MARCIO JUNIOR VIEIRA

XUL – INTERFACES DE USUÁRIOS PORTÁTEIS COM TECNOLOGIA XML.

CURITIBA – PR

2005

MARCIO JUNIOR VIEIRA

XUL – INTERFACES DE USUÁRIOS PORTÁTEIS COM TECNOLOGIA XML.

Monografia apresentado para obtenção do título de Especialização em Informática, no Curso de Pós-Graduação em Software Livre, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Luciano Silva.

CURITIBA – PR

2005

Parecer de Aprovação
Monografia de Especialização em Informática
Ênfase em Software Livre
Programa de Pós-Graduação em Informática/UFPR

Declaramos que o aluno **MARCIO JUNIOR VIEIRA** entregou a versão final da sua Monografia de Especialização em Informática da Universidade Federal do Paraná, com Ênfase em Software Livre, intitulada ***XUL – Interfaces de usuários portáteis com tecnologia XML***

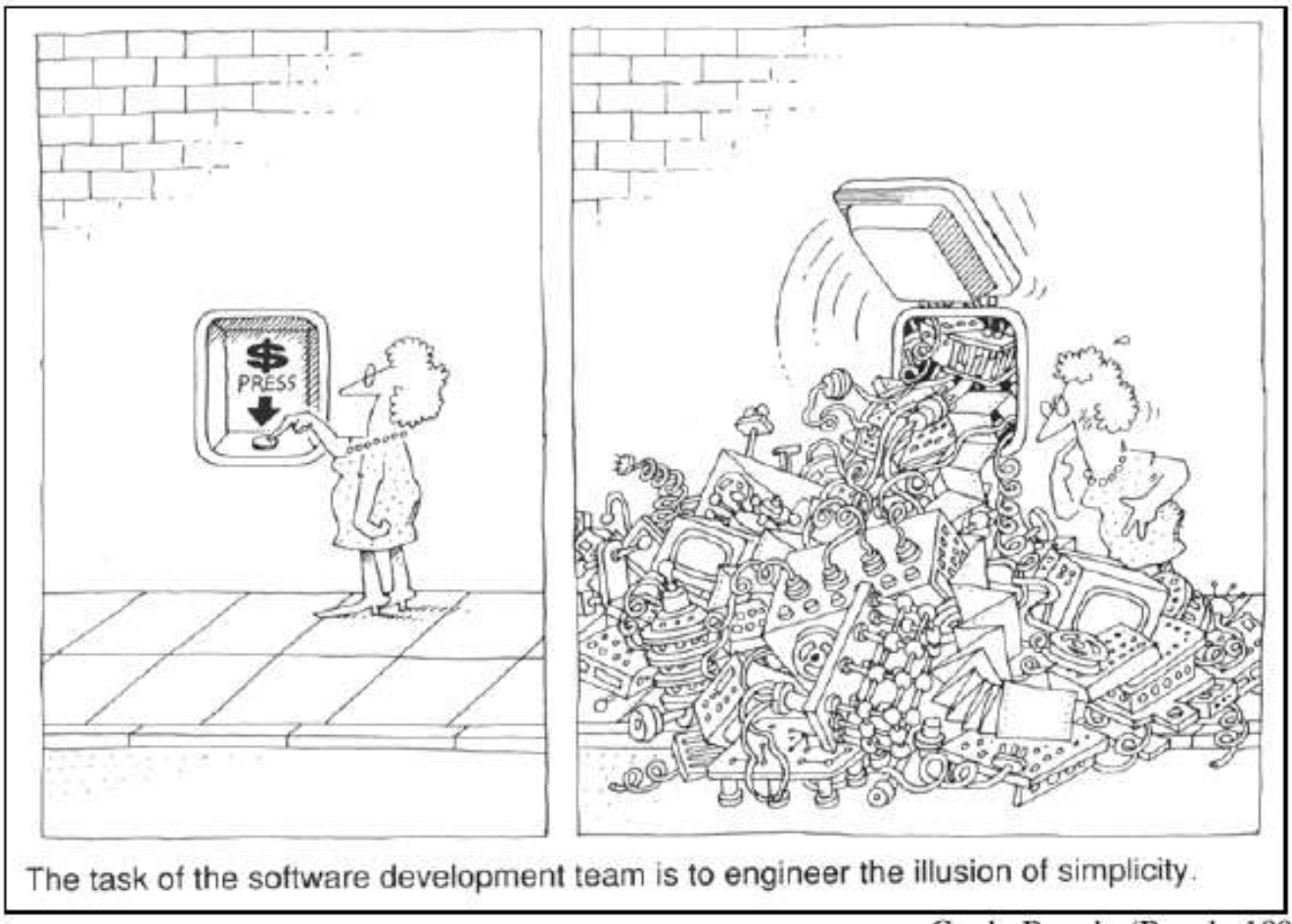
Curitiba, 18 de agosto de 2005

LUCIANO SILVA
Professor Adjunto
Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Informática
Caixa Postal 19081
CEP 81531-990 - Curitiba-PR

OLGA REGINA PEREIRA BELLON
Professor Adjunto
Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Informática
Caixa Postal 19081
CEP 81531-990 - Curitiba-PR

Dedico este trabalho a meus pais, noiva e familiares que sempre me apoiaram e tiveram ao meu lado em todos os momentos, sem também esquecer de um agradecimento a toda comunidade de software livre e open source mundial, que sem eles este trabalho nunca poderia ter sido concluído, nem se quer a menos iniciado.

Mensagem



Grady Booch (Booch, 1992)

A tarefa da equipe de desenvolvimento de software é projetar a ilusão de simplicidade.

Agradecimentos

A toda e qualquer pessoa que neste momento esta lutando pela independência tecnológica e transferência do conhecimento através do Software Livre e Open Source.

A todos colegas e professores que incentivaram e deram apoio ao desenvolvimento deste projeto.

Sumário

1. Capítulo 1 - INTRODUÇÃO....1
1.1. Apresentação....1
1.2. Objetivo geral e específicos....1
1.3. Impactos e Abrangência....2
1.4. Metodologia....3
1.5. Estrutura....3
2. Capítulo 2 - XUL....4
2.1 Conceito....4
2.2 XML....5
2.3 Interface gráficas tradicionais de usuários (GUI – Graphical User Interface)....6
2.4 O Projeto Mozilla....6
2.5 Mozilla Foundation....8
2.6 A Comunidade XUL/Mozilla no Mundo....9
2.7 Quem Usa a Tecnologia XUL ?....9
2.8 Origem do Nome XUL....10
2.9 CSS - (Cascading Style Sheets)....10
3. Capítulo 3 - ARQUITETURA XUL....10
3.1 Interfaces gráficas de usuários com XUL....10
3.2 A família de aplicativos Mozilla....11
3.3 A Organização de pacotes no Mozilla (Chrome)....11
3.4 Instalação de novos pacotes....13
3.5 Observações técnicas sobre a tecnologia XUL e XML....14
3.6 XUL Parsing....14
3.7 Estrutura de um arquivo XUL....15
4. Capítulo 4 - PORTABILIDADE COM XUL....15
4.1. Desenvolvendo portais WEB com agentes Inteligentes....15
4.2. Desenvolvendo uma barra de browser personalizada....16
4.3. Customização de Portais WEB....16
4.4 Tecnologias Suportadas....16
5. Capítulo 5 - DESENVOLVENDO INTERFACES GRÁFICAS COM XUL....18
5.1 Fontes de Dados e Modelos com RDF (Resource Description Framework)....18
5.2 eXtensible Building Language (XBL)....18
5.3 XRE (XUL Runtime Environment)....20
5.4 MathML....21
5.5 Pacotes Jar....21
5.6 XPCOM....22
5.8 XPToolkit....26
5.9 XPFE....26
5.10 XUL com JavaScript....27
5.11 XUL Com CSS....28

5.12 XUL com C/C++....29
5.13 XUL com PHP....29
5.14 XUL com Python. 29
5.15 XUL Versus Avalon XAML....30
5.16 Portabilidade na Internacionalização e Localização....30
6. Capítulo 6 - FERRAMENTAS PARA DESENVOLVIMENTO EM XUL....31
6.1 Editor de texto....31
6.2 JXUL....31
6.3 XULMaker....32
6.4 DOM Inspector....34
6.5 Venkman JavaScript Debugger....34
6.6 Cview - XPCOM Component Viewer....35
6.7 Mozilla Translator....37
7. Capítulo 7 - EXEMPLOS DE APLICAÇÕES XUL....38
7.1 Barra de Busca Amazon.com (MAB)....38
7.2 Página de pesquisa da Google.com....39
7.3 Barra personalizada para o Orkut....39
7.4 Projeto Imperador (webapp)....40
8. Capítulo 8 - ELEMENTOS DE INTERFACES GRÁFICAS DE USUÁRIOS EM XUL....41
8.1 Window. 42
8.3 Label....44
8.4 Radio....46
8.5 Radiogroup....49
8.6 Checkbox....51
8.7 Script....54
8.8 Splitter....55
8.9 Textbox....57
8.10 Treecol....59
8.12 Menu....66
8.13 Menubar....69
8.14 Menuitem....70
8.15 Progressmeter....74
9. Capítulo 9 - RESULTADOS OBTIDOS....76
9.1 XULWareHouse....76
9.2 Divulgação da Tecnologia....77
10. Capítulo 10 - CONCLUSÃO....77
11. REFERÊNCIAS....78
11.1 Livros....78
11.2 Congressos....78
11.3 Internet....78
12. GLOSSÁRIO....81
12.1 Termos....81

12.2 Abreviaturas.....82

1. Capítulo 1 - INTRODUÇÃO.

1.1. Apresentação.

Em um momento onde a Tecnologia da Informação cria uma espécie de "nova revolução industrial" no mundo, baseada em conhecimento, domínio tecnológico e domínio humano sobre os recursos naturais, e estes são utilizadas como ferramentas para exploração econômicas sem a transferência de conhecimento, e de extrema importância disponibilizar a sociedade opções onde ela tenham domínio sobre as tecnologias da informação, onde não somente sejam usuárias das tecnologias e e sim abrir um caminho para que a sociedade conheça, tenha aprendizado, ensine, aplique e também tenha condições de criar novas tecnologias, novas metodologias, softwares, etc.

As filosofias do Software Livre e do Software Aberto tem ideologias que auxiliam a independência tecnológica, onde todo o conhecimento agregado sobre as tecnologias é de poder de todos na sociedade sem nenhuma exclusão social.

Atualmente existem diversas tecnologias para criação de interfaces gráficas para usuários e o XUL é uma uma tecnologia habilitada para esta função, com a portabilidade do XML, com a força de cooperação da comunidade do software livre e ainda muito pouco disseminada entre a comunidade de desenvolvedores de software.

O XUL é tecnicamente viável, economicamente sustentáveis, socialmente justo, e um grão de conhecimento numa praia de independência tecnológica chamada Software Livre!

1.2. Objetivo geral e específicos.

Esta pesquisa tem como objetivo geral demonstrar a portabilidade na criação de interfaces gráficas de usuários (GUI) utilizando o tecnologia XUL, que é uma linguagem baseada na tecnologia XML.

Os Objetivos específicos são:

1) Apresentar o XUL e incentivar desenvolvedores de softwares, básicos ou não, a utilizarem a tecnologia como mais uma alternativa na criação de GUIs.

2) Permitindo que suas aplicações sejam portáteis para a maior quantidade de plataformas e usuários possíveis sem se preocupar muito com as características de cada arquitetura, gerando um desenvolvimento rápido, com padrões tecnológicos como os definidos pela W3C.
3) Reaproveitamento de conhecimento tecnológico já adquirido, das tecnologias que são bases para o XUL como XML, HTML, JavaScript entre outras particularidades de desenvolvimento web, assim como o reaproveitamento de componentes e regras de negocio já criados nestas tecnologias.
4) Transferência do conhecimento adquirido as vantagens e particularidades da tecnologia XUL para a criação de interfaces para WEB com funcionalidades semelhantes as aplicações chamadas de "Client Server" ou aplicativos Desktop, fazendo com que os desenvolvedores tenham opção de criar ambiente onde os usuários possam desfrutar de uma interface interativa, simples e idêntica as aplicações que ele está acostumado a trabalhar em seu desktop.
5) Geração de demandas para novas atividades econômicas como serviços de consultoria e criação de produtos baseados na tecnologia XUL.

1.3. Impactos e Abrangência.

Aqui são citamos algumas que este trabalho pode proporcionar em atividades econômicas, impactos econômicos, sociais e tecnológicos.

1) Consultoria em sistemas de informática,
2) Desenvolvimento de programas de informática entre outras atividades de informática.
3) Redução de evasão de divisas com a redução de aquisição de licenças de softwares estrangeiros.
4) Redução de custo para o desenvolvimento de software no país.
5) Valorização e transferência dos conhecimentos e objetos tecnológicos desenvolvidos.
6) Estímulo ao uso de Software Livre como mecanismo de negócios de transferência tecnológica entre a academia e o setor de desenvolvimento de software.
7) Criação de novas empresas e em conseqüência a abertura de novos ambientes de trabalho.
8) Competitividade para pequenas e médias empresas de desenvolvimento de software.

9) Possibilidade de criação de Centros Tecnológicos adequados e preparados para ensino.
10) Sustentação da competitividade econômica, com conseqüências diretas sobre a qualidade de vida com o uso de Software Livre.
11) Independência tecnologia e desenvolvimento de projetos inovadores em Software Livre.
12) Desenvolvimento de tecnologias locais.

1.4. Metodologia.

Nesta seção e descrita a metodologia adotada para o desenvolvimento deste trabalho.

1) Portabilidade para usuários é o foco deste trabalho, e ao longo de todo o desenvolvimento foram abordados assuntos que justificam a usabilidade de tecnologia para criação de ambientes para usuários portáteis.
2) Estudo preliminar da tecnologia foi realizado com a criação de elementos XUL com a intenção de testar a tecnologia.
3) Motivação para o desenvolvimento da tecnologia XUL, com um histórico da criação do XUL e sua usabilidade no desenvolvimento de projetos foi levado em consideração.
4) Estudo aprofundado da tecnologia, descrevendo a operação de XUL juntamente com outras tecnologias já conhecidas e utilizadas, e algumas ainda pouco exploradas mais necessárias para o desenvolvimento de Interfaces XUL.
5) Mapeamento de elementos, foram feitos para dar uma base dos principais elementos, podendo ser usado como referência para novos desenvolvedores XUL.

1.5. Estrutura.

Este trabalho está dividido em 10 capítulos, mais a bibliografia e um glossário. O primeiro foi uma introdução e este trabalho e quais seus objetivos na apresentação da tecnologia XUL.

O capítulo 2 apresenta o conceito de XUL e das tecnologias bases para a sua criação, assim como foi desenvolvido e o motivo que levou a sua criação.

O capítulo 3 demonstra como é a estrutura utilizada atualmente pelo XUL e a relação com os

browsers Mozilla, que suportam a tecnologia.

O capítulo 4 descreve a portabilidade e as possibilidades existentes com a tecnologia, assim como quais outras tecnologias o XUL tem suporte.

O capítulo 5 apresenta as linguagens de programações que podem ser utilizadas para trabalhar juntamente com XUL, as demais tecnologias e suas funcionalidades interagindo com XUL e algumas tecnologias futuras que pretendem libertar o XUL do browser.

O capítulo 6 apresenta ferramentas que podem auxiliar e agilizar o processo de desenvolvimento com de aplicações com a tecnologia XUL.

O capítulo 7 traz algumas interfaces de aplicações desenvolvidas com XUL para exemplificar o que pode ser criado com a tecnologia.

O capítulo 8 apresenta os principais elementos para construção de interfaces gráficas com suas características, propriedades e métodos.

O capítulo 9 apresenta alguns resultados que foram obtidos no desenvolvimento destra trabalho e no que eles podem ajudar a novos usuários de tecnologia XUL.

O capítulo 10 tem uma conclusão geral do pesquisador sobre a realização do trabalho e as experiência com XUL.

A bibliografia traz uma lista as referências bibliográficas , como livros, web sites e congressos.

2. Capítulo 2 - XUL.

2.1 Conceito.

XUL é a sigla para XML User Interface Language, que basicamente é uma linguagem multi-plataforma para desenvolver interfaces gráficas de usuários (GUI – Graphical User Interface) em aplicações, inicialmente foi criado para tornar o desenvolvimento do navegador Mozilla mais fácil, rápido e portável. Utiliza da tecnologia XML como sua base, e como o XML tem as vantagens de outras linguagens de marcação como XHTML, MathML, SVG,etc.

É uma tecnologia multi-plataforma com a portabilidade como o seu principal recurso e função, com todas as vantagens e características da linguagem de marcação XML.

Atualmente apenas os browsers da família Mozilla e Netscape e outros baseados neles, tem suporte a tecnologia XUL.

O maior benefício que o XUL traz é a definição simples e portátil de widgets comuns, o que reduz a dificuldade do desenvolvimento de software numa maneira análoga à que foi oferecida pelas linguagens de programação de quarta geração. Sua utilização em ambientes web ou desktop permite que sejam criadas interfaces gráficas com elementos e funcionalidades muitos mais ricas que as vistas atualmente, principalmente quando falamos de aplicativos para web.

2.2 XML.

XML é a sigla para eXtensible Markup Language, que é uma linguagem que faz parte de um subconjunto de uma outra linguagem chamada SGML (Stantard generalized Markup Language). A XML foi formulada em 1996, mas tem origens das linguagens de marcação generalizadas como a GML (Generalized Markup Language) da década de 60, que muitos anos após sua criação foi adotada como uma linguagem de marcação oficial da ISO (Organização Internacional para Padronização) e passou a ser chamada de SGML.

A complexidade de implementação da SGML foi um fato que incentivou a criação do XML, que além de ser disponibilizada com muita facilidade, tinha recursos da SGML e a portabilidade da HTML(Hipper Text Languagem) que anteriormente, e ainda continua sendo a mais tradicional das linguagem de marcação para WEB.

Atualmente a tecnologia XML é utilizado em inúmeros softwares , ainda mais quando falamos dos padrões de armazenagem de seus dados, dificilmente se encontram softwares que armazenam informações que não disponibilizem, de alguma maneira a geração de seus dados em um formato baseado em XML, muitos aplicativos salvam seus dados em arquivos com extensões próprias porém o conteúdo do arquivo esta no formato padrão do XML. Como as integrações entre produtos são uma realidade inevitável e cada dia mais interessantes e exigidas, o XML se tornou a melhor maneira de diferentes produtos se comunicarem. A sua simplicidade de geração, portabilidade de comunicação e rapidez de aprendizado sem dúvidas foram as principais incentivadoras de sua utilização por um conjunto tão grande diferentes aplicações.

2.3 Interface gráficas tradicionais de usuários (GUI – Graphical User Interface).

Atualmente existem diversas tecnologias para criar intefaces gráficas de usuários (GUI) , as mais conhecidas e tradicionais são : UNIX curses library, X11/Motif, Microsoft Foundation Classes, e Java AWT/Swing.

2.4 O Projeto Mozilla.

O Projeto Mozilla teve seu inicio no ano de 1998, depois que a desenvolvedora de browser Netscape compartilhou o código fonte do seu produto Netscape Communicator 4.x como open source. A partir daí uma comunidade se formou ao redor do desenvolvimento do browser Mozilla com incentivo da própria Netscape, fazendo com que o Mozilla se transforma-se em uma espécie de framework de desenvolvimento integrado ao browser. Junto com o Mozilla também eram disponibilizados o Mozilla Mail e o Mozilla Composer, formando o Mozilla Suite. Atualmente o projeto esta focado na evolução da sua nova versão de browser o Mozilla Firefox que teve seu desenvolvimento separado do cliente de e-mail que agora se tornou um outro produto independente chamado Thunderbird. Outros produtos principais do projeto Mozilla são o Bugzilla, para gerenciamento de bugs no desenvolvimento de software, o Camino que é um browser personalizado para a plataforma Mac OS X e o Calendar Project que é um gerenciador de calendário, mas também podem ser encontradas no repositório do Mozdev.org diversas outras iniciativas de softwares baseados nas tecnologias do Mozilla. Depois de um tempo foi fundada a Mozilla Foundation para organizar o projeto e dar sustentação legal e financeira. No Final do ano de 2004 no lançamento do browser Mozilla Firefox 1.0 algo de surpreendente começava a acontecer, apenas nos primeiros dias de liberação foram feitos mais de 2 milhões de downloads do browser, e isso era apenas o inicio, em alguns meses depois o site Mozilla.org anunciava chamava orgulhosamente para que mais pessoas se juntassem aos demais 64 milhões de pessoas que tinham feito o download do Firefox, juntamente com reportagens importantes como a da revista Fourbes que comentava o fenômeno, os dizeres da chamada principal no site eram r “Firefox rediscover the web” juntamente com estes anúncios a NetApplications , empresa que atua no monitoramento de trafego de internet apontava crescimentos do

uso do firefox próximos a 1% ao mês , que chegou em maio de 2005 com uma fatia de mais de 8% do mercado dos browser que somados aos browser Netscape e Mozilla ultrapassavam 10% do mercado, um numero surpreendente num espaço de tempo tão pequeno. Não era e ainda não é incomum encontrar diversos artigos na Internet onde grandes empresas e órgãos recomendado o uso do browser Firefox e comentando o motivo pelo qual o mesmo é o browser mais seguro da atualidade. O mercado dos browsers começa a se comportar como um "pêndulo" que teve seu inicio nos anos 1997 e 1998 num período chamado de "guerra dos browsers" onde a Netscape era a líder de mercado até o momento em que a Microsoft começou a liberar gratuitamente o seu produto, o Internet Explorer, que acabou tomando e acabando com o mercado de venda de browsers. Algum tempo depois estimulado pela filosofia de um livro chamado "The Cathedral and the Bazaar" (A Catedral e o Bazar) e escrito por um dos mais famosos filósofos e Hackers de software abertos , Eric Raymond , diretores da Netscape decidiram liberar todos os fontes do seu browser formando o projeto Mozilla e futuramente a Mozilla Foundation, que pro fim deu uma nova trajetória do pêndulo que caminha para um aumento da utilização dos browser da Família Mozilla. A possibilidade de alterar o fonte e redistribui-lo livremente também criou novas extensões do Firefox, que mapeadas atualmente chegam a ser mais de 120 personalizações do browser para funcionalidades especificas, como por exemplo podemos citar algumas com funcionalidades avançadas de pesquisa no Google, facilidades para traduções de textos, visualização de vídeos, controle de áudios , funcionalidades para zoom, etc. Um dos principais desenvolvedores do browser Firefox, o garoto Blake Ross, que no momento do lançamento do browser tinha apenas 17 anos, é outro fenômeno que ajudou nesta revolução. Sua atividades são citadas pela Mozilla foundation com as principais na criação do Firefox, Ross inicio seu trabalho na Mozilla Foundation com 14 anos e saiu pouco antes do lançamento oficial do Firefox. Outro fato importante também foi a maneira cooperativa e colaborativa que a Mozilla Foundation arrecadou com de 100 mil fãs do Firefox , US$ 250 mil para comprar um anuncio de página dupla no famoso jornal "The New York Times" anunciando o lançamento do seu browser. O anuncio trazia o nome dos 100 mil colaboradores e uma introdução sobre o Firefox, Veja a anúncio na **Figura 2.4a**.

E o que tudo isso tem a ver com XUL ? Simples! com isso a tecnologia XUL pode ser mais executada por milhões de browsers espalhados no mundo, novas demanda aparecem todos os

momentos , que estiver preparado para atende-las com aplicações escritas nesta tecnologia pode ter um grande diferencial e em outros casos isso pode ser essencial pela portabilidade da tecnologia XUL.

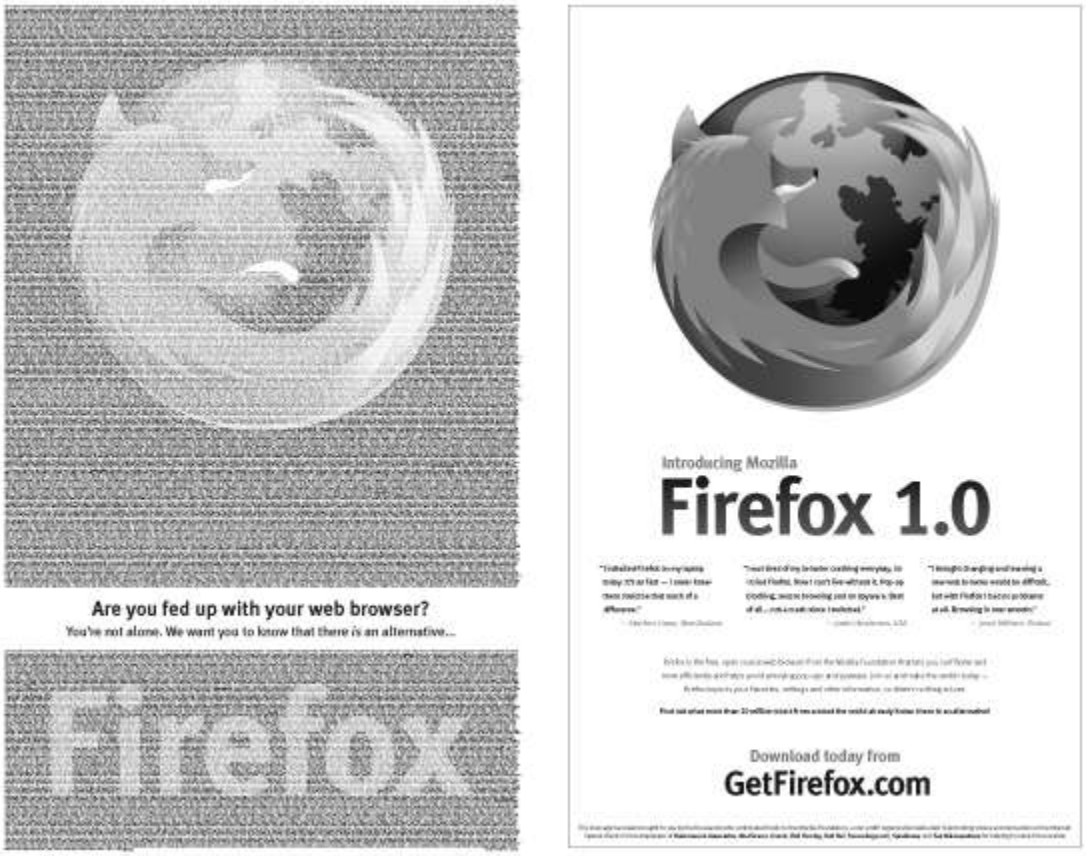


Figura 2.4a – Anúncio no *"The New York Times"* sobre Firefox 1.0 patrocinado por 100 mil usuários.

2.5 Mozilla Foundation.

A Mozilla Foundation foi estabelecida em julho de 2003, com sustentação inicial da América Netscape, a fundação do Mozilla existe para fornecer a sustentação organizacional, legal, e financeira para o projeto do software Open Source Mozilla. A fundação foi incorporada não para fins lucrativos mais para assegurar-se de que o projeto de Mozilla continuasse a existir além da participação de voluntários individuais, e permitir contribuições da propriedade intelectual, e a fornecer um veículo para limitar a exposição legal ao participar em projetos do software open source.

A Missão do projeto de Mozilla é preservar a escolha e a inovação da Internet.

2.6 A Comunidade XUL/Mozilla no Mundo.

Atualmente existem diversas comunidade trabalhando pela evolução e disseminação da tecnologia XUL no Brasil e no Mundo. Podemos citar em nosso país um portal com muitas referências , exemplos e traduções de tutoriais desta tecnologia que é o XUL Brasil, , o sítio utiliza de uma ferramenta de WIKI para disponibilizar todo seu conteúdo e facilitar a colaboração de demais participantes. Outra opção também e o Rau-tu do Mozilla/Brasil mantido no centro de computação da Unicamp , porém ainda com muito poucas perguntas sobre XUL.

O portal mais conhecido e que tem as mais ricas informações sobre a tecnologia XUL é o XULPlanet (USA), o mesmo contém tutoriais , guias de referências, artigos de altíssima qualidade e mais informações até mesmo que o sítio oficial do projeto XUL , que está no Mozilla.org , o sítio oficial tem um manual de referência do programador e algumas especificações do XUL 1.0 e links para sítios relacionados .Outros países que se destacam na internet com documentação sobre XUL são a França e o Japão.

Também não podemos deixar de citar o Mozdev.org que disponibiliza diversos projetos relacionados a tecnologia XUL.

2.7 Quem Usa a Tecnologia XUL ?.

O maior usuário da tecnologia XUL não há duvidas de que é o seu próprio desenvolvedor a Mozilla Foundation principalmente por sua necessidade de criar as aplicações portaveís, porém existem outras corporações já utilizando a tecnologia XUL em suas aplicações como é o caso da gigante Google.com que tem uma página de busca escrita em XUL. No Brasil temos uma empresa que esta criando uma aplicação de ERP nesta tecnologia o aplicativo está na web para demostração basta ter um brwser da família mozilla e ele pode ser analisado via web , mais a grande maioria dos projetos estão nas comunidades de software livre e tem como principal atividade a criação de barra de tarefas do browser para funcionalidade especificas. Uma Aplicação que se destaca e a MAB – Mozilla Amazon

Browser que tem uma interface para gerenciamento de busca nos catálogos da Amazon.com.

2.8 Origem do Nome XUL.

O nome XUL é uma referência ao filme Caça-Fantasmas, em que um fantasma chamado *Zuul* possui uma personagem chamada Dana (representada por Sigourney Weaver) e declara "Não existe Dana, apenas Zuul". Como XUL é usado para definir um interface em vez de um documento, os seus desenvolvedores adoptaram o chavão "Não existem dados, apenas XUL". Isto é referido pelo URI **namespace XML** no inicio de qualquer documento XUL, quando visto por uma aplicação capacitada para XUL, "Keymaster" e "gatekeeper" são também referências. Não é apenas a única referência ao filme nas aplicações do projeto Mozilla, por exemplo, o componente depurador de JavaScript é chamado de Venkman, uma das suas personagens principais.

2.9 CSS - (Cascading Style Sheets).

Linguagem de formatação proposta pelo W3 Consortium, que é o consórcio que define as regras e especificações para a Internet . Visa remover a formatação dos documentos HTML e XML, separando informação e formatação. Mais precisamente, em português a sigla significa "Folhas de Estilo em Cascata".

3. Capítulo 3 - ARQUITETURA XUL.

3.1 Interfaces gráficas de usuários com XUL.

O XUL tem a capacidade de criação de interfaces modernas e sofisticadas, não ficando atrás de nenhuma tecnologia tradicional para a criação de GUIs, alguns dos elementos possíveis que podemos citar são:

- Controles de entrada tais como caixas de texto e de checagens (textboxes e checkboxes).
- Barras de ferramentas com botões ou outros conteúdos.
- Menus em uma barra de menus ou menus flutuantes (popups).

- Diálogos organizados em abas.
- Árvores para informação hierárquica ou tabular.
- Atalhos de teclado.

Para que estas GUIS sejam apresentadas , necessáriamente deve existir um parsing para a interpretação das tags XUL e a disponibilização de sua aparência, que é encontrado nos browsers de Internet da Mozilla Foundation, como o Firefox e o Mozilla. As interfaces escritas em XUL são muito semelhantes a interfaces de Desktop, porém podem rodar na web, o que terna a tecnologia altamente viável para construção de Interfaces. A tecnologia se integra a diversas outras linguagem de programação para tratamento da parte de negocio ou lógica , sendo a linguagem script JavaScript a mais utilizada entre os desenvolvedores de interfaces XUL.

3.2 A família de aplicativos Mozilla.

O desenvolvimento da família de aplicativos da Mozilla Foundation, onde estão compostos os browsers Mozilla e o Firefox, foi o principal propulsor da criação da tecnologia XUL. Uma necessidade no desenvolvimento era que os aplicativos da família fossem portáteis, e com uma facilidade e agilidade na criação das GUIs, assim foi utilizando as tags XUL juntamente com um parsing para interpretação das tags, criando as GUIs das ferramentas de e-mail Mozilla Mail, do aplicativo para criação de páginas web Mozilla Composer, entre outros componentes da família Mozilla.

3.3 A Organização de pacotes no Mozilla (Chrome).

Todos os pacotes XUL distribuídos no Mozilla estão no diretório chrome, que se refere a elementos de interface gráfica de usuários da aplicação.

A partir do diretório chrome podemos observar todos os pacotes distribuídos no Mozilla, cada pacote fica em um subdiretório e tem três sub-divisões básicas: conteúdo, Skin e Localidade. Porém nada impede que a estrutura seja diferente ou mais complexa. Vamos analisar cada um destes subconjuntos de pacotes para entender melhor para que servem:

Conteúdo (content)- Janelas e scripts. As declarações das janelas e os elementos de interface de usuário são armazenados neste diretório em arquivos XUL, que têm extensão xul. Um pacote de conteúdo pode ter múltiplos arquivos XUL, mas a janela principal precisa ter um nome de arquivo que tenha o mesmo nome do pacote. Os Scripts são colocados em arquivos separados, próximos aos arquivos XUL.

Skin - Folhas de estilo e imagens. Folhas de estilo descrevem detalhes sobre a aparência da janela. Elas são armazenadas separadamente de arquivos XUL para facilitar a modificação do ski da aplicação. Qualquer imagem usada é armazenada aqui também.

Localidade (Locale)- Arquivos específicos de localidade. Todos os textos que são mostrados na janela são armazenados separadamente. Desta forma, um usuário pode ter um conjunto para o seu próprio idioma.

Na **Figura 3.3a** podemos visualizar uma estrutura de diretório padrão para pacotes instalados no Mozilla.

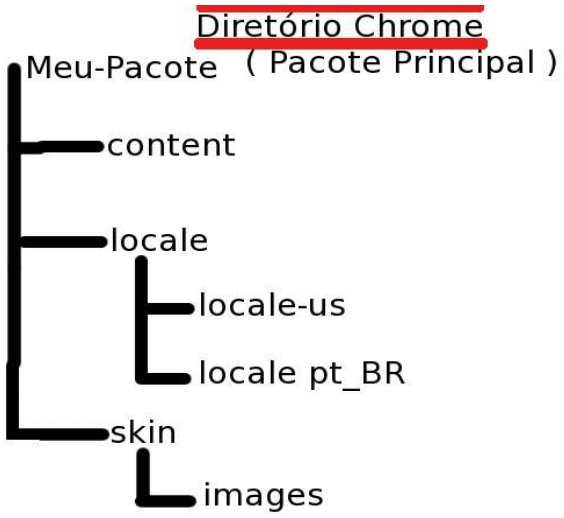


Figura 3.3a – Estrutura do diretório Chrome do Mozilla.

Esta estrutura pode estar compactada em um arquivo JAR , o Mozilla trabalha transparentemente com arquivos deste formato e para acessar qualquer pacote instalado em seu diretório chrome pode ser digitado no browser o link <chrome://nomedopacote/content/>. Exemplo para abrir o pacote messenger: <chrome://messenger/content/>.

Por padronização e para informar ao Mozilla o que contém dentro do seu pacote deve ser criado um arquivo de manifesto, este arquivo vai descrever alguns índices que interagem com o diretório chrome, deve ser um arquivo RDF (em alguns casos mais que um) com o nome ***content.rdf*** no diretório ***content.*** Podem ser especificados arquivos RDFs para os demais diretório caso o pacote tenha um grau alto de complexidade. veja na **listagem 3.3a** um exemplo de arquivo RDF .

Listagem 3.3a - Arquivo RDF (content.rdf) de Manifesto do pacote Messenger.

```
<?xml version="1.0"?>
<RDF:RDF xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:chrome="http://www.mozilla.org/rdf/chrome#">

  <!-- list all the packages being supplied by this jar -->
  <RDF:Seq about="urn:mozilla:package:root">
    <RDF:li resource="urn:mozilla:package:messenger-mdn"/>
  </RDF:Seq>

  <!-- package information -->
  <RDF:Description about="urn:mozilla:package:messenger-mdn"
        chrome:displayName="Messenger"
        chrome:author="mozilla.org"
        chrome:name="messenger-mdn"
        chrome:localeVersion="1.7"
        chrome:skinVersion="1.5">
  </RDF:Description>
</RDF:RDF>
```

3.4 Instalação de novos pacotes.

Todos os pacotes instalados no Mozilla se encontram no diretório chrome, no arquivo chrome.rdf, o Mozilla registra um índice automaticamente de acordo com os pacotes já instalados juntamente com as opções descritas no arquivo 'installed-chrome.txt'.

Caso haja a necessidade de instalar um novo pacote basta adiciona-lo no arquivo 'installed-chrome.txt' que é um arquivo menos complexo que o chrome.rdf. O Arquivo 'installed-chrome.txt' contém uma lista de instalações, uma por linha, ele existe no diretório chrome. Veja um exemplo na **Listagem 3.4a**.

Listagem 3.4a – Arquivo Installed-chrome.txt.

```
content,install,url,resource:/chrome/findfile/content/findfile/
skin,install,url,resource:/chrome/findfile/skin/findfile/
```

- **Tipo**: Definido como 'content' para pacotes chrome, 'skin' para skins ou 'locale' para localidades.
- **Forma de Instalação**: Use o texto 'install' para ter a entrada instalada. Para skins e localidades, você pode também usar 'profile' para ter a entrada instalada no diretório de perfil do usuário. Isso significa ser instalado somente para um usuário específico.
- **Tipo de URL**: Coloque 'url' para especificar uma URL onde o novo pacote, skin ou localidade está armazenado. Se você usar 'path' ao invés de 'url', pode usar um caminho de arquivo (no formato do seu sistema operacional).
- **URL**: Define a URL ou caminho de arquivo onde se encontra o pacote. Isso pode ser o diretório que contém o arquivo contents.rdf ou o caminho ou nome do arquivo JAR. Quando se referir a um diretório, esteja certo de que há um caracter barra no final.

3.5 Observações técnicas sobre a tecnologia XUL e XML.

Existem algumas particularidades da tecnologia XUL , geralmente herdadas da própria tecnologia em que ela e baseada o XML , a seguir algumas observações importantes:

- XUL e XML são Case-sensitive.
- Valores de atributo em XML devem ser colocados entre aspas, mesmo que sejam números.
- Arquivos XUL são usualmente divididos em quatro arquivos, um deles para esboço e elementos, para declarações de estilo, para declarações de entidade (usadas para localização) e para scripts.

Adicionalmente, você pode ter arquivos extra para imagens ou para dados específicos da plataforma.

3.6 XUL Parsing.

Para a execução de interfaces XUL e necessário um parsing , que atualmente existe integrado nos browsers Firefox e Mozilla, ou outros browser que tenham como base da sua criação os browsers Mozilla. Este parsing é chamado de Expat XML, sendo um parser para XML escrito em C, que foi

integrado ao Mozilla, ele faz uma analise de sintaxe baseado no Document Object Model (DOM) construindo durante o parsing, permitindo acesso rápido ao conteúdo dos arquivos e suas tags, é feita primeiramente uma validação do modelo XML e num segundo momento analisado a sintaxe XUL , caso ocorra algum erro será demonstrado em uma janela.

3.7 Estrutura de um arquivo XUL.

Um arquivo XUL pode ter qualquer nome mais necessáriamente deve ter a extensão com .xul, verifique um exemplo na **Listagem 3.7a**. Nas duas primeiras linhas do arquivos temos a especificação de um arquivo XUL, na linha 3 é iniciado um componente window , com os atributos id,title, orient, e o atributo especial e obrigatório xmlns, que apesar de ser um link para uma url o mesmo nunca é acessado na execução do arquivo XUL.

Listagem 3.7a - Arquivo XUL Simples.

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
3 <window
4    id="arquivo-exemplo"
5    title="Exemplo de Janela – File XUL"
6    orient="horizontal"
7    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
8 ...
9 </window>
```

4. Capítulo 4 - PORTABILIDADE COM XUL.

4.1. Desenvolvendo portais WEB com agentes Inteligentes.

A tecnologia XUL se aplica muito bem a portais web que podem trabalhar com uma interface muito mais rica em elementos visuais, deixando a internet mais parecida e semelhante com as aplicações desktop atuais, com isso podem ser criados uma inteligência para reconhecimento com agentes inteligentes, onde é possível identificar usuário e utilizar outros recursos deste conceito para interagir com o cliente do browser.

4.2. Desenvolvendo uma barra de browser personalizada.

Uma atividade muito explorada atualmente pelas comunidades que desenvolvem utilizando a tecnologia XUL e a criação de browsers personalizados, o Mozilla Firefox atualmente consta com mais de 100 extensões do browser padrão, onde a principal diferença esta na de barras para auxilio de navegação na Internet. Existe inúmeras possibilidades em que a criação de um browser personalizado pode melhorar alguma atividade na internet, podem ser criados barras do browser com as características operacionais de uma empresa agilizando e qualificando o seu trabalho, podem também ser criados browser com restrições especificas como na disponibilização de quiosques publico de acesso a Internet, browsers simples para iniciante, interface mais atraente e colorida para crianças, interfaces personalizadas para facilitar a aprendizado de idosos, browser especializados em pesquisa, browsers customizado para um interagir com paginas web e sistemas específicos na Internet,etc.

Para a navegação na Internet básicamente existem duas operações, uma URL e um clique, porém nas condições atuais onde os usuários ficam, reféns de popups, vírus, spywares, links indesejáveis e outras artimanhas de contaminação do seu browser, muitas vezes a internet pode estar querendo "dominar o seu browser" com estas práticas , o que é inaceitável, A possibilidade de alteração dos fontes escritos, sejam eles escritos em XUL ou em outras tecnologias, e a personalização dos browsers Mozilla permite que seus usuários tenham a Internet de volta.

4.3. Customização de Portais WEB.

Como a tecnologia XUL pode ser executada aplicativos no cliente, isso permite que o mesmo possa estar criando, caso a aplicação permita isso, áreas do Portal como "Meu Portal", um portal configurado pelo próprio usuário, com apenas os interesses do usuário, atalhos , as noticias que deseja visualizar o campo para de busca de sua preferência, com a tecnologia XUL podemos dominar e devolver a Internet ao usuário através de aplicações em seu browser ele pode gerenciar da sua melhor maneira o seu interesse. Não ficando restrito as interfaces web tradicionais.

4.4 Tecnologias Suportadas.

Atualmente já existem diversas tecnologias que podem fazer interoperabilidade com as

interfaces XUL na listagem a seguir todas as tecnologias que podem ser utilizadas integradas ao XUL.

- HTML 4.0 e XHTML
- CSS2 a alguns CSS3
- DOM2 e alguns DOM3
- XML
- MathML
- RDF
- XSLT
- XPath
- Simple XLinks
- XPointer
- XML Base
- FIXPtr
- XML-RPC
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- ECMAScript/Javascript
- XBL (eXtensible Bindings Language)
- GIF, JPG, PNG, BMP, ICO images
- HTTP 1.1

A maiorias destas tecnologias é citada e comentada no decorrer deste trabalho mostrando suas aplicações juntamente com a tecnologia XUL, a união destas tecnologias com XUL cria uma ambiente de desenvolvimento poderoso e altamente portável, Sendo assim o Mozilla pode ser considerado mais uma plataforma para desenvolvimento de aplicações web.

5. Capítulo 5 - DESENVOLVENDO INTERFACES GRÁFICAS COM XUL.

5.1 Fontes de Dados e Modelos com RDF (Resource Description Framework).

Arquivos RDF são modelos ou fontes de dados, também conhecidos como metadata , esta tecnologia que foi endossada e também é recomendada pela W3C em fevereiro de 1999 , tem como objetivos principal criar um modelo simples de dados, com uma semântica formal e usar o vocabulário URI-based, usar uma sintaxe XML-based e suportar o uso de XML. Os arquivos RDF tem três componentes básicos,recurso, propriedade, indicação, o que torna a linguagem altamente escalável.

Recurso: Qualquer coisa que pode conter um URI, incluindo as páginas da WEB, assim como elementos de um documento XML.

Propriedade: Um recurso que tenha um determinado nome e possa ser utilizado como uma propriedade.

Indicação: consiste na combinação de um recurso, de uma propriedade, e de um valor.

Na **Listagem 5.1a** descrevemos a estrutura de um arquivo RDF simples, podemos analisar que na linha um temos a definição para arquivos baseados em XML, na linha 2 é iniciado a sintaxe para o RDF , sendo que na linha 3 é informado um link para o reconhecimento de um arquivo RDF , sendo que este link é apenas para identificação e nunca realmente será acessado ao carregar um arquivo RDF.

Listagem 5.1a - (simple.*rdf*) Arquivo RDF Simples.

```
1 <?xml version="1.0"?>
2 <RDF:RDF
3   xmlns:RDF="http://www.w3.org//1999/02/22-rdf-syntax-ns#">
4  ...
5 </RDF:RDF>
```

5.2 eXtensible Building Language (XBL).

A eXtensible Bindings Language é usada para criar elementos ou componentes de interface personalizados e ligar eles a outros elementos XUL, XML ou HTML.

Com esta linguagem podemos customizar elementos, padronizando estilos de cada componente, e juntarmos elementos, sejam eles elementos XUL, XML ou HTML ou ambos.

Sua utilização é muito prática fácil e flexível, é necessário apenas uma sintaxe especifica para

criação de componentes com o XBL. Geralmente os componentes XML são alocados em arquivos com extensão .xml e chamados pelos arquivos .xul ou .css através do comando **-moz-binding : url (*"link para arquivo xml" # "componente XML"*)** .

Vamos conhecer em exemplo para criação de componentes XBL :

Primeiramente vamos criar o arquivo XML que vai descrever nosso componente XBL, vamos chamado de **botaocadastro.xml** , e vamos criar um único componente que tenha 3 botões , o botão gravar, cancelar e excluir, imaginando que queremos padronizar as telas de cadastros de um sistema qualquer. Veja **Listagem 5.2a. N**a linha 1 temos a referência utilizada para especificar um arquivo XML, na linha 2 temos a especificação de que este arquivo contém componentes XBL, na linha 3 temos a nossa conhecida referência de arquivos XUL, na linha 4 é iniciado a construção do novo componente com a tab *bindings,* como o nome (id) *BotaoCadastro,* logo a pós na linha 5 a tag *content* e responsável pelo agrupamento dos componentes XML,XUL ou HTML, neste exemplo incluiremos os 3 botões XUL. A chamada dos componentes tem uma sintaxe um pouco diferente da sintaxe XUL, na linha 6 podemos ver que primeiramente e especificado a tecnologia utilizada , neste caso XUL , em seguida o elemento e logo após os atributos e propriedades do elemento. Com este arquivo construimos um novo componente que pode ser utilizado em nossas intefaces gráficas XUL. Dentro de um arquivo XML podem ser adicionados N componentes XML , basta criar mais tags binding no arquivo.

Listagem 5.2a - (*botaocadastro.xml*) Arquivo XBL Simples com três componentes buttons.

```
1 <?xml version="1.0"?>
2 <bindings xmlns="http://www.mozilla.org/xbl"
3
xmlns:xul="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
4   <binding id="BotaoCadastro">
5     <content>
6       <xul:button label="Gravar" />
7       <xul:button label="Cancelar" />
8       <xul:button label="Excluir" />
9     </content>
10  </binding>
11 </bindings>
```

Agora vamos analisar o arquivo chamado **cadastro.css** que terá todas as definições padrões de nossos componentes utilizados para cadastro e seus estilos, veja **Listagem 5.2b**. Na linha 1 nomeamos um componente CSS que será chamado através das interfaces XUL , e na linha 3 temos uma

chamada para o nosso componente XBL, composta por uma url , que especifica o local onde o arquivo XML se encontra, a marcação # e em seguida e o componente XBL que deseja adicionar.

Listagem 5.2b - (*cadastro.css*) Arquivo CSS Simples com chamada para um componente XBL.

```
1 box.BotaoCadastro {
2     -moz-binding: url('botaocadastro.xml#BotaoCadastro');
3 }
```

Chegamos ao último passo, a criação do nosso arquivo XUL, que receberá um componente XBL personalizado, na linha 3 temos a referência para o arquivo CSS que será utilizado, na linha 6 temos a chamada para o o CSS chamada de *BotaoCadastro,* a tab box é o primeiro nível da referência descrita no arquivo CSS e a propriedade class traz o segundo nível descrito no arquivo Css, esta chamada será responsável pela criação do componente XBL a interface.

Listagem 5.2c - (*telacadastro.xul*) Arquivo XUL utilizando componente XBL.

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
3 <?xml-stylesheet href="cadastro.css" type="text/css"?>
4 <window
5     xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
6   <box class="BotaoCadastro"/>
7 </window>
```

O XBL e muito útil para criação de componentes XUL, HTML, agilizando e padronizando tarefas, e inibindo o retrabalho para construção de novos componentes. Uma boa dica para a criação de bibliotecas e a criação de arquivos xml com componentes XUL divididos em grupos como um grupo de barras de taferas , telas de cadastros, interfaces para ajuda, telas de mensagens e tantos outros componentes que pode ser utilizado em mais de uma localidade dentro de um software.

5.3 XRE (XUL Runtime Environment).

O XRE é um projeto original de 4 colaboradores da Netscape que foi lançado na comunidade do projeto Mozilla em 2003, que tem como seu principal foco criar um ambiente para execução de programas XUL/JavaScript independente. Atualmente tudo que é produzido em XUL somente pode ser executador nos browsers da Mozilla Foundation (Mozilla e Firefox) . O XRE pretende dar independência para as aplicação XUL , sendo ele um toolkit de sistema. Atualmente a linguagem XUL

e dependente dos browser Mozilla o que pode dificultar em alguns casos na distribuição de novos aplicativos.

O Primeiro passo do projeto e executar um programa *"Hello World"* em XUL independente do browser Mozilla.

5.4 MathML.

MathML é uma linguagem de marcação para exibição de equações matemáticas, recomendada pela W3C. Foi liberada em fevereiro de 2001, sendo ela amplamente suportada pelos browsers mozilla que desde 1999 tem um projeto especifico chamado "MathML in Mozilla"
para suportar esta especificação.

Um exemplo simples de Tags MathML pode ser visto a seguir na **Listagem 5.4a** e em seguida o seu resultado após a sua interpretação:

Listagem 5.4a - Arquivo com tags simples MathML.

```
1 <math xmlns="&mathml;">
2 <mfrac>   <mi>a</mi>   <mi>b</mi> </mfrac> </math>
```

Resultado demonstrado no browser com suporte a MathML:

a

b

5.5 Pacotes Jar.

JAR significa *Java ARchive*, e é simplesmente um arquivo compactado, assim como arquivos ZIP, com uma extensão diferente apenas. Estes arquivos são usados para facilitar a distribuição de pacotes e manter a simplicidade na hora de configurar os ambientes de desenvolvimento, uma vez que é possível colocar todos seus pacotes (conjunto de classes) dentro de um único arquivo JAR.

Os Arquivos JAR devem ter a mesma estrutura do pacotes originais compactado para que sua utilização seja perfeita, a **á**rvore de um pacote padrão tem como subdiretórios o content,locale, e ski podendo ser adicionados outros dependendo da necessidade. O Mozilla confia nos arquivos jar

instalados no diretório chrome e os executa através do browser.

5.6 XPCOM.

XPCOM é um sistema de objetos utilizados pelo Mozilla, a tradução da sigla representa Modelo Objeto de Componente Multi-plataforma, este recurso e fundamental para a portabilidade de aplicações desenvolvidas em XUL , pois com o XPCOM podemos chamar códigos nativos de maneira fácil. Em muitos casos somente o uso da tecnologia JavaScript para eventos e lógicas pode ficar limitada, sendo necessário uma outra tecnologia. Quando isso acontece utilizamos o XPCOM para fazer as chamadas de aplicativos externos.

O XUL tem diversas interfaces XPCOM para comunicação com objetos nativos , por padrão todos os nomes destas interfaces são inicializadas por “nsl” , para serem facilmente reconhecidas , a maioria dos componentes XPCOM são implementados de forma nativa e quando a tecnologia JavaScript não é suficiente para executar, um exemplo de componentes XPCOM é a interface nslSound utilizada para tocar arquivos de aúdio, que pode ser chamada em uma aplicação XUL. Existem diversas intefaces já escritas e a tecnologia permite que muitas outras sejam criadas para atender as mais variadas demandas na construção de interfaces para usuário.

O processo de chamada XPCOM de um script é chamado de XPConnect , que representa uma camada que traduz objetos de script para objetos nativos.

O XPCOM é uma ferramenta fantástica para modularidade em aplicativos, Além da modularidade, ele oferece a agilidade através das plataformas. É suporta apor todas as plataforma que hospedagem um bom compilador C++, podemos citar:

Microsoft Windows (all flavors)

- Linux
- HP-UX
- AIX
- Solaris
- OpenVMS

- MacOS
- BSD

O XPCOM não é uma particularidade do Mozilla ele é utilizado por outros diversos aplicativos tais como :

- Komodo (IDE)
- Chatzilla (Cliente de IRC)
- Games (MozInvaders, Xultris, outros)
- Jabberzilla (Mensagens Instantâneas)
- NS6 (Netscape's browsers)

5.7 Interfaces XPCOM.

O Mozilla inclui muitos componentes XPCOM, a seguir são caracterizados a maioria dele com suas funcionalidades:

- **Accessibility.**

  Permitem que os dispositivos de acessibilidade recuperem a informação sobre elementos da interface original de usuário.
- **Address Book.**

  Usado para interagir com uma estruturas de dados de livros de endereço.
- **Browser Components.**

  Componentes relacionados a browsers como bookmarks, history,auto-completo,etc.
- **Clipboard and Selection.**

  Utilizada para tarefas de recuperação de texto como copiar, colar e armazenar as transações utilizadas no ação de desfazer.
- **Content and Layout.**

  Índice e disposição em geral.
- **Cookies.**

Interface para gerenciar informação armazenada em cookies do HTTP.

- **Core.**

  Utilizado para trabalhar diretamente com o Core do XPCOM.

- **DOM.**

  Interface do DOM suportadas pelo Mozilla. A maioria das interfaces é correspondente as definidas pelas recomendações de W3C DOM.

- **DOM Inspector.**

  Responsável pela examinação de um DOM que é carregado.

- **Docshell and Content Viewers.**

  Responsável por gerencial a visualização de um documento.

- **Editor.**

  Interface com o editor do Mozilla.

- **Files and Streams.**

  Interface para Arquivos e diretórios, possibilitando a leitura e escrita.

- **Graphics and Widgets.**

  Interface para gráficos e widgets especiais. A execução depende da plataforma do usuário.

- **IPC.**

  Interface de comunicação interProcess entre aplicações diferentes de Mozilla/GRE.

- **Images.**

  Para carregamento de imagens , suportando diversos formatos diferentes.

- **JavaScript.**

  Interface do Javascript e do XPConnect. O Mozilla usa o Javascript como sua linguagem scripting default, porém esta interface não é necessariamente específicas do Javascript e pode aplicar-se a outras linguagens de script.

- **JavaScript Debugger.**

  Interface de debugger de Javascript do Mozilla.

- **LDAP.**

  Execução de LDAP no Mozilla

- **Localization.**

  Interface para localização e internacionalização.

- **Mail.**

  Interface com o Mozilla Mail e para envio de mensagens.

- **Network.**

  Interface para a rede, chamada a proxies , protocolos http,ftp.

- **News.**

  Interface para leitor de noticias do Mozilla.

- **Plugins.**

  Interface para plugins e Java.

- **Preferences.**

  Interface para leitura de perfis de usuários.

- **RDF.**

  Manipulação de datasources RDF.

- **Security and Privacy.**

  Interface de segurança para certificados p3p.

- **Web Browser.**

  Interface para componentes em geral relacionados a browsers

- **Web Services.**

  Interface para Web Services , entre os recursos mais conhecidos suportados estão o SOAP , XML-RPC e WSDL, porém estas interfaces são acessadas também diretamente por JavaScript

- **Window Management.**

  Responsável pela manipulação de janelas.

- **XML.**

  Interface com características XML.

- **XML Schema.**

  Interface para leitura de arquivos schema XML.

- **XPInstall.**

  Interface para instalação de componentes no Mozilla.

- **XUL .**

  Interface para XUL , para elementos especiais tais como trees, popups e templates.

5.8 XPToolkit.

O XPToolkit é um conjunto de elementos de interface criados em XUL.

5.9 XPFE.

Ambiente criado a partir do XPToolkit, o XPFE utiliza de diversos padrões existentes na WEB como CSS, javaScript e XML (Linguagem em que o XUL é baseado) formando um estrutura de tecnologias, permitindo que sejam desenvolvidas aplicação multiplataformas com padrões da WEB. O Livro "Creating Applications with Mozilla" demonstra em um dos seus capítulo uma figura que exemplifica claramente como é formado a estrutura XPFE, **Figura 5.9a** , e uma comparação entre DHTML e o XPFE, **Figura 5.9b.**

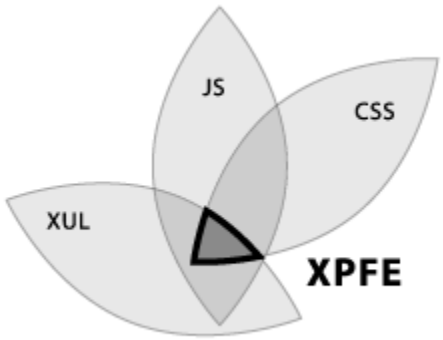


Figura 5.9.a – Estrutura do XPFE.

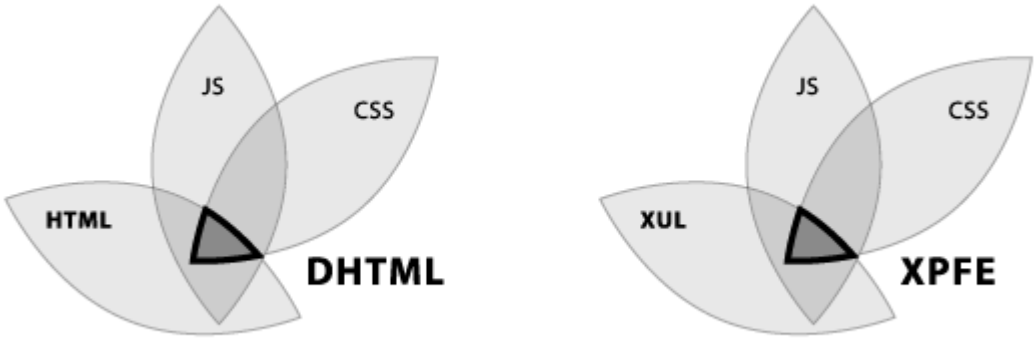


Figura 5.9b – Uma comparação entre DHTML e XPFE.

5.10 XUL com JavaScript.

A linguagem de programação mais utilizada pela maioria dos programadores que desenvolvem aplicações com XUL e o JavaScript, basta verificar os arquivos dos browser do Mozilla para constatar isso, são diversos arquivos .js, o que torna a tecnologia XUL muito agradável para programadores acostumados em desenvolvimentos para Internet.

Para que as interfaces XUL possam executar linhas de código JavaScript, existem duas maneiras , pode ser especificada uma Tag "<script/>" no arquivo XUL com todo o conteúdo das rotinas em JavaScript, ou apenas apontado para um arquivo com os conteúdos escritos em JavaScript, deve ser utilizado o atributo *type* para especificar qual é a tipo do script que será carregado, neste caso *application/x-javascript* e o local onde se encontra o arquivo no atributo *src*.

As chamadas para as funções JavaScript são idênticas as chamadas HTML, podem ser adicionadas a qualquer elemento que tenha propriedades que chamam comandos como o onclick do elemento button por exemplo.

Exemplo de JavaScript no próprio arquivo XUL.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="xuledit"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
title="XUL and JavaScript" onload="init()">
<script  type="application/x-javascript">
function msg() {
  alert('JavaScript Test');
  return false;
}
</script>
<button id="open" label="Button with JavaScript" onclick="msg()"/>
</window>
```

Exemplo de arquivo XUL com uma chamada para arquivo JavaScript.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="xuledit"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
title="XUL and JavaScript" onload="init()">
<script  type="application/x-javascript" src=”test.js” />
<button id="open" label="Button with JavaScript" onclick="msg()"/>
</window>
```

Arquivo Test.js com o conteúdo JavaScript chamado no exemplo anterior.

```
function msg() {
  alert('JavaScript Test');
  return false;
}
```

5.11 XUL Com CSS.

O XUL trabalha com arquivos de estilos da mesma maneira que o HTML, as definições de estilos são salvas em arquivos .css e são chamados através da tag **xml-stylesheet** do próprio XML.na **Listagem 5.11a** temos um arquivo XUL que chama as definições de Skin padrões definidas pelo chrome, e na Listagem **Listagem 5.11b** temos um exemplo onde são definidos dois arquivo de estilo o padrão e mais um o arquivo que contém as definições de estilo personalizadas.

Listagem 5.11a – Chamada a CSS padrão do browser.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
</window>
```

Listagem 5.11b – Trabalhando com arquivo CSS padrão e personalizado.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<?xml-stylesheet href="estilo.css" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
</window>
```

5.12 XUL com C/C++.

Através do XPCOM podemos também trabalhar com bibliotecas compartilhadas desenvolvidas em C/C++ , como os famosos arquivos DLL do windows ou os arquivos SO das variações de Linux/Unix. As interfaces XUL podem interagir com estas bibliotecas para execução de regras de negocio ou acesso a dispositivos como som , teclado, vídeo, etc. Existem Diversos componentes já desenvolvidos em XPCOM e integrados ao Mozilla facilitando em muitos casos o desenvolvimento.

5.13 XUL com PHP.

Como o XUL e uma linguagem baseada em XML , HTML , CSS existe a possibilidade de interagir XUL com PHP, da mesma forma que o PHP interage com o HTML, quando a linguagem de scripts JavaScript não for a linguagem preferida ou o seu conhecimento nesta linguagem de script não for o suficientemente bom para programar a parte lógica da aplicação, pode ser feito uma conexão entre XUL e a linguagem script PHP. Provavelmente serão necessários alguns arquivos Javascript para intermediar a ligação entre XUL e PHP.

5.14 XUL com Python.

Atualmente existe uma implementação de XPCOM chamada PyXPCOM que permite a integração da tecnologia XUL com a linguagem Python, com este aplicativo instalado podem ser escritos diversos componentes Python que são chamados através do XPCOM e interagem com as

interfaces XUL.

5.15 XUL Versus Avalon XAML.

Em um futuro breve teremos uma tecnologia semelhante ao XUL que está sendo desenvolvida pela Microsoft na sua nova versão de sistema operacional o Windows Longhorn, que está previsto para 2006, a tecnologia se chama Avalon XAML, que também utiliza-se do XML , e permitirá que todas as aplicações e interfaces do sistema operacional sejam rodadas no browser, porém está tecnologia ainda está em desenvolvimento, diferentemente da tecnologia XUL que já está pronta e funcional, as principais vantagens atuais são a baixa complexidade da criação do XML e XUL, multiplataforma, código aberto, reutilização de componentes e recursos técnicos como o conhecimento em tecnologias para WEB , HTML, JavaScript, XML , diversos trabalhos desenvolvidos podem ser adaptados facilmente ao XUL, o que o torna viável também quando pensamos no tempo destinado a treinamento de novas tecnologias.

5.16 Portabilidade na Internacionalização e Localização.

Os produtos Mozilla são muito poderosos quando falamos em internacionalização (I18N) e Localização (L10N) , o Mozilla tem uma boa separação e estrutura para facilitar a tradução de suas interfaces, todos os arquivos de internacionalização e localização são arquivos DTD, quem podem ser editados também com o auxilio de uma ferramenta chamada Mozilla Translator que facilita o processo de tradução permitindo até mesmo pessoas não técnicas e sem nenhum conhecimento em XUL , XML ou outras tecnologias relacionadas editem e traduzam os arquivos DTD para sua linguagem nativa, o que permite o portabilidade de qualquer aplicativo construindo em XUL e a possibilidade de utilização da mesma ferramenta.

6. Capítulo 6 - FERRAMENTAS PARA DESENVOLVIMENTO EM XUL.

6.1 Editor de texto.

Para o desenvolvimento em XUL é necessários apenas um editor de texto de sua preferência como vi, kwrite, pico, e-macs, openoffice, notepad, textpad entre outros famosos editores.

6.2 JXUL.

Projeto do jXUL é um projeto open source para integrar a linguagem XUL a plataforma Java. Há diversas maneiras para esta integração ocorrer. No fato, este projeto expandirá com seus membros encontrar maneiras novas integrar estas duas tecnologias.

É um projeto ambicioso mais aparentemente sem continuidade, visto que seu último build no site da sorceforg.net foi do ano de 2001, seu código fonte inicial , versão 1.0 está disponível para download, o que pode ser aproveitado e a idéia de desenvolver uma API Java de interação com XUL. O projeto demonstra uma proposta de arquitetura na **Figura 6.2a**

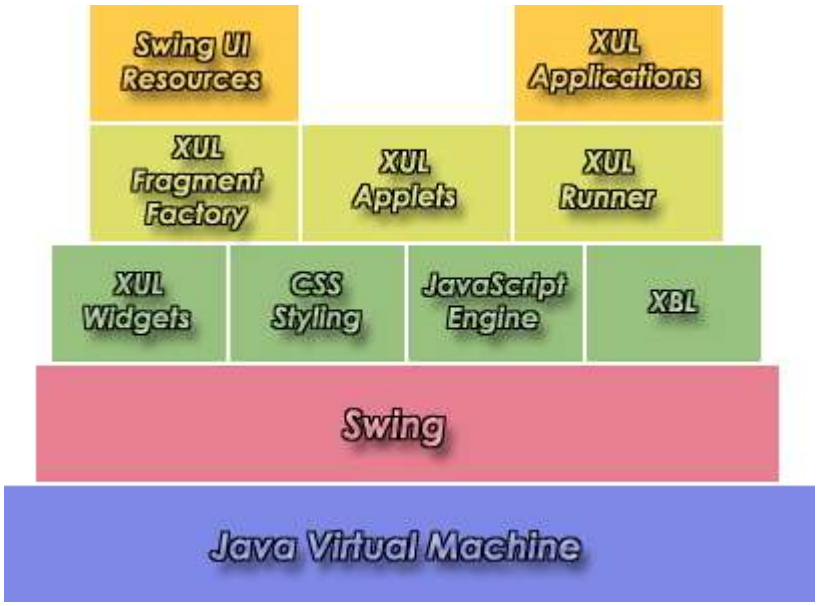


Figura 6.2.a – Arquitetura do JXUL.

6.3 XULMaker.

XULMaker é uma aplicação visual para construção de interfaces XUL, com ele é possível construir suas interfaces gráficas de forma visual , porém este projeto no momento da confecção deste material se encontrava fora de atividade, possivelmente descontinuado pelos seus desenvolvedores. Há última versão lançada que constava no sítio do projeto, era de 17 de agosto de 2004, a versão 0.51, que foi disponibilizada para trabalhar com o Mozilla 7.1 . Na **Figura 6.3a** podemos ver a ferramenta trabalhando com o Design de elementos e na **Figura 6.3b** a visualização do código fonte.

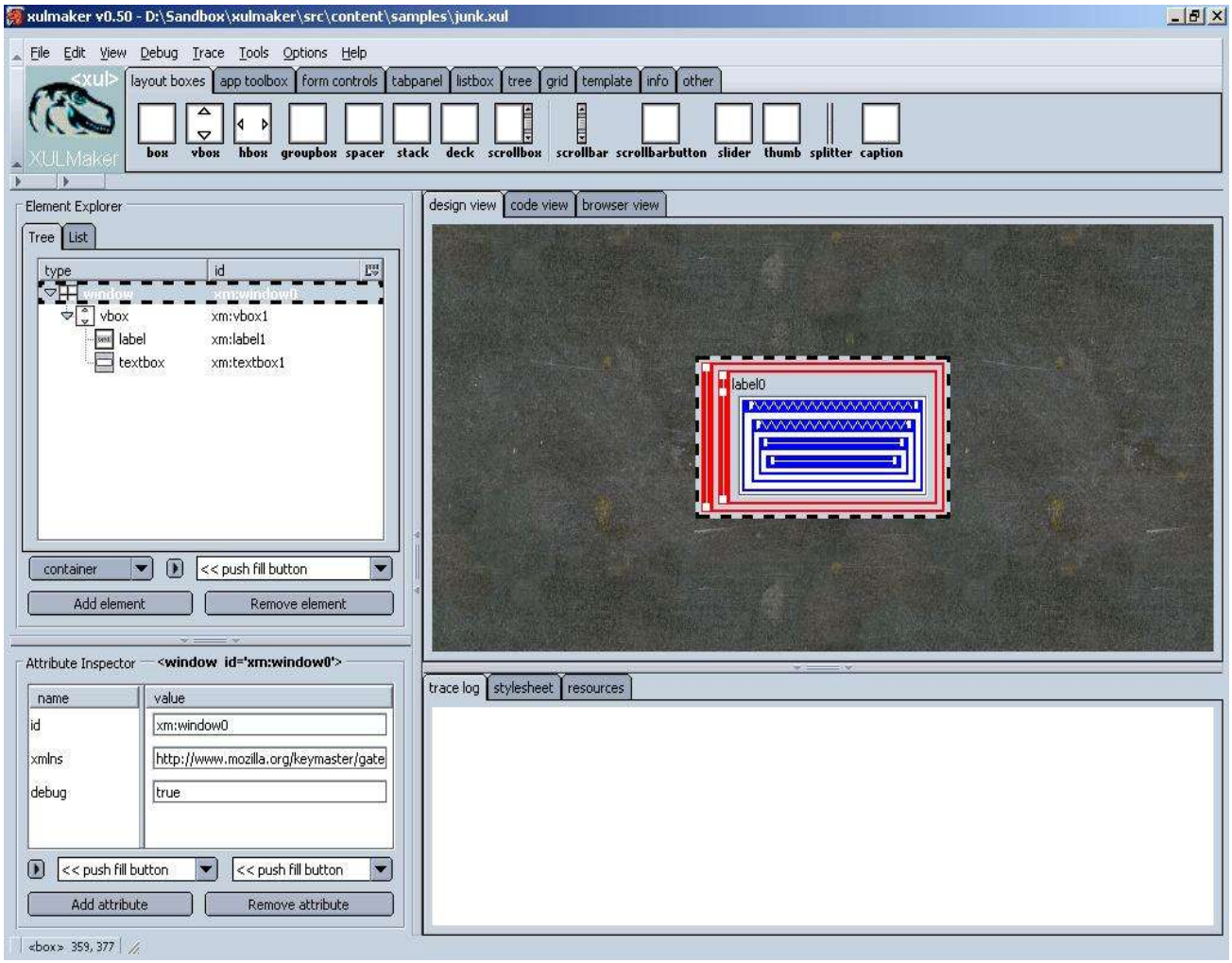


Figura 6.3.a – Visualização do Design no XULMaker.

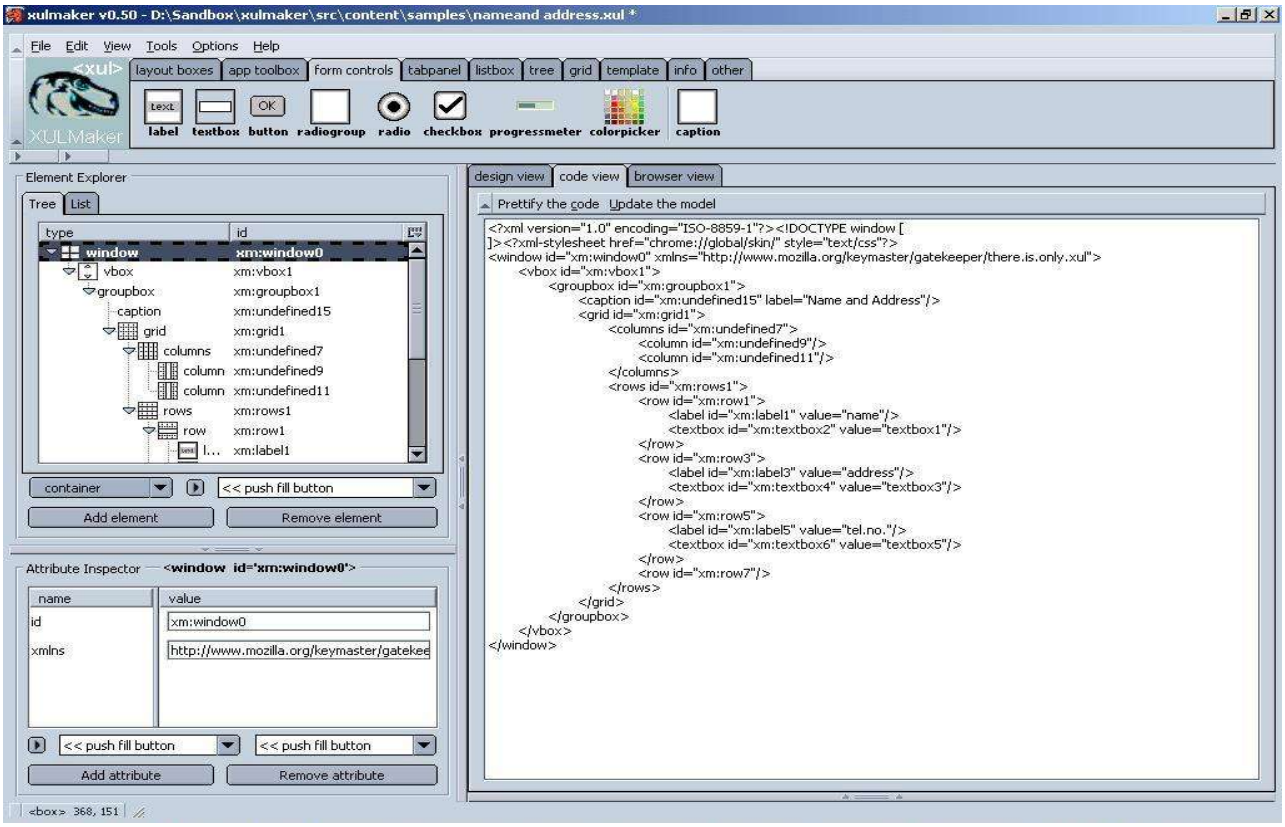


Figura 6.3.b – Visualização do código fonte no XULMaker.

Apesar de existir está ferramenta ainda muitos programadores da tecnologia XUL optam por ser mais rápido a edição diretamente por editores de textos. Existe um outro projeto no Mozdev.org com o nome de XULAPP que pretende criar uma IDE completa para a construção de interfaces XUL porém ainda foi iniciado muito recentemente.

6.4 DOM Inspector.

O DOM Inspector é uma ferramenta utilizada para inspecionar e editar DOM em tempo de execução na web ou em uma aplicação XUL. Podendo navegar na hierarquia do DOM usando o painel e a janela que permite uma variedade de vistas diferentes de todos os nós. Veja o DOM Inspector em ação na **Figura 6.4a**.

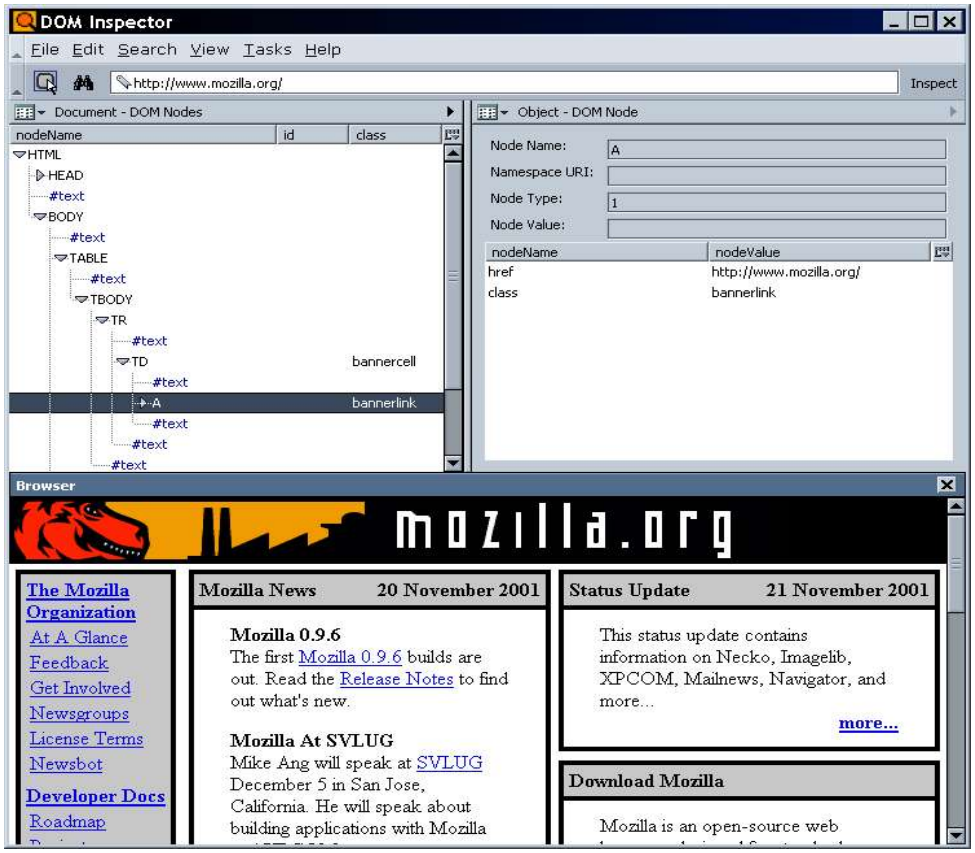


Figura 6.4.a – Visualização de nós do DOM com o DOM Inspector.

6.5 Venkman JavaScript Debugger.

Venkman JavaScript Debugger é o debugger de Javascript para browsers baseados Mozilla, tais como Mozilla 1.x, Netscape 7.x, e Phoenix/Firebird, fornecer do um ambiente poderoso de eliminar erros de Javascript para browsers baseados Mozilla, incluindo a série de browsers Netscape 7.x. Está ferramenta já vem integrada ao Mozilla desde a versão 1.0 que acompanhava com o Venkman 0.8.5.

XUL – Interfaces de usuários portáteis com tecnologia XML.

Atualmente já está sendo desenvolvidadas versões 0.9.x que contem muitas funcionalidades adicionais. A ferramenta se encontra em pleno desenvolvimento, na **Figura 6.5a** podemos visualizar o Venkman JavaScript Debugger.

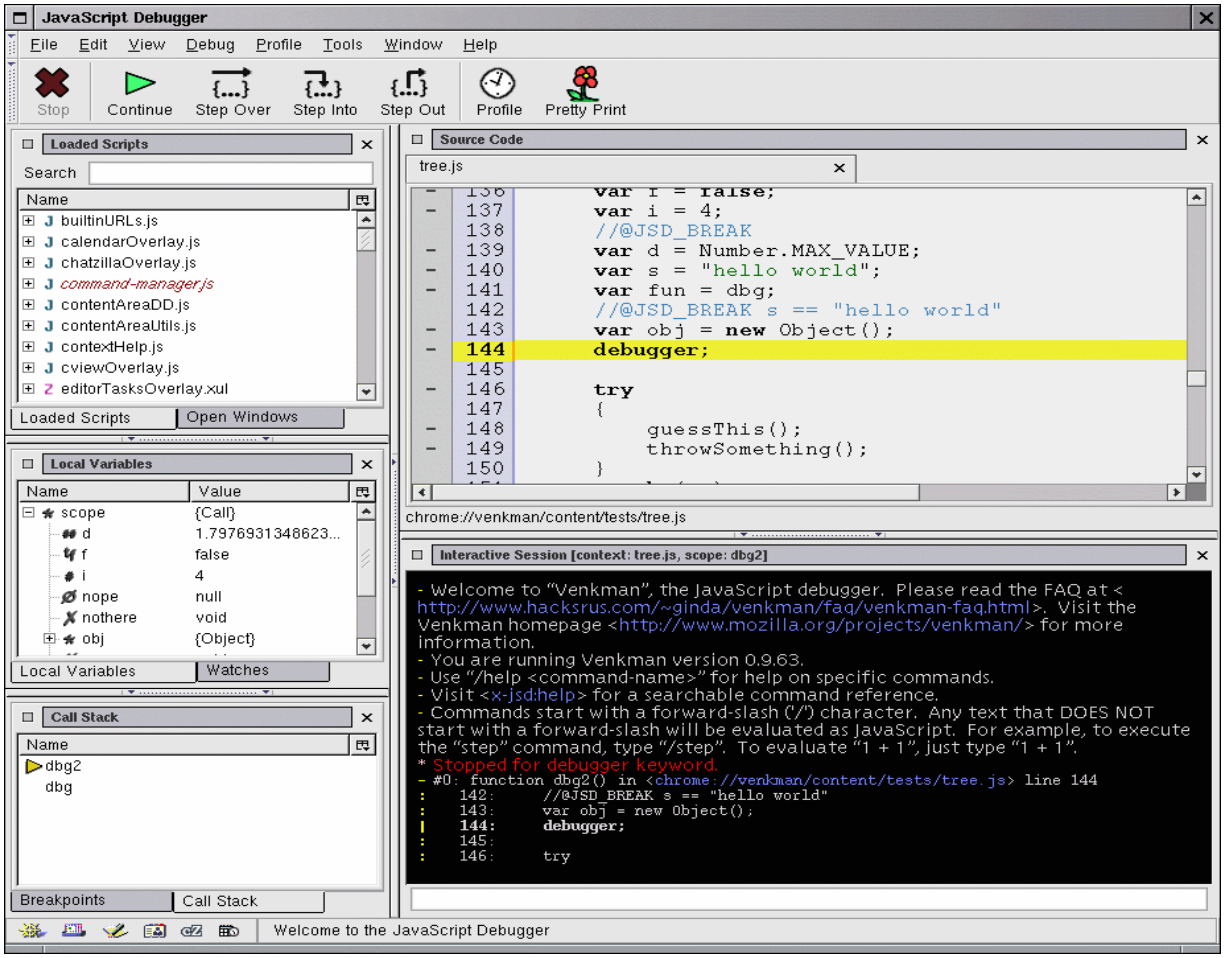


Figura 6.5.a – Venkman JavaScript Debugger.

6.6 Cview - XPCOM Component Viewer.

A ferramenta Cview é um componente visualizador de XPCOM, permite que você possa

navegar através dos componentes XPCOM incluídos em seu browser baseado no Mozilla, facilitando a consulta de componentes. A ferramenta é desenvolvida em XUL com Javascript, sendo assim possível de ser executada como um componente adicional do browser, a instalação pode ser feita automaticamente do sítio Hacksrus , que copiará um arquivo cview.jar para o diretório chrome do browser (instalação por arquivo xpi).

Após baixa-lo, para acessa-lo diretamente, também pode ser informado no browser o link <chrome://cview/content/cview.xul>. Veja **Figura 6.6a**.

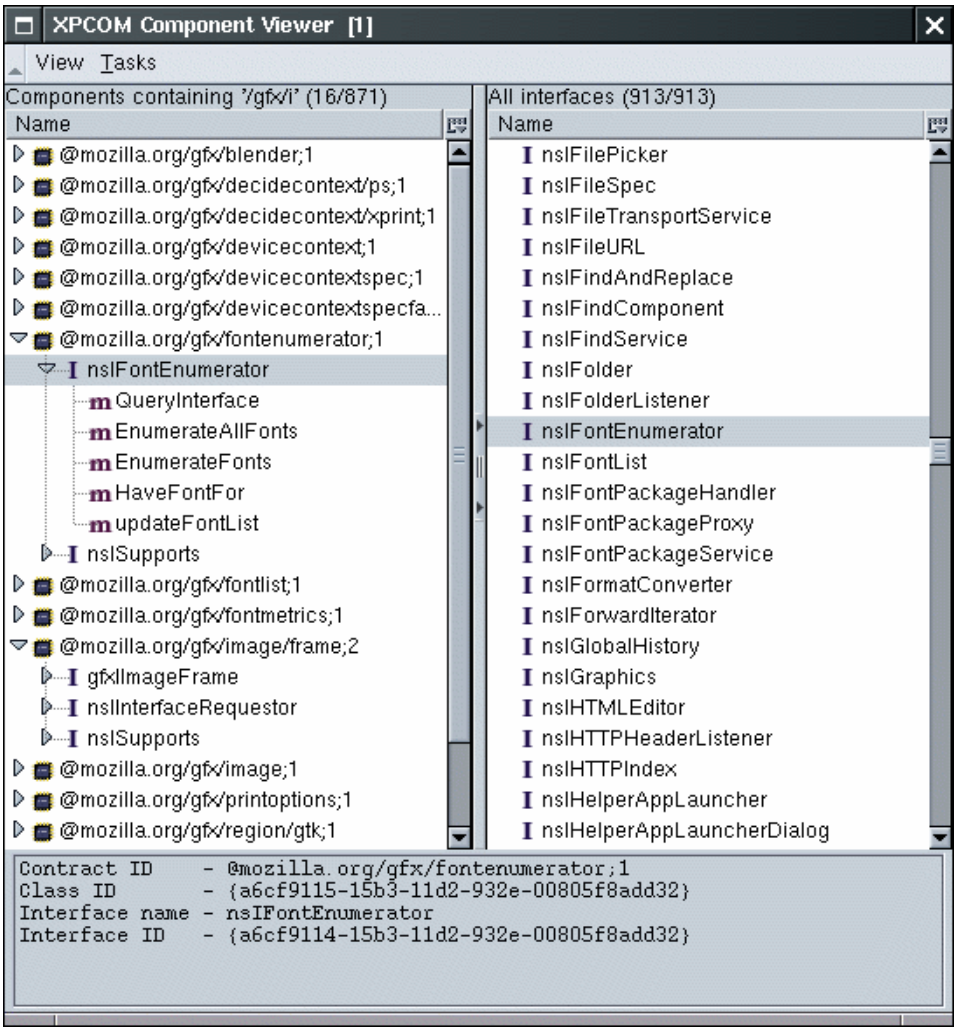


Figura 6.6.a – Cview em ação.

6.7 Mozilla Translator

O Mozilla Translator é um aplicativo especializado em editar arquivos DTD para internacionalização (I18N) e localização (L10N) do Mozilla ou aplicativos integrados. Ele permite facilmente o acesso aos dados dos arquivos DTD organizando a agilizando na tradução dos mesmos para outras Linguagens. A interface do Mozilla Translator pode ser vista na **Figura 6.7a.**

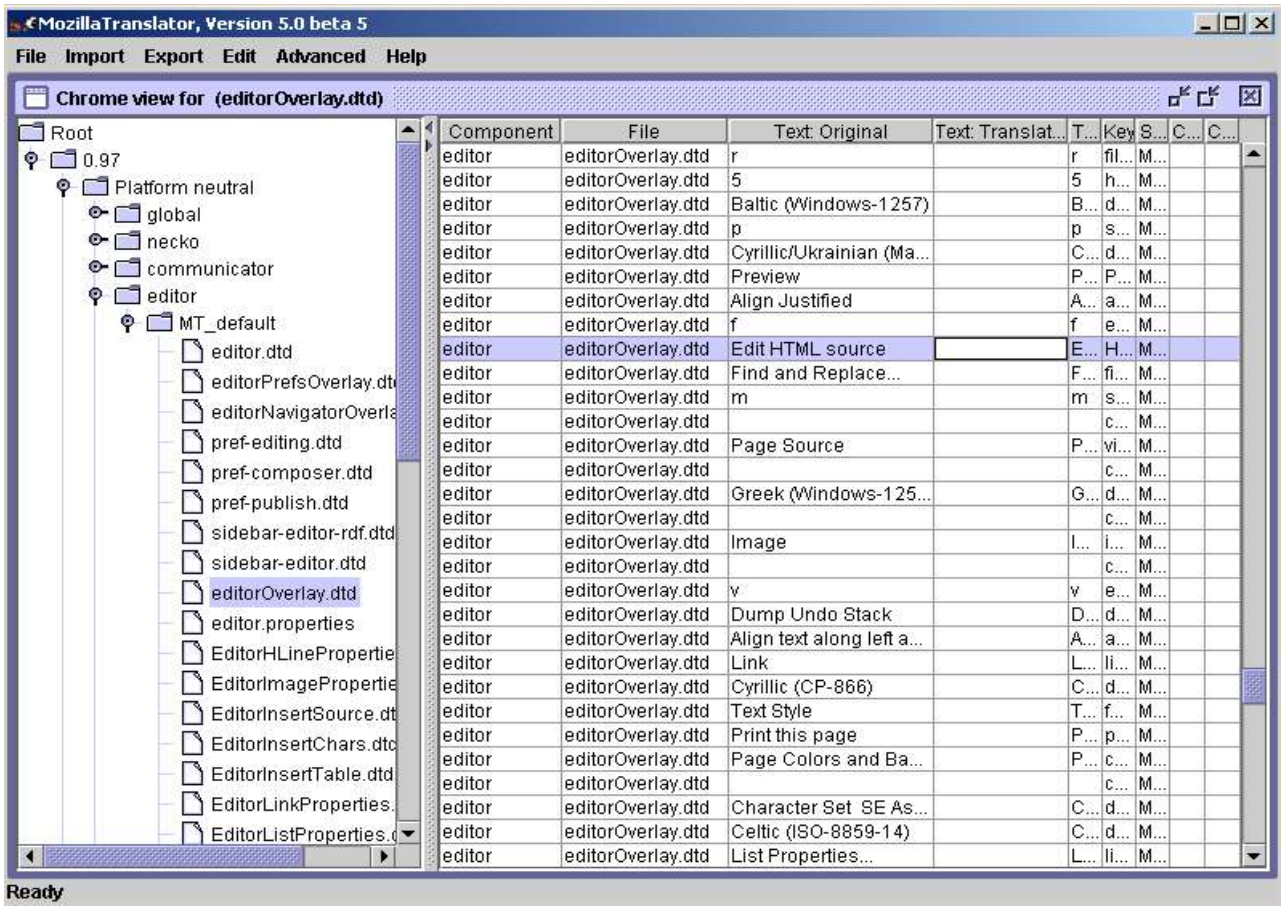


Figura 6.7a. - Mozilla Translator trabalhando com arquivos DTD.

7. Capítulo 7 - EXEMPLOS DE APLICAÇÕES XUL.

7.1 Barra de Busca Amazon.com (MAB).

Uma aplicação que pode mostrar algumas das funcionalidades e utilizades do XUL para web e uma interface que foi construída para fazer buscas na Amazon.com através de web services ,é uma interface rica em elementos XUL. O produto está na sua versão 1.3 e no site do projeto pode ser utilizado on-line a sua visualização pode ser vista na **Figura 7.1a.**

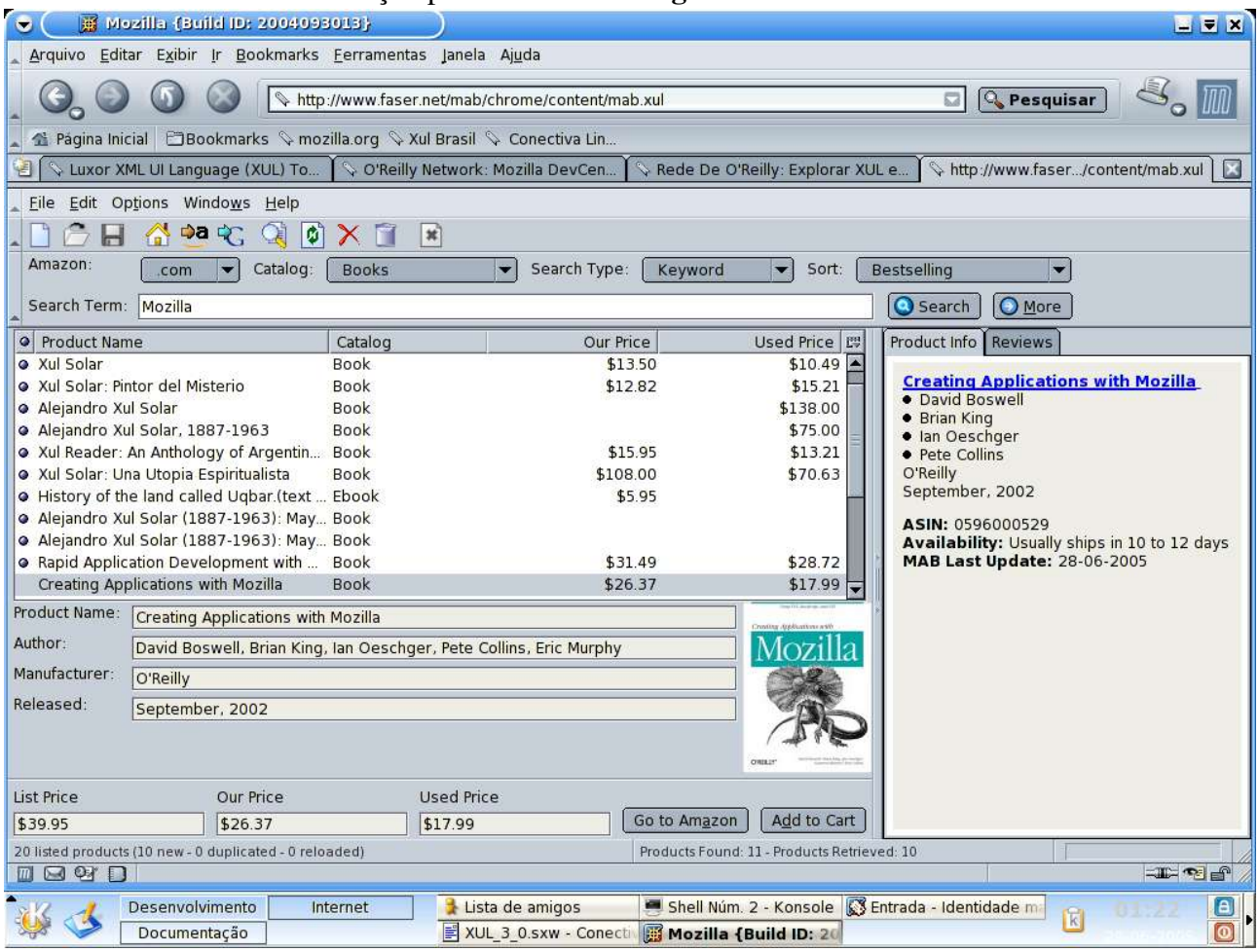


Figura 7.1a - Barra de Busca Amazon.com com interface rica em elementos.

7.2 Página de pesquisa da Google.com.

A Google também disponibiliza uma aplicação de consulta escrita na tecnologia XUL. Veja **Figura 7.2a**.

Figura 7.2a – Barra de Pesquisa XUL da Google.

7.3 Barra personalizada para o Orkut.

Com o XUL também é possível gerenciar um dos maiores sítios de relacionamento do mundo, o Orkut, para isso basta instalar o Oktut Toolbar nos browser da família Mozilla. Ele facilita na navegação e funcionalidades que existem no Orkut. Na **Figura 7.3a** o Orkut Toolbar.

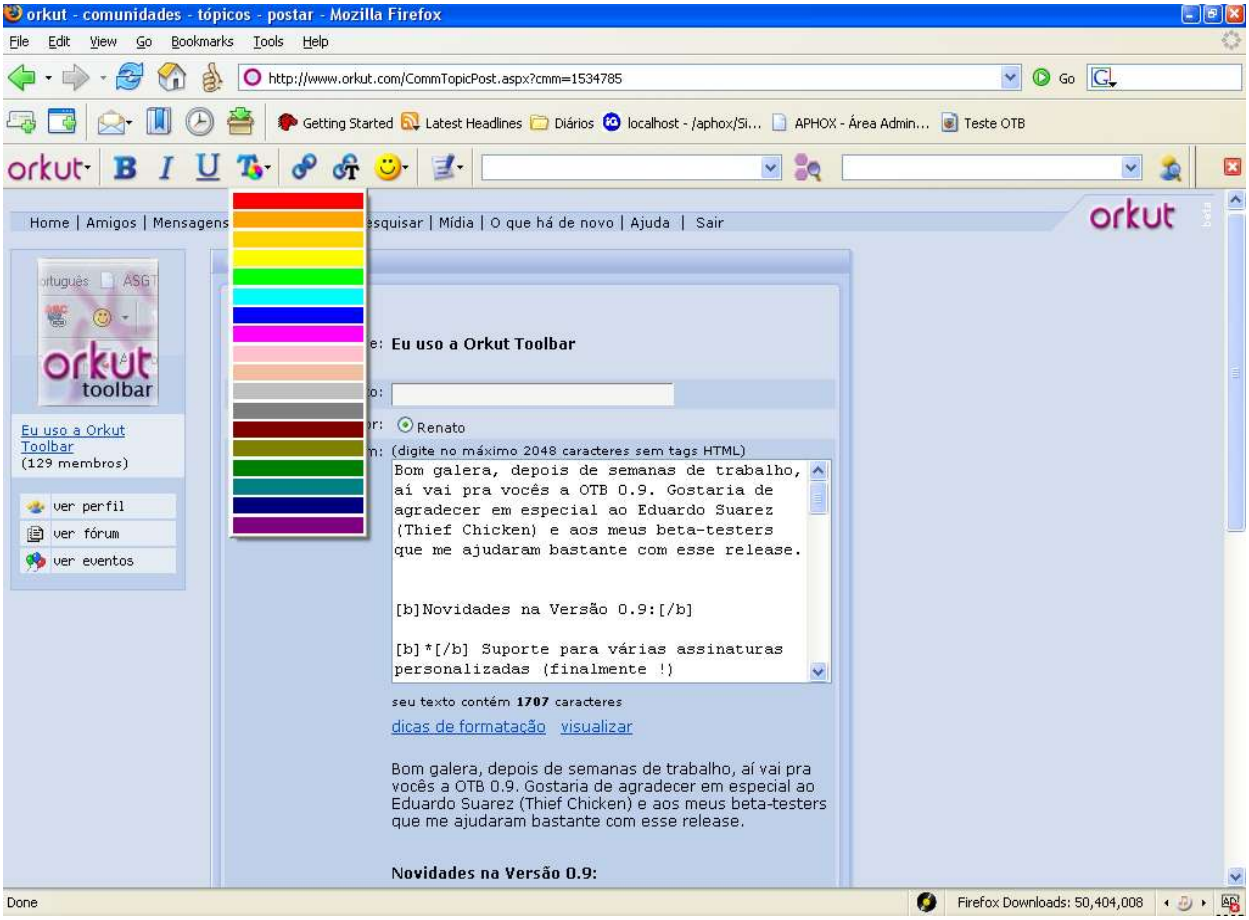


Figura 7.3a – Orkut Toolbar.

7.4 Projeto Imperador (webapp).

O projeto Imperador é uma Webapp que integra vários sistemas de gestão Administrativa como contas a pagar, contas a receber , controle de tesouraria, faturamento,etc. Este projeto trabalha com a tecnologias XUL integrado com a tecnologia Java e Arquivos DTD. Na **Figura 7.4a** é possível visualizar a interface do Imperador no browser Mozilla, o aplicativo fica disponível para demonstração na WEB.

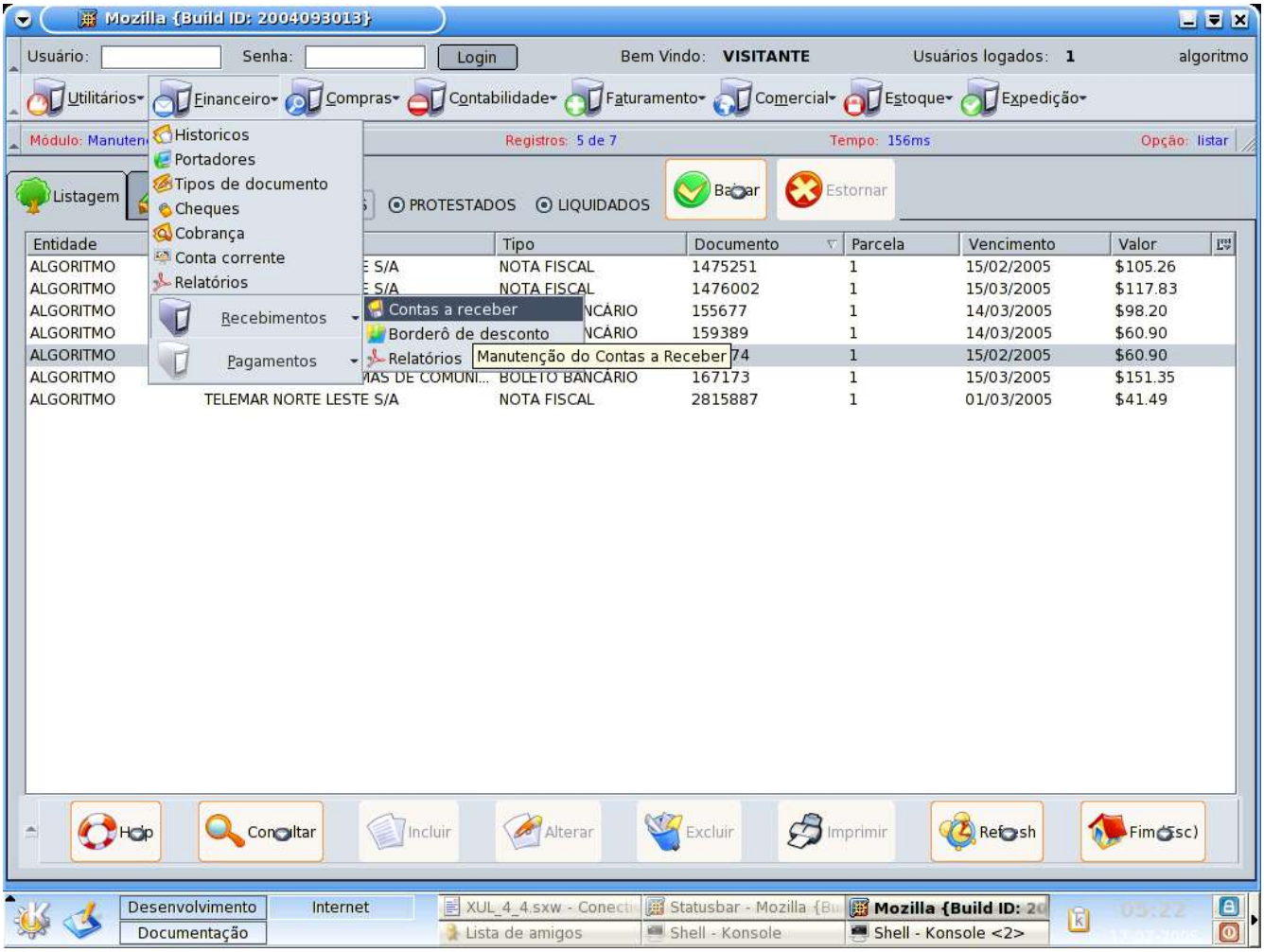


Figura 7.4a – Projeto Imperador na WEB acessado pelo browser Mozilla.

8. Capítulo 8 - ELEMENTOS DE INTERFACES GRÁFICAS DE USUÁRIOS EM XUL.

Serão descritos neste capítulo os principais elementos que podem ser utilizados para a criação de interfaces gráficas com a tecnologia XUL. Para que este capítulo possa ser utilizado também como uma referência de desenvolvimento de aplicações com a tecnologia XUL, serão apresentados os elementos com suas funcionalidades, um arquivo XUL de exemplo do elemento para que possam ter idéia da simplicidade da junção das tags XUL nos arquivos com estrutura XML, imagens do componente para visualizarem a riqueza dos elementos e sua semelhança com tecnologias para interfaces gráficas para desktop e a descrição de seus principais atributos, propriedades e métodos,

lembrando que serão especificados apenas alguns elementos sendo que existem muitos outros que não serão aqui apresentados.

8.1 Window.

O elemento window representa a janela principal de uma interface gráfica, nos arquivos XUL é o primeiro componente para iniciar o desenvolvimento de uma interface gráfica, entre as tags window ficarão descritos os demais elementos para a janela. Veja uma exemplo de uma janela criada pelo elemento window na **Listagem 8.1a** e o resultado na **Figura 8.1a**.

Listagem 8.1a – Elemento window.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
</window>
```

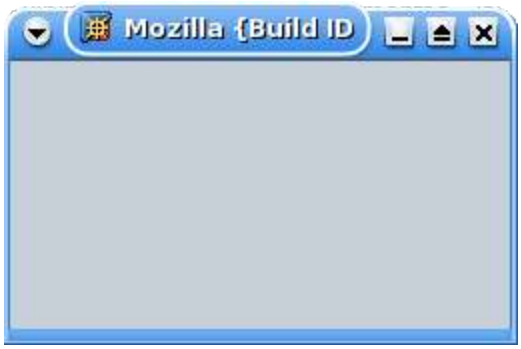


Figura 8.1a – Elemento window.

Atributo	*Tipo*	*Descrição*
Height	Integer	Altura da Janela.
Hidechrome	Booleano	Registra o chrome para poder ter ele incluído no titlebar hidden.
Id	Element id	Identificador da Janela.
ScreenX	Integer	Posição vertical que o janela aparece na tela.

Atributo	*Tipo*	*Descrição*
ScreenY	Integer	Posição vertical que o janela aparece na tela.
Sizemode	Pré-definido	Estado da Janela. • **Maximized** – Maximiza a janela, Ocupando a tela completa. • **Minimized** - A Janela e Minimizada , ou escondida. • **Normal** - A janela aparece em um estado normal.
Title	Alfa-Numérico	Texto da Barra de título da janela.
Width	Integer	Largura da Janela.
Windowtype	Alfa-Numérico	Ajuste uma string que possa ser usada identificar o tipo de janela.

8.2 Button.

Este elemento permite a criação de botões na interface, ele tem diversos atributos e propriedades para sua criação, podendo ser adicionado imagens no lugar de um label, ou ambos e tem como um dos principais atributo o command que é responsável por disparar uma execução de processo chamados através dos eventos de mouse ou teclado. Na **Figura 8.2a** podemos verificar um botão comum e na **Figura 8.2a** um botão com uma imagem e na **Listagem 8.2a** dois botões em uma interface.

Listagem 8.2a – Janela com Button Simples e Button com Imagem.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<button  label="Pressione-me"
          oncommand="alert('Você me pressionou!'); return false;"/>

<button  image="save.gif"  label="Pressione-me com imagem"
           oncommand="alert('Você pressionou'); return false;"/>
</window>
```

Figura 8.2a – Button.

Figura 8.2b – Button com Imagem.

8.3 Label.

Este elemento é utilizado para cria uma etiqueta de texto para um elemento de controle, diversos elementos tem este elemento como atributo como por exemplo o button, radio, etc. Na **Listagem 8.3a** temos um janela com um elemento label e na **Figura 8.3a** a sua visualização.

Listagem 8.3a – Elementos Label.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<label value="Element Label"  />
<label disabled="true" value="Element Label disabled"  />
<label class="header"  value="Element Label with header" />
<label class="indent"  value="Element Label with indent" />
<label class="monospace"  value="Element Label with monospace" />
<label class="plain"  value="Element Label with plain" />
<label class="small-margin"  value="Element Label with small-margin" />
</window>
```

Figura 8.3a – Elementos Label.

Atributo	***Tipo***	***Descrição***
Accesskey	caracter	Indica qual será a letra para utilizada como tecla de atalho do elemento, esta letra ficará sublinhada.Quando o usuário pressionar ALT (A tecla de acessibilidade pode variar de acordo com a plataforma) e a chave de acesso a função do elemento será ativada.
Control	Elemento	Neste atributo deve ser ajustado o elemento ao qual o label está associado para ajustado a sua identificação.
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. ● **start** O texto será encolhido para seu lado esquerdo. ● **end** O texto será encolhido para seu lado direito. ● **left** o texto será encolhido para seu lado esquerdo. ● **right** o texto será encolhido para seu lado direito. ● **center** O texto será encolhido para ambos os lados. ● **none** O texto será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).
Disabled	booleano	Indica se elemento está desabilitada ou não. Quando verdadeiro mostra o elemento em uma cor mais transparente que os demais.
Value	Alfa-numérico	Texto que aparecerá no elemento de controle.

Propriedades e Métodos	***Tipo***	***Descrição***
AccessKey	Caracter	Ajusta o valor do atributo accesskey.
Accessible	nslaccessible	Retorna o objeto da acessibilidade para o elemento.
Control	Elemento	Ajusta o valor do atributo control.

Propriedades e Métodos	***Tipo***	***Descrição***
Crop	Alfa-numérico	Ajusta o valor do atributo crop.
Disabled	Booleano	Ajusta o valor do atributo disabled.
Value	Alfa-numérico	Ajusta o valor do atributo value.

Classes de Estilo(class)	***Descrição***
Header	Utilizada para deixar o texto em negrito.
Indent	Mostra uma indentação do texto.
Monospace	Mostra o texto com espaçamento único entre os caracteres.
Plain	Utilizado para que seja desconsiderada qualquer margem do texto.
Small-margin	Mostra uma pequena margem no texto.

8.4 Radio.

Este elemento sempre é utilizado dentro em conjunto com um grupo de elementos , outros elementos radio e um radiogroup que é responsável pelo agrupamento de elementos radio. Somente um elemento radio pode estar selecionado dentro do radiogroup. Na **Listagem 8.4a** temos componentes radio agrupados a três dentro de um elemento radiogroup na **Figura 8.4a** pode ser visualizado o resultado.

Listagem 8.4a – Elementos Radio.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<groupbox>
<radiogroup>
  <radio value="1" selected="true" label="radio 1"/>
  <radio value="2" label="radio 2"/>
  <radio value="3" label="radio 3"/>
</radiogroup>
</groupbox>
<groupbox>
<caption label="radios with image"/>
<radiogroup>
  <radio value="a" label="radio a" src="save.gif" />
  <radio value="b" label="radio b" src="print.gif"/>
  <radio value="c" label="radio c" src="start.gif"/>
</radiogroup>
</groupbox>
</window>
```

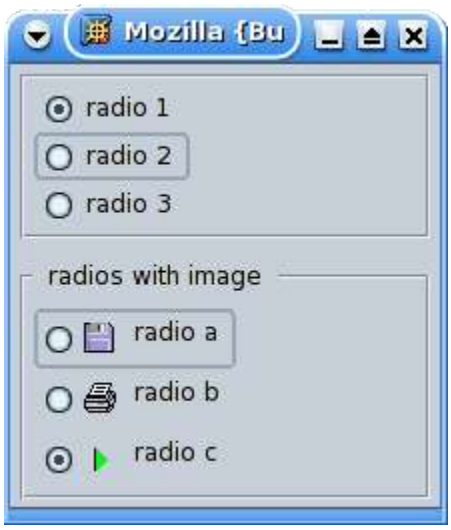


Figura 8.4a – Elementos Radio.

Atributo	***Tipo***	***Descrição***
Accesskey	caracter	Indica qual será a letra para utilizada como tecla de atalho do elemento, esta letra será sublinhada.Quando o usuário precionar ALT (A tecla de acessibilidade pode variar de acordo com a plataforma) e a chave de acesso a função do elemento será ativada.
Command	Elemento	Ajusta a identificação do elemento ao comando a ser executado.
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. ● start O texto será encolhido para seu lado esquerdo. ● end O texto será encolhido para seu lado direito. ● left **(desaprovado)** o texto será encolhido para seu lado esquerdo. ● right **(desaprovado)** o texto será encolhido para seu lado direito. ● center O texto será encolhido para ambos os lados. ● none O texto não será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).
Disabled	booleano	Indica se elemento está desabilitada ou não.geralmente mostra o elemento em uma cor mais transparente que os demais.
Focused	Booleano	Ajusta o foco do cursor no de elemento.
Label	Alfa-numerico	Texto que aparecerá no elemento de controle.
Selected	booleano	Este atributo ajusta qual elemento radio será selecionado dentro de um elemento radiogroup. Para mudar o radio selecionada atualmente, modifique o selectedIndex ou a propriedade do selectedItemdo radiogroup.
Src	Alfa-numerico (url)	A URL da imagem que aparece na tecla. Caso não seja informado nenhuma imagem aparecerá.

Atributo	***Tipo***	***Descrição***
value	Alfa-Numérico	Associa um valor dos dados com cada radio button.

Propriedades e Métodos	***Tipo***	***Descrição***
Accesskey	Caracter	Ajusta o valor do atributo accesskey.
Accessible	Nslaccessible	Retorna o objeto da acessibilidade para o elemento.
Crop	Alfa-numérico	Ajusta o valor do atributo crop.
Disabled	Booleano	Ajusta o valor do atributo disabled.
Label	Alfa-numérico	Ajusta o valor do atributo label.
RadioGroup	RadioGroup	Retorna o RadioGroup que o elemento radio está incluido.
Selected	Booleano	Retorna verdadeiro se o radio for selecionado.
Value	Alfa-numérico	Ajusta o valor do atributo value.

8.5 Radiogroup.

Este elemento e utilizado para agrupar elementos radio. Permite que apenas um elemento radio seja selecionado de cada vez. Na **Listagem 8.5a** podemos ver o radiogroup fazendo o seu trabalho de agrupamento de elementos radio, o visual pode ser visto em **Figura 8.5a**.

Listagem 8.5a – Elemento radiogroup.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<groupbox>
<caption label="radiogroup 1" value="rg1" />
<radiogroup >
  <radio label="radio a" />
  <radio label="radio b" />
  <radio label="radio c" />
</radiogroup>
</groupbox>
</window>
```

Figura 8.5a – Elemento Radiogroup.

Atributo	***Tipo***	***Descrição***
Disabled	booleano	Indica se elemento está desabilitada ou não.geralmente mostra o elemento em uma cor mais transparente que os demais.
Focused	Booleano	Ajusta o foco do cursor no de elemento.
Tabindex	Inteiro	A ordem da aba do elemento. é a ordem em que o foco é movido quando o usuário pressiona a chave de aba.
Value	Alfa-numérico	Associa um valor de dado com o elemento.

Propriedades e Métodos	***Tipo***	***Descrição***
Accessible	nslaccessible	Retorna o objeto da acessibilidade para o elemento.
AppendItem (label e value)	Radio	Cria um elemento radio novo e adiciona-o à extremidade do radiogroup.
CheckAdjacentAlement (dir)	Nenhum	Desmarca a seleção que está feita atualmente e seleciona o radio seguinte ou procedente.
Disabled	Booleano	Retorna e Ajusta o Atributo Disabled.
FocusedItem	Radio	Prende o item atualmente focalizado no radiogroup.
InsertItemAt (index,label ou value)	Radio	Cria um elemento de radio novo e o introduz em uma posição específica no radiogroup.

Propriedades e Métodos	***Tipo***	***Descrição***
RemoveItemAt (index)	Radio	Remove um item do radiogroup
SelectedIndex	Radio	Prende o índice do item atualmente selecionado no radiogroup.
SelectedItem	Radio	Prende o item atualmente selecionado no radiogroup.
Value	Alfa-numérico	Ajusta o valor do atributo value.

8.6 Checkbox.

Este elemento é utilizado para criação de caixas de seleção, com uma opção de caixa , que pode indicar marcada ou desmarcada pelo usuário. Na **Listagem 8.6a** e **Figura 8.6a** demonstramos o elemento checkbox em uma interface onde o usuário dever marcar as linguagens de programação que ficarão habilitadas.

Listagem 8.6a – Elementos Checkbox.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<checkbox label="Habilitar C#" checked="false"/>
<checkbox label="Habilitar Ruby" checked="false" />
<checkbox label="Habilitar Java" checked="true"/>
<checkbox label="Habilitar XUL" checked="true"/>
<checkbox label="Habilitar Boo" checked="false"/>
</window>
```

Figura 8.6a – Elementos checkbox.

Atributo	***Tipo***	***Descrição***
Accesskey	caracter	Indica qual será a letra para utilizada como tecla de atalho do elemento, esta letra será sublinhada. Quando o usuário pressionar ALT (A tecla de acessibilidade pode variar de acordo com a plataforma) e a chave de acesso a função do elemento será ativada.
Checked	Booleano	Indica se estado do checkbox.
Command	Elemento	Ajusta a identificação do elemento ao comando a ser executado.

Atributo	***Tipo***	***Descrição***
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. ● start O texto será encolhido para seu lado esquerdo. ● end O texto será encolhido para seu lado direito. ● left **(desaprovado)** o texto será encolhido para seu lado esquerdo. ● right **(desaprovado)** o texto será encolhido para seu lado direito. ● center O texto será encolhido para ambos os lados. ● none O texto será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).
Disabled	booleano	Indica se elemento está desabilitada ou não. geralmente mostra o elemento em uma cor mais transparente que os demais.
Image	Alfa-numerico (url)	A URL da imagem que aparece na tecla. A posição da imagem e determinada pelo atributo dir e pelo atributo orient.
Label	Alfa-numerico	Texto que aparecerá no elemento de controle.
Tabindex	Inteiro	A ordem do elemento. é a ordem em que o foco é movido quando o usuário pressiona a chave.

Propriedades e Métodos	***Tipo***	***Descrição***
Accesskey	Caracter	Ajusta o valor do atributo accesskey.
Accessible	NslAccessible	Retorna o objeto da acessibilidade para o elemento.

Propriedades e Métodos	***Tipo***	***Descrição***
Checked	Booleano	Ajusta o valor do atributo Checked.
Command	Elemento	Ajusta o elemento do atributo command.
Crop	Alfa-numérico	Ajusta o valor do atributo crop.
Disabled	Booleano	Ajusta o valor do atributo disabled.
Image	Url	Ajusta o valor do atributo Image.
Label	Alfa-numérico	Ajusta o valor do atributo label.

8.7 Script.

Este elemento é utilizado para declarar scripts que serão utilizados pela janela XUL , os scripts devem ser escritos em arquivos separados e ser referenciados por está tag, mais nada impede que os scripts sejam escritos dentro da tag script , mas por organização sempre e melhor cria-los em outros arquivos. Na **Listagem 8.7a** temos um exemplo de um arquivo XUL contendo um script JavaScript e na **Listagem 8.7b** temos uma declaração de uma chamada a um arquivo de script no arquivo XUL.

Listagem 8.7a - Arquivo XUL contendo Script.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="xuledit"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
title="XUL and JavaScript" onload="init()">
<script  type="application/x-javascript">
function msg() {
  alert('JavaScript Test');
  return false;
}
</script>
<button id="open" label="Button with JavaScript" onclick="msg()"/>
</window>
```

Listagem 8.7b - Declaração de uma chamada a um arquivo de script.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="xuledit"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
onload="init()">
<script  type="application/x-javascript" src=”test.js” />
<button id="open" label="Button with JavaScript" onclick="msg()"/>
</window>
```

Atributo	*Tipo*	*Descrição*
Src	Script URL	URL para o arquivo de script.
Type	Linguagem	Deve ser especificado o tipo da liguagem script. Uma das linguagens script mais utilizadas e a JavaScript com a seguir: **'application/x-javascript'.**

8.8 Splitter.

O componente Splitter se comporta como uma margem divisória da interface gráfica , uma linha para separação de componentes, que pode ser utilizada na vertical ou horizontal dos componentes. Na **Listagem 8.8a** temos um Splitter separando dois elementos textbox e na **Figura 8.8a** a sua visualização.

Listagem 8.8a – Componente Splitter.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<textbox multiline="true" />
<splitter id="split" />
<textbox multiline="true" />
</window>
```

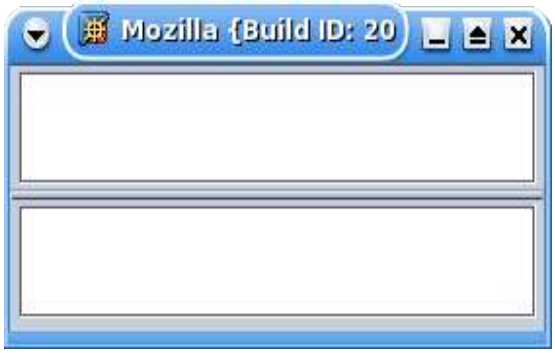


Figura 8.8a - Elemento splitter separando dois textbox.

Atributo	*Tipo*	*Descrição*
Collapse	Pré-definido	Determina o tipo de margem de um elemento splitter e a ação que ocorre ao clicar sobre o elemento. • **None** – Nada Ocorre. • **Before** – Define que ao clicar no splitter ele diminuirá a altura, dividindo elementos, neste caso na divisão anterior. • **after** - Define que ao clicar no splitter ele diminuirá a altura, dividindo elementos, neste caso na divisão posterior.
Resizeafter	Pré-definido	Indica que o elemento à direita ou abaixo do splitter deve alterar o tamanho quando o splitter e reposicionado. • **Closest** – Elemento imediatamente à direita ou abaixo. • **Farthest** – Elemento que está mais distante à direita ou abaixo. • **Grow** – Os elementos à direita ou abaixo não alteram o tamanho.
Resizebefore	Pré-definido	Indica que o elemento à esquerda ou acima do splitter deve alterar o tamanho quando o splitter e reposicionado. • **Closest** - Elemento imediatamente à esquerda ou acima. • **Farthest** - Elemento que está mais distante à esquerda ou acima.

Atributo	*Tipo*	*Descrição*
State	Pré-definido	Indica se o splitter tem índice collapse ou não. Este atributo será atualizado automaticamente quando o splitter e movido. Tem uma aparencia diferente para cada estado. • **Open** - índice qualquer um antes ou depois do divisor. É o default. • **Collapsed** - índice qualquer um antes ou depois do divisor, não é visível. • **Dragging** - O usuário ajusta a posição do divisor com o mouse.

A Classe de Estilo permite o uso de elementos de estilo no elemento splitter já pré-determinado , esta classe é utilizada em conjunto do elemento treecol onde o splitter faz a separação das colunas do objeto permitindo que o usuário araste as colunas.

Estilo de Classe	*Descrição*
Tree-splitter	É utilizado com o elemento treecol para a separação de colunas.

8.9 Textbox.

Este elemento é utilizado para dar entrada em um texto, é chamado de caixa de texto, podendo ser um texto apenas de uma linha ou utilizado o atributo multiline para especificar diversas linha de texto. Na **Listagem 8.9a** podemos analisar alguns elementos textbox e a sua visualização na **Figura 8.9a**.

Listagem 8.9a – Elementos Textbox.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul" >
 <label control="textbox" value="Simple Textbox:"/>
 <textbox id="textbox" />
 <label control="textbox2" value="Textbox disabled:"/>
 <textbox id="textbox2" disabled="true" value="Text in the Textbox disabled"/>
 <label control="textbox3" value="Textbox Multiline:"/>
 <textbox id="textbox3" multiline="true" />
 <textbox id="textbox4" value="Textbox with readonly" readonly="True" />
</window>
```

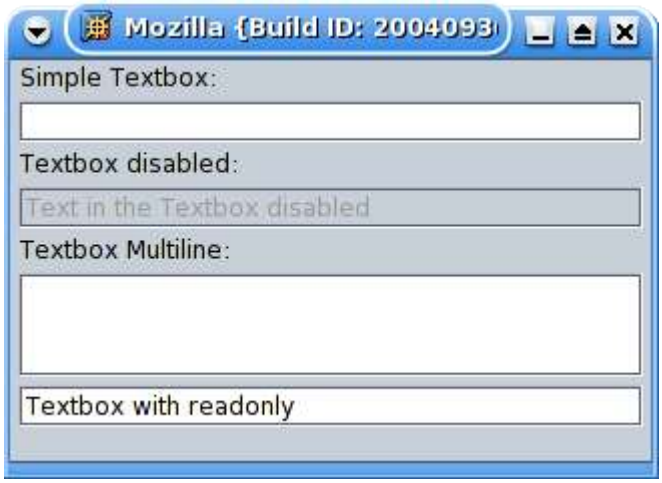


Figura 8.9a – Elementos Textbox.

Atributo	***Tipo***	***Descrição***
Cols	Inteiro	Numero de colunas da area de texto.
Disabled	Boolean	Indica se elemento está desabilitada ou não. Quando verdadeiro mostra o elemento em uma cor mais transparente que os demais.
Maxlength	Inteiro	Numero maximo de caracteres do textbox.
Multiline	Boolean	Quando verdadeiro permite que sejam criadas novas linhas no elemento textbox.
Onchange	Script code	Este evento é acionado quando o valor do textbox é alterado. somente quando o foco for movido para outro elemento.

Atributo	***Tipo***	***Descrição***
Oninput	Script code	Este evento é emitido quando um usuário adicionaum texto a um textbox.
Readonly	Boolean	Quando verdadeiro deixa o elemento textbox com permissão apenas de leitura para o usuário, não sendo permitido alteração no texto que se encontra no textbox.
Rows	Inteiro	Numero de linhas do Textbox.
Size	Inteiro	Numero de caracteres mostrados no textbox.
Tabindex	Inteiro	A ordem da aba do elemento. é a ordem em que o foco é movido quando o usuário pressiona a chave de aba.
Timeout	Inteiro	O número em milissegundos antes do temporizador acionar um evento de comando.
Type	Pré-definido	Indica o Tipo de Textbox, para textbox normais não especificar este atributo. • **Autocomplete** – Tem suporte a autocomplete. • **Password** – O texto é protegido, em formado de senha (***) • **Timed** – dispara um evento de comando.
Value	Caracter	Descrição ou texto do elemento.
Wrap	Caracter	Ajuste este atributo com o valor **'off'** para desabilitar o word wrapping no textbox.

8.10 Treecol.

Este elemento e responsável pela construção de colunas, é muito utilizado para criar colunas dentro do componente tree, formando o cabeçalho das colunas. Na **Listagem 8.10a** temos descrito dois exemplos de elementos treecol um com o rótulo descrito para cada coluna e outro com imagens para cada coluna, a visualização pode ser vista na **Figura 8.10a**

Listagem 8.10 – Elementos TreeCol.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<label value="Basic Treecol" />
<treecols>
  <treecol persist="hidden width" flex="1" id="meta-1" label="column 1"/>
  <splitter class="tree-splitter"/>
  <treecol persist="hidden width" flex="1" id="meta-2" label="column 2"/>
  <splitter class="tree-splitter"/>
  <treecol persist="hidden width" flex="1" id="meta-3" label="column 3"/>
</treecols>
<label value="Treecol with Tree Element" />
<tree id="metatree2" flex="1">
<treecols>
  <treecol-image src="save.gif" />
  <treecol persist="hidden width" flex="1" id="metaX-1" label="column 1" />
  <splitter class="tree-splitter"/>
  <treecol persist="hidden width" flex="1" id="metaX-2" label="column 2"/>
  <splitter class="tree-splitter"/>
  <treecol persist="hidden width" flex="1" id="metaX-3" label="column 3"/>
  </treecols>
  <treechildren flex="1"/>
</tree>
<label value="Treecol with image"/>
<tree id="metatree3" flex="1">
<treecols>
  <treecol-image src="save.gif" />
  <treecol persist="hidden width" flex="1" id="metaY-1" class="treecol-image"
src="save.gif" />
  <splitter class="tree-splitter"/>
  <treecol persist="hidden width" flex="1" id="metaY-2" class="treecol-image"
src="print.gif" />
  <splitter class="tree-splitter"/>
  <treecol persist="hidden width" flex="1" id="metaY-3" class="treecol-image"
src="start.gif"/>
  </treecols>
<treechildren flex="1"/>
</tree>
</window>
```

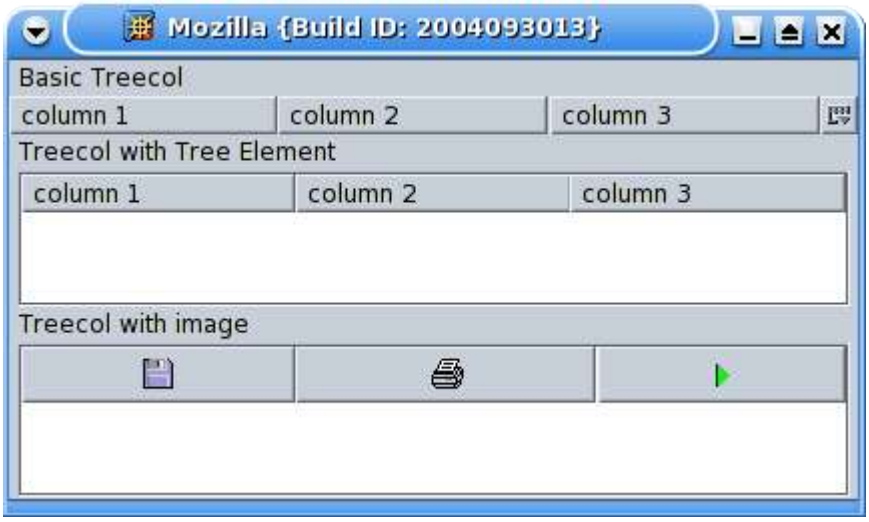


Figura 8.10a – Elementos Treecol.

Atributo	***Tipo***	***Descrição***
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. ● **start** O texto será encolhido para seu lado esquerdo. ● **end** O texto será encolhido para seu lado direito. ● **left** O texto será encolhido para seu lado esquerdo. ● **Right** O texto será encolhido para seu lado direito. ● **center** O texto será encolhido para ambos os lados. ● **none** O texto não será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).

Atributo	***Tipo***	***Descrição***
Cycler	Booleano	Quando este atributo é verdadeiro o conteúdo da coluna e demonstrado com um circulo que pode ser marcado ou desmarcado, um elemento parecido e com as mesmas funcionalidade de um checkbox, porém com uma aparência circular.
Dragging	Booleano	Este atributo será ajustado para identificar se a coluna estiver sendo arrastada. Este atributo é ajustado automaticamente.
Fixed	Booleano	Se verdadeiro, o tamanho da coluna no treecol não pode ser ajustado pelo usuário.
Hidden	Booleano	Esconde a coluna.
Hideheader	Booleano	Indica que a descrição da coluna do treecol deve ser indicado vazia.
Ignorecolumnpicked	Booleano	Se verdadeira, a coluna não aparece no picker da coluna.
Label	Alfa-Númerico	Descrição da coluna.
Sort	URI of an RDF predicate	Classifica dados vindo de arquivos RDF.
SortActive	Pré-definido	Este atributo define como a coluna e ordenada. • **ascending**: Os dados são ordenados em ordem crescente. • **descending**: Os dados são ordenados em ordem decrescente. • **natural:** os dados não são ordenados e são demonstrados conforme a ordem que foram inseridos.
Src	URL imagem	Informa a imagem que será mostrada no local do label da coluna, para que a imagem aparece deve ser adicionado o atributo class com “treecol-image” ao elemento treecol.

Atributo	*Tipo*	*Descrição*
Type	Pré-definido	Define o tipo do conteúdo que será apresentado no treecol • **checkbox** mostra as colunas como um elemento checkbox • **progressmeter** mostra as colunas como um elemento progressmeter. • **text** mostra as colunas como texto.

Propriedades e Métodos	*Tipo*	*Descrição*
Acessible	Nslaccesible	Retorna acessibilidade do objeto treecol.

Estilo Classe	*Descrição*
Treecol-image	Esta classe adiciona uma imagem no cabeçalho da coluna do elemento treecol. Deve ser definido atributo class e o src com o URI para imagem.

8.11 Description.

Este elemento é utilizado para a criação de blocos de texto, o texto pode ser inserido de duas maneiras, no atributo value ou ainda entre o inicio e o fim das tags description. Na **Listagem 8.11**a podemos observar três elementos description entre o elemento splitter, um description com o texto separado pelas tags e outros dois utilizando do atributo value, que resultam na imagem apresentada na **Figura 8.11a**.

Listagem 8.11a – Elementos Description.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<description>
text between the Tags, text between the Tags, text between the Tags.
</description>
<splitter/>
<description value="text in the attribute value" />
<description value="text in the attribute value with disabled true"
disabled="true" />
</window>
```

Figura 8.11a – Elementos Description.

Atributo	***Tipo***	***Descrição***
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. ● **start** O texto será encolhido para seu lado esquerdo. ● **end** O texto será encolhido para seu lado direito. ● **left** O texto será encolhido para seu lado esquerdo. ● **Right** O texto será encolhido para seu lado direito. ● **center** O texto será encolhido para ambos os lados. ● **none** O texto não será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).
Disabled	Boolean	Indica se elemento está desabilitada ou não. Quando verdadeiro mostra o elemento em uma cor mais transparente que os demais.
Value	Caracter	Descrição ou texto do elemento.

Propriedades e Métodos	***Tipo***	***Descrição***
Accessible	nslAccessible	Retorna o objeto de acessibilidade para o description
Crop	Alfa-numérico	Ajusta o valor do atributo crop.
Disabled	Booleano	Ajusta o valor do atributo disabled.
Value	Caracter	Ajusta o valor do atributo value.

Estilo Classe	***Descrição***
Header	Utilizada para deixar o texto em negrito.
Indent	Mostra uma indentação do texto.

Estilo Classe	*Descrição*
Monospace	Mostra o texto com espaçamento único entre os caracteres.
Plain	Utilizado para que seja desconsiderada qualquer margem do texto.
Small-margin	Mostra uma pequena margem no texto.

8.12 Menu.

O elemento menu trabalha em conjunto com outros elementos como a barra de menu (menubar) , o menu em si, o chamada do menu(menupopup) e uma lista de itens (rule) e os itens do menu (menuitem). sendo este elemento também responsáveis pela criação de submenus. Na **Listagem 8.12a** podemos observar um simples menu com um menupopup e com três subitens descritos pelo elemento menuitem, sendo o resultado apresentado na **Figura 8.12a**.

Listagem 8.12a – Elemento Menu.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul" >
<menu id="menu" label="Menu" accesskey="M" >
        <menupopup id="menupopup" >
                <menuitem label="Item 1" acceltext="ALT+1" />
                <menuitem label="Item 2" acceltext="ALT+2"/>
                <menuitem label="Item 3 (disabled)" disabled="true"
acceltext="ALT+D"/>
        </menupopup>
</menu>
</window>
```

Figura 8.12a – Elemento Menu com Elementos Menuitem.

Atributo	***Tipo***	***Descrição***
Acceltext	Caracter	Texto para indicar tecla de atalho do menu ao lado de descrição do mesmo.Exemplo: Menu 1 ALT+1
Accesskey	Caracter	Indica qual será a letra para utilizada como tecla de atalho do elemento, esta letra será sublinhada.Quando o usuário precionar ALT (A tecla de acessibilidade pode variar de acordo com a plataforma) e a chave de acesso a função do elemento será ativada.
Allowevents	boolean	Quando verdadeiro os eventos serão aplicados também aos submenus relacionados.

Atributo	***Tipo***	***Descrição***
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. • **start** O texto será encolhido para seu lado esquerdo. • **end** O texto será encolhido para seu lado direito. • **left** O texto será encolhido para seu lado esquerdo. • **Right** O texto será encolhido para seu lado direito. • **center** O texto será encolhido para ambos os lados. • **none** O texto não será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).
Disabled	Boolean	Indica se elemento está desabilitada ou não. Quando verdadeiro mostra o elemento em uma cor mais transparente que os demais.
Key	Chave do elemento	Identificação chave do elemento para ser usado como atalho em comandos do menu.
Label	Caracter	Descrição que aparecerá no menu.
Menuactive	Boolean	Este atributo é ajustado em um item do menu quando o mesmo está selecionado.
Open	Boolean	Este atributo e ajustado quando o menu é aberto.
Sizetopopup	Caracter	Indica como a largura do menu e a largura do popup sào determinadas.
Value	Caracter	Descrição associada ao elemento menu.

Propriedades e Métodos	***Tipo***	***Descrição***
Value	Caracter	Ajusta o valor do atributo value.

8.13 Menubar.

Este elemento cria uma barra especializada para trabalhar com elementos menu, nesta barra podemos criar uma estrutura de menus organizados para os aplicativos. Na **Listagem 8.13a** demonstramos como criar um menubar com dois elementos menu que fim contém elementos menuitem, que podem ser visualizados na **Figura 8.13a**.

Listagem 8.13a – Elemento Menubar.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul" >
<menu id="menu" label="Menu" accesskey="M" >
        <menupopup id="menupopup" >
                <menuitem label="Item 1" acceltext="ALT+1" />
                <menuitem label="Item 2" acceltext="ALT+2"/>
                <menuitem label="Item 3 (disabled)" disabled="true"
acceltext="ALT+D"/>
        </menupopup>
</menu>
</window>
```

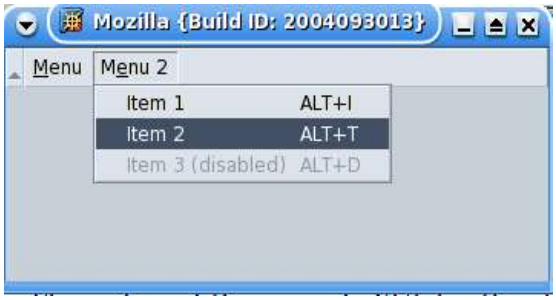


Figura 8.13a – Elemento Menubar.

Atributo	***Tipo***	***Descrição***
Accessible	NsIAccessible	Retorna o objeto da acessibilidade para o elemento.
Grippyhidden	Boolean	Quando verdadeiro o grippy será escondido. (Não Válido para Firefox)
Statusbar	Elemento statusbar	Ajusta à identificação de um elemento statusbar (somente Firefox).

Propriedades e Métodos	***Tipo***	***Descrição***
Statusbar	Elemento statusbar	Ajusta o valor do atributo statusbar.

8.14 Menuitem.

Este elemento apresenta os itens contidos dentro de um elemento menu, cada item deste acrescentado dentro de um elemento menu cria uma linha no menu. Este elemento pode trabalhar também com as mesmas características dos elementos checkebox e radio. Na **Listagem 8.14a** temos diversos elementos menuitem para exemplificar suas propriedades e funcionalidades , sendo que na **Figura 8.14a** apresentamos os elementos gerados.

Listagem 8.14a – Elementos Menuitem.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul" >
<menu id="menu" label="Menu" accesskey="M" >
        <menupopup id="menupopup" >
                <menuitem label="MenuItem 1" acceltext="ALT+1" />
                <menuitem label="MenuItem 2 (disabled)" disabled="true"
acceltext="ALT+D"/>
                <menuitem label="MenuItem 3 (Checkbox)" type="checkbox"
autocheck="true" checked="true" acceltext="ALT+3"/>
                <menuitem label="MenuItem 4 (Autocheck)" type="checkbox"
autocheck="true" acceltext="ALT+A"/>
                <menuitem label="MenuItem 5 (Checkbox)" type="checkbox"
autocheck="true" checked="true" acceltext="ALT+C"/>
                <menuitem class="menuitem-iconic" label="MenuItem 6 (image)"
image="save.gif" key="ALT+I"/>
        </menupopup>
</menu>
</window>
```

Figura 8.14a – Elementos Menuitem.

Atributo	***Tipo***	***Descrição***
Acceltext	Caracter	Texto para indicar tecla de atalho do menu ao lado de descrição do mesmo.Exemplo: Menu 1 ALT+1
Accessible	nslaccessible	Retorna o objeto da acessibilidade para o elemento.

Atributo	***Tipo***	***Descrição***
Accesskey	Caracter	Indica qual será a letra para utilizada como tecla de atalho do elemento, esta letra será sublinhada. Quando o usuário pressionar ALT (A tecla de acessibilidade pode variar de acordo com a plataforma) e a chave de acesso a função do elemento será ativada.
Allowevents	boolean	Quando verdadeiro os eventos serão aplicados também aos submenus relacionados.
Autocheck	Boolean	Permite que o item trabalhe como um checkebox, sendo que ao clicar o usuário setará este atributo automaticamente.
Checked	Boolean	Indica se o menuitem está selecionado ou não quando se está trabalhando com o atributo type igual a checkbox ou radio.
Command	Elemento	Ajusta a identificação do elemento ao comando a ser executado.
Crop	Pré-definido	Se o label do elemento for maior que o espaço dado pelo elemento, o texto estará encolhido pelo lado especificado pelo atributo crop. Será apresentado ... (três pontos) no lugar do texto encolhido. Se o sentido da caixa for invertido, o cropping estará invertido. ● start O texto será encolhido para seu lado esquerdo. ● end O texto será encolhido para seu lado direito. ● left **(desaprovado)** o texto será encolhido para seu lado esquerdo. ● right **(desaprovado)** o texto será encolhido para seu lado direito. ● center O texto será encolhido para ambos os lados. ● none O texto não será encolhido usando os (...). Porém, o texto estará cortado simplesmente de acordo com o tamanho do botão. O lado depende do alinhamento do texto do CSS (caso exista).
Disabled	booleano	Indica se elemento está desabilitada ou não. geralmente mostra o elemento em uma cor mais transparente que os demais.

Atributo	***Tipo***	***Descrição***
Image	URL imagem	URL para imagem que aparecerá no menuitem (para que a imagem apareça deve ser setada a classe **'menuitem-iconic'** na classe).
Key	Chave do elemento	Identificação chave do elemento para ser usado como atalho em comandos do menu.
Label	Caracter	Descrição que aparecerá no menu.
Name	Caracter	Utilizado para agrupar elementos com o mesmo nome formando um grupo, que irá trabalhar juntamente com o elemento item radio permitindo que apenas um menuitem do grupo seja selecionado.
Selected	Boolean	Indica que o menuitem está selecionado em um menulist.
Type	Pré-definido	Utilizado para indicar se o menuitem tem característica de Checkbox ou radio. • **Checkbox** – O menuitem trabalha como um checkbox. • **Radio** - O menuitem trabalha como um elemento radio.
Validate	Pré-definido	Este atributo indica se serão carregadas a imagens do cache ou não. • **always** - A imagem é verificada sempre para ver se deve ser recarregada. • **Never** - A imagem será carregada do cache se possível.
Value	Caracter	Descrição associada ao elemento menu.

Propriedades e Métodos	***Tipo***	***Descrição***
Label	Caracter	Ajusta o valor do atributo Label.
Value	Caracter	Ajusta o valor do atributo value.

8.15 Progressmeter.

Um medidor de processo tem a função de indicar o tempo restante para o término de execução de uma determinada tarefa, em processos mais longos, este elemento tem duas modalidades:

Determinado.

São usados quando o tempo de execução é conhecido, pode ser encontrado com algum calculo ou algoritmo, sendo o medidor de processo uma processo que vai de 0 a 100%, demonstrado no medidor, veja um medidor de processo em XUL na **Figura 8.15a**.

Figura 8.15a – Medidor de processo determinado.

Indeterminado

Os Medidores de processos são usados para tarefas que não existe como saber o tempo de sua execução, por isso são chamados de indeterminados, veja na **Figura 8.15b** a medidor de processo indeterminado que tem uma animação de uma espécie de cones giratórios

Figura 8.15b – Medidor de processo determinado

Na **Listagem 8.15a** temos um exemplo com um progressmeter determinado e um indeterminado que resultam na **Figura 8.15c**.

Listagem 8.16a – Elementos progressmeter.

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="progressmeter" title="Progressmeter" orient="horizontal"
    xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<vbox flex="1">
  <description>
    Progressmeters determined.
  </description>
  <hbox>
    <progressmeter id="pm1" value="25%" style="margin: 4px;"/>
    <spacer flex="1"/>
  </hbox>
  <description>
   Progressmeter undetermined.
  </description>
  <hbox>
    <progressmeter id="pm5" style="margin: 4px;" mode="undetermined"/>
  </hbox>
</vbox>
</window>
```

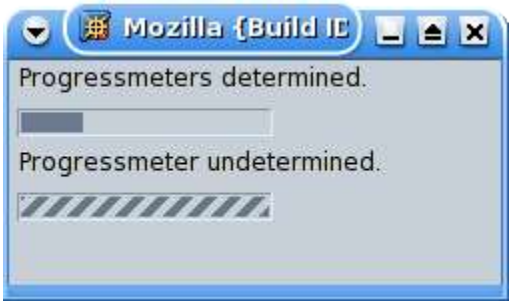


Figura 8.15c – Elementos progressmeter.

Atributo	*Tipo*	*Descrição*
Mode	Pré-definido	O tipo do medidor de progresso. Se ele estiver como determined, o medidor de progresso é um medidor de progresso determinado, preenchendo quando a tarefa for concluída. Se ele estiver como **undetermined**, o medidor de progresso será indeterminado, onde você não sabe a quantidade de tempo. O valor **determined** é o padrão se você não especificar este atributo.

Atributo	*Tipo*	*Descrição*
value	integer	O valor atual do medidor de progresso. Você só pode usar esse valor para um medidor de progresso que seja determinado. O valor pode ser definido para uma porcentagem de 0% a 100%. O valor pode ser mudado por um script quando a tarefa terminar.

Propriedades e Métodos	*Tipo*	*Descrição*
Accessible	nslAccessible	Retorna o objeto de acessibilidade para o progressmeter.
mode	Alfa-numerico	Ajusta o valor do atributo da modalidade.
value	Alfa-numerico	Começa a ajustar o valor do atributo do valor.

9. Capítulo 9 - RESULTADOS OBTIDOS.

9.1 XULWareHouse.

O XULWareHouse Project é um Armazém de Elementos e componentes XUL para Interfaces Gráficas para facilitar e agilizar o desenvolvimento de aplicações com XUL e também ser um repositório para compartilhamento de elementos personalizados. Este projeto pretende facilitar e reaproveitar todos os esforços feitos para a construção de componentes XUL, sendo que em diversos casos em softwares diferentes temos a necessidade de fazer um mesmo componente.

Um Exemplo disso poderia ser o caso de eu ter de criar uma calculadora em XUL, uma uma tela de confirmação do Usuários com os botões SIM ou Não ou uma componente para acesso a base de dados,etc. Neste casos ou em casos similares podemos ser aproveitados os códigos disponíveis no XULWareHouse, ou até mesmo quando tiver uma necessidade básica de saber como deve ser escrito um componente padrão do XUL, como um botão ou uma simples janela, tudo isso pretende estar disponível no projeto. Com a visualização teste da tecnologia de uma maneira rápida pode ser aberto um caminho de o acesso a novos desenvolvedores de XUL.

Este projeto foi resultado desta monografia, que teve a necessidade de codificar diversos

componentes que foram escritos para testar a tecnologia e podem agora ser aproveitados, tanto para didática de novos interessados como para troca de componentes de desenvolvedores XUL . O projeto será disponibilizado em um sítio da internet e terá a possibilidade para que os usuários baixem componentes , melhorem componentes e entrem com mais componentes, se tornando uma *"Base de Componentes e aplicações básicas XUL"* .

O sítio temporário do projeto está em <www.ambientelivre.com.br/xulwarehouse/index.html> e muito ainda pode ser melhorado e incrementado.

9.2 Divulgação da Tecnologia.

Este trabalho será disponibilizado para download no site do projeto Xul:Brasil, após sua conclusão, sendo este atualmente o site de maior referência sobre está tecnologia no Brasil, para divulgar e auxiliar novos adeptos da tecnologia, o trabalho também será levado em um formado te palestra e outro de artigo para ser divulgado em congressos de tecnologia livres, fazendo com que está monografia tenha o seu foco de pesquisa e desenvolvimento compartilhando com o maior numero de pessoas a possibilidade de acesso a mais uma tecnologia com padrões abertos.

10. Capítulo 10 - CONCLUSÃO.

Após uma ampla pesquisa da tecnologia XUL e de tecnologias relacionadas, posso concluir que a tecnologia XUL é uma ótima opção para a criação de interfaces gráficas para usuários portáteis. O fato da tecnologia ser totalmente open source é um grande beneficio para quem pretende usufruir desta tecnologia, a possibilidade de pode executar as aplicações na web aumentam ainda mais a sua portabilidade, sendo que a sua semelhanças com tecnologias XML e HTML e a possibilidade de trabalhar com scripts JavaScript uma facilidade para profissionais que já trabalham com o desenvolvimento web utilizar deste conhecimento para a criação de interfaces em XUL. A tecnologia XUL já está madura, e pode ser muito mais utilizada na criação de aplicativos do que está sendo usada atualmente, mais iniciativas devem ser criadas para divulgar o XUL que é muito pouco conhecido entre desenvolvedores de softwares. O XUL será uma das principais tecnologias concorrentes das novas tecnologias baseadas em XML da Microsoft , o XAML.

11. REFERÊNCIAS.

11.1 Livros.

Desenvolvendo Sites com XML, Mílton Bender Jr, Serial Sistemas e Consultoria Ltda.

Essencial XUL Programming, Autores: Vaughn Bullard,Kevin T. Smith,Michal C. Daconta - editora Jonh Wiley disponível em <www.xulbook.com> acessado em 02 mai. de 2005.

Creating Applications with Mozilla, Autores : David Boswell, Brian King, Ian Oeschger, Pete Collins e Eric Murphy – Editora O'Reilly & Associates, Inc. Disponível em <http://books.mozdev.org/html/> acessado em 26 de jun. 2005.

11.2 Congressos.

2° Congresso Internacional de Software Livre **CONISLI** – Palestra Tecnologia XUL, Gustavo Araújo Bittencourt e Régis Fernandes Gontijo.

11.3 Internet.

Xul:Brasil . Disponível em: <http://twiki.im.ufba.br/bin/view/XUL/WebHome> . Acesso em 10 jan. 2005.

XULPlanet Disponível em: <http://www.xulplanet.com/>. Acesso em 10 jan. 2005.

Tableless Design CSS XHTML HTML Web Standards. O que é CSS. Disponível em: <http://www.tableless.com.br/wiki/wiki.asp?page=O+que+%E9+CSS%3F> . Acesso em 31 abr. 2005.

INTERFACES básicas em XUL. **iMasters**. Disponível em: <http://www.imasters.com.br/artigo.php?cn=3077&cc=214> . Acesso em 21 abr. 2005.

ELEMENTOS em em XUL. **iMasters**. Disponível em: <http://www.imasters.com.br/artigo.php?cn=3142&cc=214> . Acesso em 20 abr. 2005.

About the Mozilla Foundation. **Mozilla Foundation**. Disponível em : <http://www.mozilla.org/foundation/> Acesso em 21 abr. 2005.

101 coisas que o Mozilla pode fazer e o IE não. **Mozilla Brasil** Disponível em: <http://www.mozilla.org.br/saite/conteudo.php?inc=conteudo/texto&tx=23&id_aba=9> Acesso em 10 de mai. 2005.

Infinite XBL. **O'Relly Net** Disponível em: <http://www.oreillynet.com/pub/a/network/2000/08/18/magazine/infinite_xbl.html> Acesso em 10 de mai. 2005.

Resource Description Framework (RDF). **W3C** Disponível em : <http://www.w3.org/RDF/> Acesso em 22 de mai. 2005.

What is RDF? O'REILLY **XML.com.** disponível em : <http://www.xml.com/pub/a/2001/01/24/rdf.html> Acesso em 22 de mai. 2005.

Wikipédia. Disponível em : <http://pt.wikipedia.org/> Acesso em 07 de jun. 2005.

Next Generation Applications in XUL: Microsoft Avalon XAML for the Rest of Us, **Open Source Convention**. Disponível em : <http://conferences.oreillynet.com/cs/os2004/view/e_sess/5326 > Acesso em 23 de jul. 2005.

Getting started with PyXPCOM. **IBM.** Disponível em : <http://www-128.ibm.com/developerworks/webservices/library/co-pyxp1.html> Acesso em 27 de jul. 2005.

Classpath, entendendo o que é e para que serve. **GUJ.** Disponível em :
<http://www.guj.com.br/java.tutorial.artigo.108.2.guj> Acesso em 27 de Jul. 2005.

The Open-source framework for building cross-plataform User Interface. **JXUL.** Disponível em :
<http://jxul.sourceforge.net/> Acesso em 28 de Jul. 2005.

XPCOM Component Viewer.**Cview**. Disponível em :
<http://www.hacksrus.com/~ginda/cview/> Acesso em 28 de Jul. 2005.

Mozdev Community. **Mozidev.org**. Disponível em :
<http://mozdev.org/> Acesso em 28 de Jul. 2005.

Adolescente cria Firefox, e inicia nova guerra dos Browsers. **Estadão**. Disponível em :
<http://www.linuxit.com.br/modules.php?name=News&file=article&sid=3892> Acesso em 28 de Jul. 2005.

XML User Interface Language (XUL). **Mozilla**. Disponível em :
<http://www.mozilla.org/projects/xul/> Acesso em 21 Jul. 2005.

Search web pages. **Google**. Disponível em : <http://www.google.com/mozilla/google.xul> Acesso em 01 Jun 2005.

Projeto Imperador. **Algoritimo** informática. Disponível em:
<http://www.algoritimo.com.br/imperador.jsp> Acesso em 21 Jul. 2005.

XUL Maker. Mozdev.org. Disponível em: <http://xulmaker.mozdev.org/> Acesso em 14 de mai. 2005

XPCOM Component Viewer. **Hacksrus**. Disponível em: <http://www.hacksrus.com/~ginda/cview/>

XUL – Interfaces de usuários portáteis com tecnologia XML.

Acesso em 12 mai. 2005.

XULWareHouse Project. **Ambiente Livre.** Disponível em: <www.ambientelivre.com.br/xulwarehouse/index.html> Acesso em 15 de Jul. 2005.

Mozilla Amazon Browser. **MAB**. Disponível em : <http://www.faser.net/mab/chrome/content/mab.xul> Acesso em 12 mai. 2005.

12. GLOSSÁRIO.

12.1 Termos.

Browser de Internet: Um navegador (também conhecido como **web browser** ou simplesmente **browser**) é um programa que habilita seus usuários a interagirem com documentos HTML,XML,XUL,etc, hospedados em um servidores web.

Javascript: É uma linguagem de programação criada pela Netscape.

Netscape Communications Corp: Empresa americana, criadora do Netscape Navigator que por sua vez após a abertira de seu código deu origem ao projeto Mozilla.

Parser : É um programa de computador (ou apenas um componente de um programa) que serve para analisar a estrutura gramatical de um input, manipulando os tokens, que são segmentos de texto ou símbolos que podem ser manipulados.

Tags: São estruturas de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim.

Widget: É um termo sem tradução que designa *componentes de interface gráfica com o usuário*

(GUI). Qualquer item de uma interface gráfica é chamada de widget, por exemplo: janelas, botões, menus e itens de menus, ícones, barras de rolagem, etc.

12.2 Abreviaturas.

CSS - Cascading Style Sheets.

GUI - Graphical User Interface

JAR – Java Archive.

URI - Unifor Resource Identifier.

XUL - XML User Interface Language.

XML - Extensible Markup Language.

W3C - World Wide Web Consortium.