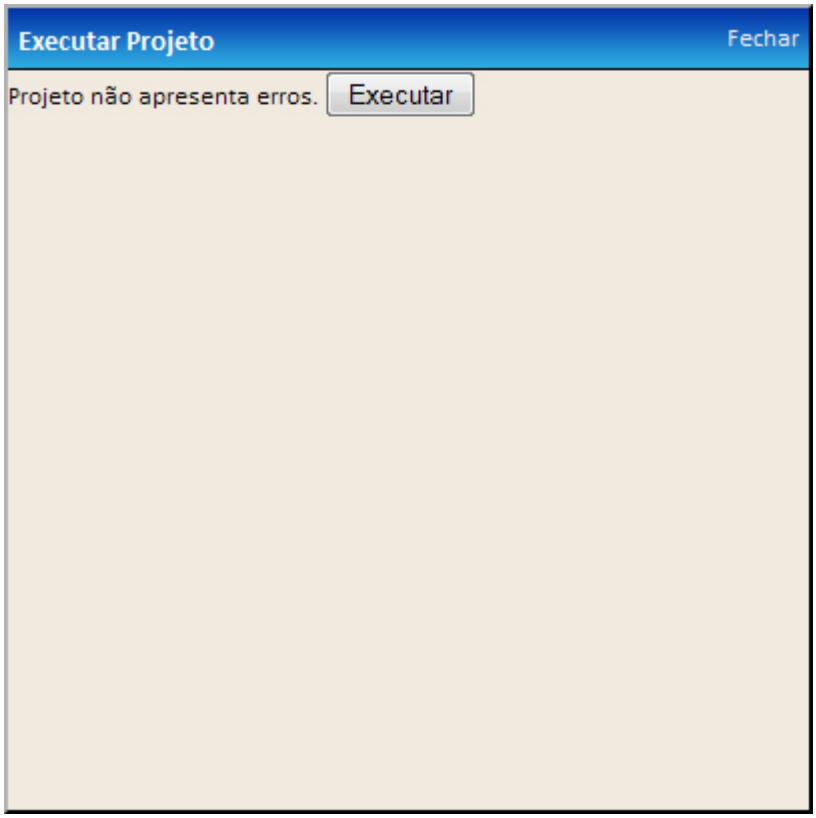


Figura 29 – Execução de um projeto.

Na figura 30, apresenta a página inicial do projeto com as imagens, textos, tabelas e *links* definidos pelo usuário.

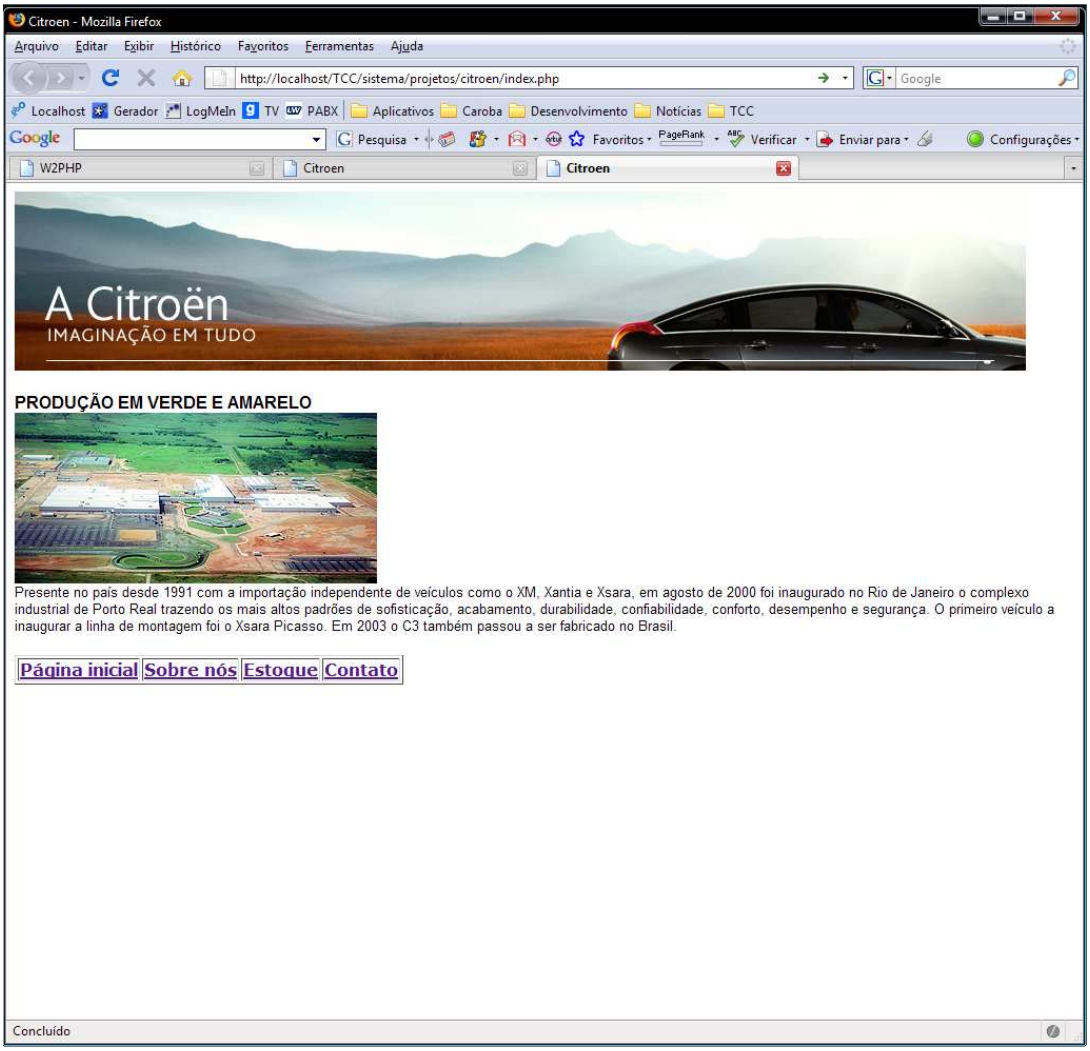


Figura 30 - Capa do projeto

Na figura 31, mostra a página de estoque criado pelo usuário. A página apresenta a listagem da tabela estoque, com os campos selecionados pelo usuário.

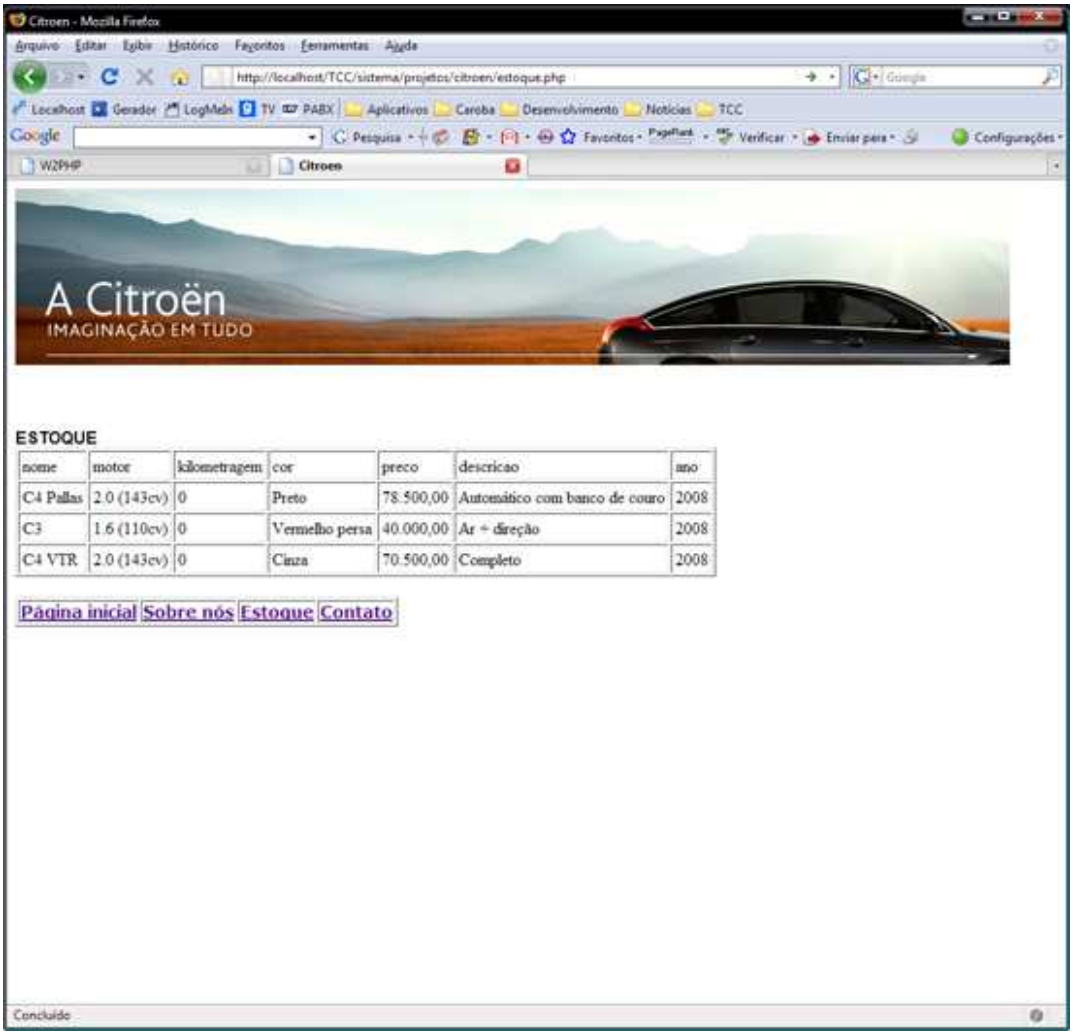


Figura 31 – Página de estoque criada pelo usuário

3.5 RESULTADOS E DISCUSSÃO

No período de desenvolvimento da ferramenta, mesmo não sendo um requisito, procurou-se tornar compatível com todos os navegadores. Porém alguns testes foram realizados no navegador Microsoft Internet Explorer e foram identificados inúmeros problemas de compatibilidade de recursos, principalmente no editor HTML onde diversos recursos têm instâncias diferentes entre os navegadores.

Um exemplo é a forma de permitir que um `div` possa ser editado numa página que é completamente diferente entre o navegador Internet Explorer e Mozilla Firefox. E os recursos suportados pelo editor pela função `execComand` são diferentes entre os diversos navegadores.

Koch (2007) afirma que existem propriedades que tem compatibilidade totalmente diferente entre os navegadores. Um exemplo, é a propriedade `indent` que no Microsoft Internet Explorer 6 e Opera 9 estão implementados de forma incorreta, no Safari 3 está implementado mas apresenta erros e apenas no Firefox 2 e versões superiores está implementado de forma correta. Além disso, algumas propriedades do `execComand` no Firefox apresentam problemas conhecidos, que dificultou o desenvolvimento de alguns recursos no editor da ferramenta.

Em virtude destes problemas, a ferramenta foi baseada apenas no funcionamento para o navegador Mozilla Firefox 2 ou superior. Porém com a evolução dos navegadores, futuramente, pode-se se tornar com facilidade compatível com outros navegadores necessitando modificações em alguns trechos de código na biblioteca DHTML da ferramenta.

As funções que utilizam o objeto XMLHttpRequest para o funcionamento de recursos baseados em AJAX, estão compatível com todos os navegadores.

Diante que nenhum trabalho correlato permitia personalização das páginas geradas o recurso de personalização da interface gráfica foi o grande diferencial da ferramenta W2PHP. É utilizado um conceito diferente para inserir funcionalidades de banco de dados dentro das páginas geradas. As tabelas com suas respectivas colunas e dados podem ser manipulados através da ferramenta para administração do banco de dados.

Havendo alguma tabela criada o usuário pode arrastar componentes no quadro com as funcionalidades do MySQL até o editor e configura as operações que a ferramenta deve realizar com as tabelas criadas. Na execução do projeto as operações com banco de dados MySQL serão realizadas. Da mesma forma alguns recursos da linguagem PHP estão disponíveis para serem utilizados dentro das páginas.

A ferramenta CodGer desenvolvida por Menin (2005) permite que seja criada a geração automática de código para a tecnologia JSP através da leitura da estrutura armazenada em um banco de dados MySQL. Inicialmente o desenvolvedor precisa ter definido a estrutura do dicionário de dados no banco de dados MySQL. Feita a especificação do modelo de dados, o desenvolvedor faz a conexão com o banco de dados e a ferramenta faz a leitura da estrutura de tabelas, campos e tipos. Após a estrutura ser lida pela ferramenta, o usuário pode realizar a geração de código na linguagem JSP para relatórios e formulários de cadastros, alterações e exclusões.

A ferramenta ASP.NET Generate desenvolvida por Heiden (2005), lê o dicionário de dados do SQL Server 2000 e gera páginas na linguagem ASP.Net que permitem cadastrar páginas, especificar os tipos de acessos que terão cada tabela como consulta, cadastro,

alteração e exclusão.

A ferramenta Netsis desenvolvida por Castilhos (2004) tem a finalidade de definir, documentar e armazenar informações necessárias para a geração do banco de dados SQL Server com os relacionamentos entre as tabelas, assim garantindo a integridade referencial. Após a geração do banco de dados, é possível a geração de código na linguagem ASP da parte para a administração das informações do site com cadastros completos incluindo opções de inclusão, alteração, exclusão, localizar e consultar por campo, paginação dos registros e hierarquia de campos. A Netsis assemelhasse com a W2PHP, no modo em que é executado, integralmente via web.

Em razão da especificidade da ferramenta desenvolvida neste trabalho optou-se por não criar um quadro comparativo de características.

4 CONCLUSÕES

Os objetivos inicialmente propostos foram contemplados. A ferramenta W2PHP foi desenvolvida com o intuito de que usuários leigos em programação conseguissem criar páginas na Web com dinamismo e com interação com banco de dados. Ou que programadores experientes usem a ferramenta para acelerar o processo de desenvolvimento de software.

O grande diferencial da ferramenta é a execução integralmente via web, tendo como requisito apenas que o usuário tenha instalado no seu computador o Mozilla Firefox 2.0 ou superior. Através da utilização de DHTML e AJAX, a ferramenta apresenta caixas de configuração semelhantes a ferramentas *desktop*. Os componentes disponibilizados na ferramenta apresentam interação através de quadros e menus, entre outros objetos dinamicamente criados. Além disso, informações são requisitadas e carregadas a partir do servidor, de forma implícita.

O editor de texto desenvolvido permite que o usuário crie páginas personalizando o conteúdo, inserindo imagens e alguns recursos do HTML, sem nenhum contato com código fonte. Arrastando o mouse com componentes do PHP ou MySQL o usuário consegue inserir funcionalidades da linguagem de programação PHP e interações com banco de dados MySQL.

4.1 EXTENSÕES

Sugere-se como extensões para a ferramenta apresentada:

a) permitir que a ferramenta seja compatível com outros navegadores além do Mozilla Firefox;
b) permitir que os arquivos PHP gerados consigam interagir com outros banco de dados além do MySQL;
c) permitir que possam ser inseridos funcionalidades do MySQL além de inserção, atualização, exclusão e listagem de dados numa página. Também permitir o relacionamento entre tabelas;
d) permitir que o usuário ao enviar uma imagem no gerenciador de arquivo, possa fazer pequenas alterações na imagem ajustando o seu tamanho diretamente no

arquivo e não só no editor alterando no código HTML;

e) aumentar o número de recursos no editor;

f) adicionar recursos prontos como, por exemplo, um módulo de comércio eletrônico que facilite e deixe muito mais rápido para que o desenvolvedor crie um *site* de comércio eletrônico;

g) permitir importar banco de dados de outro banco criado por outras ferramentas ou pelo próprio desenvolvedor, e exportar os bancos de dados criados pela ferramenta.

REFERÊNCIAS BIBLIOGRÁFICAS

ASLESON, Ryan; SCHUTTA, Nathaniel. **Fundamentos do AJAX**. Rio de Janeiro: Alta Books, 2006.

BOLONHA, João C.; HAMPSHIRE, Paulo; LEÃO, Marcelo. **DELPHI 8**: para plataforma .NET. Rio de Janeiro: Axcel Books do Brasil, 2004.

BORBA, Fernando E. **AJAX**: guia de programação. Tatuapé: Érica, 2006.

CANTÚ, Marco. **DEPLHI 7**: a bíblia. São Paulo: Pearson Education do Brasil, 2003.

CASTILHOS, Cristiano. **Ferramenta CASE para geração de páginas ASP**. 2004. 73 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <http://www.bc.furb.br/docs/MO/2004/279064_1_1.pdf>. Acesso em: 12 maio 2008.

FERREIRA, Élcio. **DHTML crossbrowser**: um guia rápido para desenvolvedor. [S.l.], 2004. Disponível em: <http://elcio.com.br/crossbrowser>. Acesso em: 15 maio 2008.

GARRET, James. **AJAX**: a new approach to web applications. [S.l.], 2006. Disponível em: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>. Acesso em: 18 maio 2008.

GOOGLE INC. **Google maps.** [S.l.], 2004. Disponível em: <http://www.google.com.br/maps>. Acesso em: 18 maio 2008.

HEIDEN, André C. **Ferramenta de geração de código a partir do dicionário de dados do SQL Server para tecnologia ASP.NET**. 2005. 83 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <http://www.bc.furb.br/docs/MO/2005/305464_1_1.pdf>. Acesso em: 5 maio 2008.

KOCK, Paul P. **execComand compatibility.** [S.l.], 2007. Disponível em: <http://www.quirksmode.org/dom/execCommand.html>. Acesso em: 26 maio 2008.

LIMEIRA, José L. S. **Utilização de AJAX no desenvolvimento web**. 2006. 44 f. Trabalho de Conclusão de Curso (Especialização em Web e Sistemas de Informação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: <http://www.limeira.eti.br/monografia_ajax.pdf >. Acesso em: 15 maio 2008.

MENIN, Juliane. **Gerador de código ASP baseado em projeto de banco de dados MySQL**. 2005. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <http://www.bc.furb.br/docs/MO/2005/306325_1_1.pdf>. Acesso em: 15 maio 2008.

PREFEITURA MUNICIPAL DE SÃO BENTO DO SUL. **Portal da Prefeitura de São Bento do Sul**. São Bento do Sul, 2006. Disponível em: <http://www.saobentodosul.com.br>. Acesso em: 25 maio 2008.

PROTOPAGE. **Click to title your page**. [S.l.], 2006. Disponível em: <http://www.protopage.com>. Acesso em: 20 maio 2008.

RODRIGUES, Cláudio. **Padrões de programação**: para fábricas de software, analistas e programadores. Rio de Janeiro: Ciência Moderna, 2006.

ROGÉRIO, Pedro. **XHTML**: primeiros passos. [S.l.], 2006. Disponível em: <http://www.pinceladasdaweb.com.br/blog/2006/05/28/xhtml-primeiros-passos>. Acesso em: 12 maio 2008.

SOARES, Walace. **PHP 5**: conceitos, programação e integração com banco de dados. Tatuapé: Érica, 2004.

SOUSA, Marcos C. **Unindo JavaServer Faces a Ajax.** [S.l.], 2006. Disponível em: <http://www.devmedia.com.br/articles/viewcomp.asp?comp=3199>. Acesso em: 18 maio 2008.

TEAGUE, Jason C. **DHTML and CSS**: for world wide web. Berkeley: Peachpit Press, 2001.