



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

RODOLFO ANTUNES DIAS CINTRA

DESENVOLVIMENTO DE PORTAIS CORPORATIVOS

Assis
2010

DESENVOLVIMENTO DE PORTAIS CORPORATIVOS

RODOLFO ANTUNES DIAS CINTRA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito de Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Dr. Alex Sandro Romeo de Souza Poletto

Analisador (1): Fernando Cesar de Lima

ASSIS
2010

DEDICATÓRIA

Dedico este Trabalho a minha esposa, minha filha, ao meu orientador, a todos que me ajudaram direta e indiretamente nestes quatro anos.

AGRADECIMENTO

Ao Prof. Dr. Alex Sandro Romeo de Souza Poletto, pela compreensão e auxílio na orientação, transmitindo-me conhecimento de forma clara e objetiva, com muito profissionalismo e inteligência, tornando sua ajuda indispensável para a conclusão deste trabalho.

A minha esposa Bibiani, minha filha Giovanna pelo apoio, compreensão e por estarem presentes nos momentos mais difíceis.

Aos meu avós Daniel e Vilza por tornarem isso tudo possível.

RESUMO

Este trabalho tem por objetivo mostrar a tecnologia de portal utilizando a especificação JSR 168, bem como oferecer um modelo para a implementação de um portal, apoiado no Framework JetSpeed 2 com o Sistema Gerenciador de Banco de Dados MySQL 5.

Palavras-chaves: Portal, Portlet, WEB2.0

ABSTRACT

This paper aims to show the portal technology using the JSR 168 as well as offer a model for the implementation of a portal, supported by the Framework Jetspeed 2 with the System Manager Database MySQL 5.

Keywords: Portal, Portle, WEB2. 0

LISTA DE ILUSTRAÇÕES

Figura 1. Evolução do número de hosts conectados na Internet.....	14
Figura 2. Primeira página Web criada.	15
Figura 3. Arquitetura de um Portal.	22
Figura 4. Exemplo de uma página de porta.....	25
Figura 5. Várias instâncias do mesmo portlet em uma página.	26
Figura 6. Fluxo de criação de uma página de portal.	27
Figura 7. Anatomia de um portlet.	31
Figura 8. Proposta de Trabalho.....	33
Figura 9. Pagina de download do JetSpeed - 2.....	34
Figura 10. Tela de Boas Vindas da instalação do JetSpeed - 2.....	35
Figura 11. Tela com a licença do JetSpeed	36
Figura 12. Tela de seleção de opção	36
Figura 13. Tela de configuração do loção a ser instalado	37
Figura 14. Tela de configuração do loção a ser instalado.	37
Figura 15. Tela de seleção do tipo a ser instalado.	38
Figura 16. Tela de seleção dos componentes a ser instalado.....	38
Figura 17. Tela de seleção do SGBD.....	39
Figura 18. Tela de configuração da conexão com o SGBD.	40
Figura 19. Tela de seleção do driver jdbc do SGBD.	40
Figura 20. Tela de teste de conexão com o SGBD.	41
Figura 21. Tela de informação.....	41
Figura 22. Tela de inicialização de instalação.	42
Figura 23. Tela instalação completa.....	42
Figura 24. Tela inicial do portal instalado.	43

LISTA DE SIGLAS E ABREVIATURAS

SGBD	Sistema Gerenciador de Banco de Dados
SBD	Sistema de Banco de Dados
SOA	Arquitetura Orientada a Serviços
JSR	Java Specification Request
ECM	<i>Enterprise Content Management</i>

SUMÁRIO

1. INTRODUÇÃO	9
1.1 OBJETIVO.....	9
1.2 JUSTIFICATIVA	10
1.3 MOTIVAÇÕES.....	10
1.4 ESTRUTURA DO TRABALHO	10
2. EVOLUÇÃO DO USO DA INTERNET ATÉ A WEB 2.0.....	12
2.1. O INICIO.....	12
3. MASHUP E WEB 2.0	17
3.1 MASHUP	17
3.1.1 Aplicações em Cinco Minutos	17
3.2 WEB 2.0	19
4. PORTAIS	21
4.1 SEGURANÇA.....	22
4.2. CONTEÚDO.....	23
4.3 PORTLET.....	24
4.3.1 Contêineres de Portlet e Contêineres de Servlets.....	27
4.4 JSRs 168/286.....	29
4.5 JSF/Portlet Bridge	32
5. PROPOSTA DE TRABALHO	33
6. ESTUDO DE CASO	34
6.1 INSTALAÇÃO.....	34
7. CONSIDERAÇÕES FINAIS	44
7.1 CONTRIBUIÇÕES	44
7.2 LIMITAÇÕES.....	44
REFERÊNCIAS.....	45

1. INTRODUÇÃO

Por volta de 1995, a Internet realmente começou. Com isso, desde então o valor real tomou forma e é maior difundida a idéia de que as empresas pudessem ter suas informações *online*. Logo após, muitas empresas foram criadas com o objetivo de organizar toda essa informação. O termo "Portal" foi criado para descrever uma aplicação Web, criada para apresentar um grande volume de informações de forma concisa e centrada, tornando a Internet mais fácil de usar. Exemplos mais populares de portais de informação são o Yahoo.com, Uol.com.br, etc.

Desde então, as ferramentas de construção de portais no sentido não apenas de informações, mas na forma de concentração de aplicativos (sistemas) foram ignoradas e discriminadas por muitos programadores que as tinham como de uso exclusivo para Web Designers. Mas com o surgimento da SOA (arquitetura orientada a serviços) e a busca das empresas em fornecer soluções mais baratas e produtivas, esse tipo de ferramenta voltou com toda a força para fazer o que sempre fez: facilitar a vida de todos.

Este trabalho visa apresentar o conceito de portal utilizando a JSR 168, mostrando suas características. Para isto será utilizado o *Framework Opensource Jetsepeed 2* da fundação Apache, junto com o sistema de banco de dados MySQL da IBM ilustrando passo a passo como instalar e configurar um portal de aplicações.

1.1 OBJETIVO

O objetivo deste trabalho é apresentar as tecnologias para desenvolvimento de portais, que utilizam ou que sejam compatíveis com a especificação das JSRs 168/286 para portlets. Desenvolvendo por fim, um ambiente único e simplificado para acesso as informações e serviços de uma empresa através da web, usando o framework JetSpeed 2

1.2 JUSTIFICATIVA

Atualmente as empresas buscam soluções para integrar suas informações e serviços em um ambiente unificado, ou seja, de fácil acesso e interface agradável. Um portal usando as especificações JSRs 168/286 fornece aos usuários esta possibilidade.

1.3 MOTIVAÇÕES

Com o passar do tempo as empresas acumulam aplicações para facilitar e agilizar suas tarefas diárias, porém grande parte destas aplicações são independentes, gerando com isto diversos problemas, pois dificulta a vida dos usuários quando necessário a requisição de informações de mais de um local. Tendo a visão que um portal integra estas aplicações em um único ambiente, após uma análise no mercado, notasse grande defasagem de profissionais capacitados para uso desta tecnologia.

1.4 ESTRUTURA DO TRABALHO

O trabalho será dividido em sete capítulos, conforme descritos a seguir:

Capítulo 1 – Introdução

Resumo do trabalho e como o mesmo será montado.

Capítulo 2 – Evolução do uso da internet até a Web 2.0

Descrever um estudo sobre a evolução do uso internet mostrando sua evolução.

Capítulo 3 – *Mashup* e Web 2.0

Definição dos termos *mashup* e web2. 0.

Capítulo 4 – Definição de portais

Descrever portal e suas funcionalidades.

Capítulo 5 – Propostas de Trabalho

Descrever a proposta de trabalho

Capítulo 6 – Estudo de Caso/Experimento

Validar a Proposta de Trabalho - Desenvolvimento de um Portal Corporativo utilizando o framework Jetspeed 2

Capítulo 7 – Considerações Finais e Projeções Futuras

Será apresentada a conclusão do trabalho.

Referências utilizadas no desenvolvimento da pesquisa.

2. EVOLUÇÃO DO USO DA INTERNET ATÉ A WEB 2.0

2.1. O INICIO

No final de 1957, ocorreu um evento que iria mudar o mundo. A União Soviética lançou com sucesso o primeiro satélite na órbita da Terra. Chamado “*Sputnik 1*”, ele chocou o planeta, inclusive os Estados Unidos, que tinha seu próprio programa de lançamento de satélites, mas ainda não havia lançado.

Com isso o Departamento de Defesa dos Estados Unidos criou a ARPA (Agência de Projetos de Pesquisa Avançada), devido à necessidade de uma organização que pudesse pesquisar, desenvolver idéias e tecnologia avançada para além das necessidades identificadas atualmente. Talvez o seu mais famoso projeto foi a criação da Internet.

O “ARPANET” plano para esta rede de computadores foi apresentado em Outubro de 1967, e em Dezembro de 1969 a primeira rede de quatro computadores estava pronta e funcionando. O grande problema em criar uma rede era como conectar redes físicas separadas sem que as ligações aumentassem os recursos de rede para *links* constantes. A técnica que solucionou este problema é conhecida como troca de pacotes e envolve requisições de dados divididos em “pacotes”, que podem ser processados rapidamente sem bloquear a comunicação de outras partes, este princípio ainda é usado para Internet hoje.

Com isso houve o surgimento de várias outras redes que usufruam desta técnica de troca de pacotes, por exemplo, X.25 (desenvolvida pela União Internacional de Telecomunicações) formando as bases da primeira rede universitária do Reino Unido JANET (Conjunto de rede acadêmica), permitindo todas as universidades britânicas a enviarem e receberem arquivos e e-mails, e a rede pública americana *CompuServe* (um empreendimento comercial permitindo pequenas empresas e indivíduos a acessarem recursos computacionais por um tempo compartilhado, e depois acesso à Internet). Estas redes apesar de terem muitas conexões, foram mais redes privadas que a Internet de hoje.

O surgimento de diferentes protocolos de rede logo se tornou um problema, para fazer redes separadas se comunicarem. Havia uma solução à vista “*Internet Transmission Control Program*” (“Programa de Controle de Transmissão Entre Redes”).

Esta especificação reduziu o papel da rede e passou a responsabilidade de manter a integridade da transmissão para o servidor. O resultado final foi que ela tornou possível acessar com facilidade quase todas as redes simultaneamente. A ARPA financiou o desenvolvimento do software, e em 1977 foi conduzida uma demonstração de uma comunicação entre três redes diferentes. Em 1981, a especificação foi finalizada, publicada e adotada; e em 1982 as conexões da ARPANET para fora dos EUA foram convertidas para usar o novo protocolo “TCP/IP”, foi o início da Internet.

2.2 O NASCIMENTO DA *WORLD WIDE WEB*

A Internet foi criada oficialmente, em 1991, por um cientista do CERN, chamado Tim Berners Lee. Numa iniciativa para simplificar a navegação da Inter-Network (até então restrita somente aos Campus Universitários e, obviamente, sistemas dos governos), Tim Berners-Lee propôs em 1989 um projeto de hipertexto que permitia as pessoas trabalharem em conjunto, combinando o seu conhecimento numa rede de documentos.

Foi este projeto que ficou conhecido como *World Wide Web* (WWW), ou Internet, como hoje é conhecida. Funcionou primeiramente dentro do CERN para em 1991 ganhar o mundo. Depois do seu lançamento foram criados diversos órgãos para padronizar e regulamentar a Internet, sendo o mais conhecido e respeitado o *World Wide Web Consortium*, ou W3C.

Foram padronizados o uso do HTML (sigla para *HyperText Markup Language*), CSS, Javascript, dentre outros padrões que hoje regem a apresentação dos sites.

Para que todo esse conteúdo fosse exibido também foi necessária a criação de navegadores Web. O primeiro navegador gráfico foi o *NCSA Mosaic*, que não exibia

nem imagens, lançado em 1993. Logo essa primeira versão foi atualizada. O seu criador logo se demitiu da NCSA e criou a Netscape, lançando o Navigator, que impulsionou e revolucionou a forma de exibir as páginas da época. Logo depois, a Microsoft comprou o Internet Explorer da *Spyglass* para competir nesse mercado.

De acordo com o instituto ISC (*Internet Systems Consortium*), em janeiro de 1993 havia cerca de um 1,3 milhão de dispositivos conectados na rede e este número cresceu de ano em ano:

- 2,2 em 1994;
- 4,9 em 1995;
- 9,5 em 1996;
- 16,1 em 1997 e assim por diante até chegar nos 542 milhões de *hosts* em janeiro de 2008, conforme mostra a Figura 1.

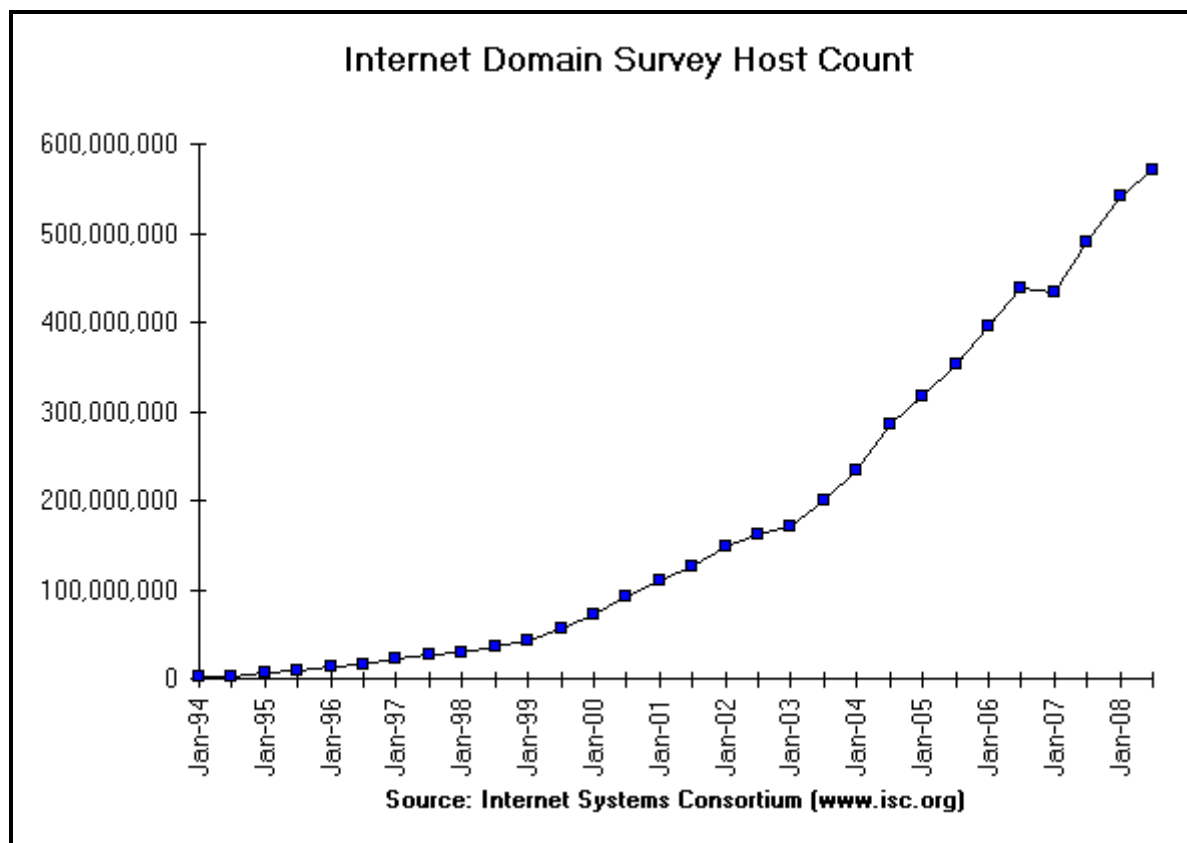


Figura 1. Evolução do número de hosts conectados na Internet.

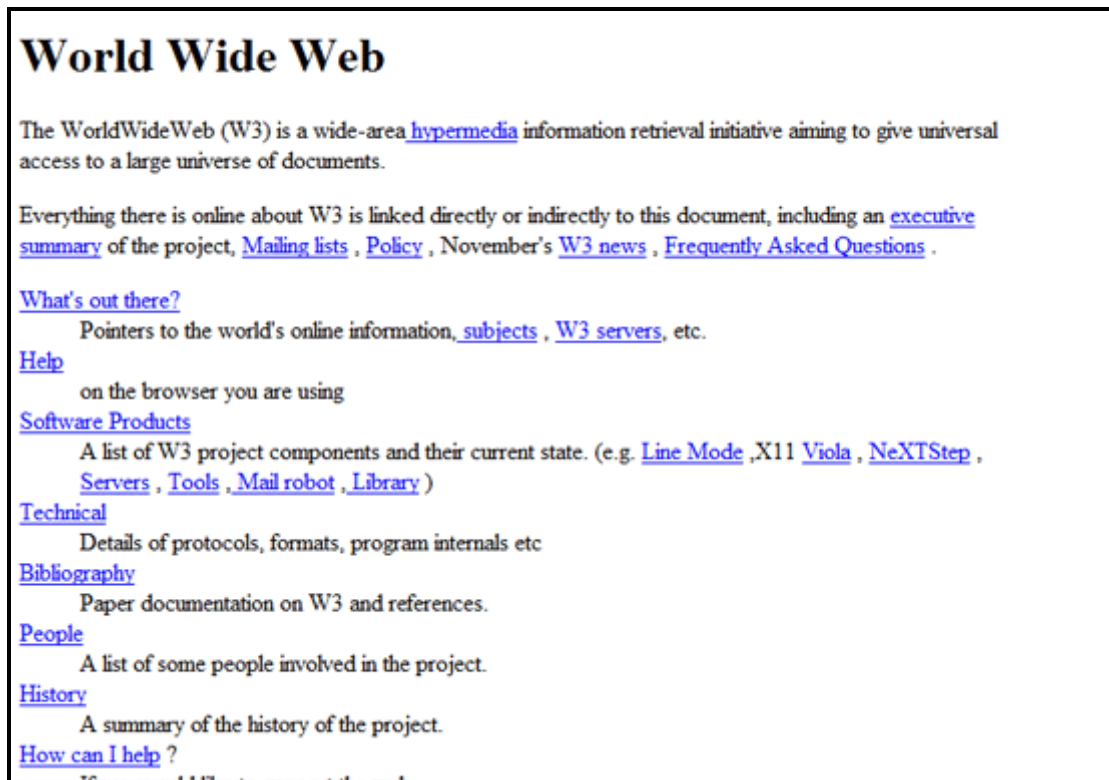


Figura 2. Primeira página Web criada.

A primeira página criada (Figura 2) originalmente estava no endereço <http://nxoc01.cern.ch/hypertext/WWW/TheProject.html>, mas este link já saiu do ar há muito tempo, atualmente pode ser vista no endereço <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html>.

2.3 A EVOLUÇÃO DA INTERNET

Com toda essa popularidade, o crescimento da Internet foi estrondoso. A chamada Web 1.0, que foi popularizada entre 1994 e 2004, com suas tabelas, imagens, a disputa dos browsers, a bolha da Internet, o motivo de noites e noites em claro dos

programadores e Web Designers, etc. Mas essa Internet inicial, com todas as suas dificuldades, abriu um mundo de possibilidades para todos.

No início do nosso século, houve uma revolução na forma de se criar os Websites. Tanto na parte gráfica, eliminando as temidas tabelas, quando na parte de programação, com novos conceitos como Ajax, *Webstandards*, CSS, etc. Com todo esse movimento de padronização também se desenvolveu o conceito de romper as barreiras do desktop, em que suas aplicações não fiquem mais no seu computador, mas na Internet, podendo serem acessadas de qualquer lugar a qualquer momento. Estava criado o conceito de Web 2.0.

Segundo Tim O'Reilly (2009) Web 2.0 é a mudança para uma Internet como plataforma, e um entendimento das regras para obter sucesso nesta nova plataforma. Entre outras, a regra mais importante é desenvolver aplicativos que aproveitem os efeitos de rede para se tornarem melhores quanto mais são usados pelas pessoas, aproveitando a inteligência coletiva.

Com isso você já sabe: surgiram aplicativos de tipos diversos como: aplicativos para criação de documentos *online*, planilhas, apresentações, aplicativos para armazenamento e gerenciamento de fotos, vídeos, e muitos outros.

Outra faceta da Web 2.0 é o ambiente de colaboração que foi construído, Blogs, Wikis, sites sociais, sites profissionais que permitem que qualquer usuário faça a sua história. O usuário da Internet deixou de ser um agente passivo para ser um agente ativo da formação da Internet, do conteúdo gerado, de informações.

3. MASHUP E WEB 2.0

3.1 MASHUP

Antenadas com a Web 2.0, as empresas começam a mesclar aplicações. Imagine usar um tipo de mixagem semelhante ao das músicas do hip-hop para criar aplicações, misturando informações e serviços de dentro e de fora da empresa? Pois é exatamente essa a idéia por trás do conceito de *mashup* corporativo, que explora o uso de recursos da Web 2.0 e tecnologias como Ajax, Java e RSS.

O termo *mashup* deriva da prática do hip-hop de mixar trechos de música e vem sendo empregado por diversos sites na Internet, com o objetivo de combinar informações de várias fontes num único endereço. No ambiente corporativo, esse recurso traz uma visualização fácil e rápida dos dados espalhados pela empresa, e até fora dela, com informações vindas, por exemplo, de sites na Web.

Uma das empresas que vêm evangelizando esse conceito é a IBM, que lançou o QEDWiki, sua solução para *Enterprise Mashup*. “É uma tecnologia de framework que utiliza serviços Web e recursos *wiki* para ajudar a criar *mashups* que combinem serviços, ferramentas e informações externas em uma aplicação flexível e de baixo custo”, diz Rod Smith, vice-presidente da IBM para tecnologias emergentes.

3.1.1 Aplicações em Cinco Minutos

A idéia da IBM é aproveitar essa flexibilidade da Web 2.0 para permitir que os próprios usuários criem aplicativos para necessidades específicas, instantaneamente, sem precisar recorrer à equipe de TI da empresa — em geral, sobrecarregada.

Em 2005, a IBM utilizou os recursos da Web 2.0 na construção de um site destinado a ajudar as pessoas desalojadas pelo furacão Katrina a encontrar novos empregos.

Batizado de *Jobs4Recovery*. O site funciona como um portal de buscas que integra informações sobre oportunidades de emprego disponíveis em outros endereços na web — como Yahoo! HotJobs.com, Indeed.com e JobCentral.com. Ao digitar no campo de busca o tipo de emprego que deseja, o usuário recebe uma lista de opções, coletadas nos diversos serviços, com a indicação de sua localização no Google Maps. Smith afirma que, graças à facilidade das tecnologias da Web 2.0, os programadores conseguiram colocar esse portal no ar em poucos dias. Outro exemplo real de aplicação está na *National Association of Broadcasters* (NAB), associação internacional que reúne as emissoras de rádio e televisão. “Trabalhamos com a NAB para desenvolver *mashups* para a indústria de arte e entretenimento”, diz Smith. “Equipes de produção trabalham em colaboração em projetos específicos, em tempo real, usando recursos como Ajax, Atom e mensagens instantâneas. Com a solução da IBM é possível conectar toda a equipe de pós-produção de um filme — som, efeitos especiais, edição etc. — em uma aplicação que nos permite acompanhar a evolução do trabalho e os recursos usados, distribuir tarefas, gerenciar orçamentos e atualizar conteúdos.”

O *mashup* corporativo segue filosofia semelhante à do modelo SOA, em que as aplicações são quebradas em componentes de serviços, que, por sua vez, podem ser combinados e misturados com outros serviços de acordo com as necessidades do negócio. Ambos permitem a reutilização de informações e de serviços já disponíveis para a criação de novas aplicações sob medida para o usuário. Isso pode simplesmente mudar o modelo de desenvolvimento de software adotado até agora nas empresas.

Para o instituto Gartner, as aplicações compostas, criadas com base na combinação e na reutilização de informações, são um dos aspectos mais poderosos do SOA. Elas estão na base da estratégia da IBM e, também, de outros fornecedores de tecnologia. Entre outros exemplos, o Gartner cita a Microsoft, que incluiu no Office 2007 ferramentas para a criação de aplicações compostas. Uma das primeiras iniciativas nesse sentido foi o projeto Mendocino, desenvolvido com a SAP, que coloca o Office na base dos serviços corporativos criados em plataforma SAP.

3.2 WEB 2.0

O termo WEB 2.0 foi designado para marcar o atual momento de mudanças pelas quais a Internet está passando, na sua busca pela otimização do meio. A evolução dos softwares utilizados em função do comportamento do usuário na Rede vem proporcionando um aprimoramento cada vez maior do meio que, após sua euforia inicial, passou a uma fase de decadência, necessitando de uma nova estratégia de Marketing que a traga aos holofotes. Duas grandes conferências que contaram com empresas do mais alto porte na Internet, como Amazon, Google, Microsoft e Yahoo, dentre outras, ocorreram em outubro de 2004 e 2005 em São Francisco (EUA), para discutir as tendências da Internet.

O termo WEB 2.0 foi lançado por Tim O'Reilly, da O'Reilly Media, junto com Dale Dougherty, da MediaLive International, num primeiro encontro, em 2004, no qual se fixaram idéias de inaugurar uma fase da Web que permitisse mais liberdade ao usuário, que deixa de ser passivo e passa também a ter o papel de produzir, mixar e classificar o conteúdo.

Ao contrário do que pode-se pensar, as rupturas que ocorrem na transição WEB 1.0 para a 2.0, não estão no campo das tecnologias utilizadas, mas na forma de como se utilizam, sejam elas novas ou antigas. As rupturas não se limitam a recursos, mas à tendência de circulação de conteúdo proposta pelo ideário da nova Web.

Os gigantes da Internet participantes da *Web 2.0 Conference* têm sua parcela de responsabilidade sobre todas as mudanças, já que estão implementando e fazendo valer os ideais da Web 2.0, investindo nas aplicações e alavancando os números de acessos nos seus sites que ficam conectados por muito mais tempo. Estão colhendo o que plantaram de forma muito rápida. E atrás deles há outros grandes, médios e pequenos que, obviamente, querem obter o mesmo sucesso, utilizando a mesma fórmula, que faz a Internet renascer como local de partilha de conhecimento, informação e oportunidades, sob os ideais de democracia da informação e liberdade de expressão.

A Internet se torna um meio de comunicação mais acessível, em constante crescimento, e a Web 2.0 não se sustenta sem grande quantidade de usuários,

mantendo-se durante um tempo de acesso cada vez maior, numa relação em que o conteúdo dos sites e sua usabilidade são hoje muito mais importantes que a apresentação visual ou simples presença como um indicador de status.

Neste ambiente novo da Internet, o usuário é peça chave nas aplicações e desenvolvimentos para a grande rede. Ele atua diretamente no conteúdo e obriga aos grandes investidores do setor e desenvolvedores a criarem soluções que o tornem cada vez mais ativo e entusiasmado com este poder sobre essa mídia que mais cresce em popularidade em todo o mundo. Dessa forma, os fundamentos dessa transformação são aqui considerados e abordados a partir de três macro-tendências: os sites de relacionamento, os sites de conteúdo participativo e os *softwares online*, enfocando seu potencial de aproveitamento, bem como os desafios a serem trabalhados nas relações entre gestores, empreendedores e usuários.

4. PORTAIS

No caso de Portais Corporativos, imaginamos grandes sites (ou intranets) dentro de uma empresa que agrega vários tipos de conteúdo, meios de colaboração, conhecimentos, aplicativos transacionais e gráficos de toda sorte, todos concentrados em uma única interface.

Para atender a demanda de desenvolvimento para este tipo de portal, o JCP disponibilizou as especificações das JSRs 168/286 que definem o desenvolvimento de portlets.

Portlets são os componentes principais de um Portal, são componentes web reutilizáveis que fornecem uma visão a um sistema de informação ou uma aplicação, através de fragmentos de marcação (geralmente HTML) que podem ser agregados ou plugados em qualquer página de portal, desde que seja compatível com a especificação.

Para a execução dos Portlets, o servidor de aplicação precisa dispor de um ambiente de execução, chamado Contêiner de Portlets. Diversas empresas fornecem soluções para execução de portlets compatíveis com as JSRs, entre elas podemos destacar a Oracle, IBM, Sun e Apache.

É possível também desenvolver portlets utilizando o *framework JavaServer Faces*, porém, para o seu funcionamento em um ambiente de portal, é preciso utilizar uma biblioteca chamada *JSF/Portlet Bridge*, que faz uma ponte ou um proxy entre a aplicação JSF e o contêiner de Portlets.

Se buscarmos a definição de Portal no dicionário, na maioria deles veremos que ele é definido como “uma porta, portão ou entrada”. Quando falamos em Portais web, estamos nos referindo a sites especiais da web ou intranet, que são designados a agir como um gateway de acesso a outros sites.

Um portal agrega informações de múltiplas fontes e torna estas informações disponíveis para diversos usuários. Além de disponibilizar diversas fontes de informações, eles fornecem um guia de serviços que auxiliam os usuários a se direcionarem no meio de tantas informações na internet.

Mais especificamente, um portal não deve ser visto *somente* como gateway para outros sites, mas para todos os recursos acessíveis na rede, que envolvem intranets, extranets ou a própria Internet. Em outras palavras, um portal oferece acesso centralizado para aplicações e todo conteúdo relevante para as empresas/usuários finais. Na Figura 3 apresentamos a arquitetura de um portal, formado por um servidor web, um contêiner de Servlets/Portlets e as aplicações (portlets) para o portal

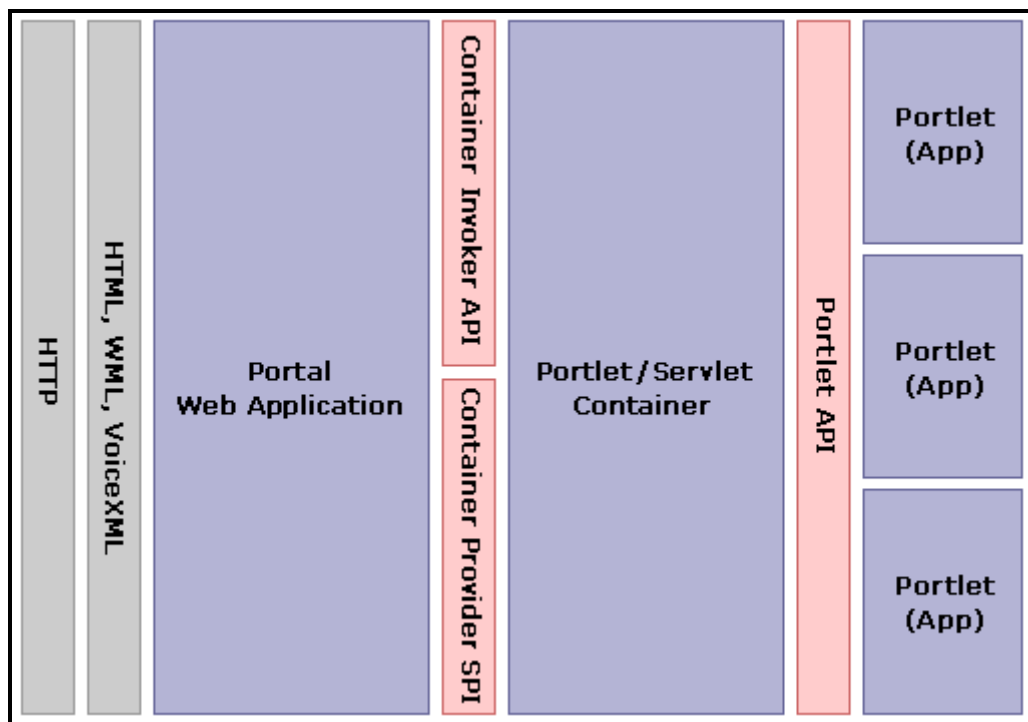


Figura 3. Arquitetura de um Portal.

4.1 SEGURANÇA

A etapa mais cansativa e torturante para um desenvolvedor é gastar seu tempo desenvolvendo módulos de segurança de uma aplicação, sendo que esses módulos geralmente são constituídos por telas de autenticação, tratamentos de usuários e grupos de usuário, utilização de repositórios LDAP, e em algumas vezes a integração com um autenticador para desfrutar do tão desejável *Single Sign On*.

Além de ter todas essas características, muitas vezes é necessário desenvolver soluções “*self-service*”, para que o administrador do ambiente de Portal tenha autonomia para gerenciar o repositório de usuários e dar permissões de uso para as funcionalidades da ferramenta.

Praticamente todas as soluções de portal possuem todo esse mecanismo pronto, possibilitando ao programador se dedicar somente no desenvolvimento das aplicações de automatização do negócio.

4.2. CONTEÚDO

No desenvolvimento completo de uma solução como uma “Intranet”, é comum ver que todo o tempo de desenvolvimento é gasto na criação de aplicações para gerenciamento de conteúdo, como funcionalidades de notícias, biblioteca de documentos, áreas institucionais e qualquer outra seção que necessite de um gerenciamento de informações por parte do usuário do sistema. Em um projeto deste tipo, também é comum que todas essas funcionalidades passem por uma governança editorial, e que inevitavelmente necessitem de um mecanismo de *workflow*. Essas são funcionalidades típicas nas ferramentas de portal, o que fez com que todos achassem que esse tipo de ferramenta só proporcionava esse benefício.

Há empresas que buscam outras soluções que vão além de um simples gerenciamento de conteúdo, indicando que elas devem ser mais robustas e possuir módulos de digitalização, transformação e busca de conteúdo. Esses sistemas, chamados ECM (*Enterprise Content Management*), muitas vezes dependem de uma ferramenta de portal para disponibilizar o conteúdo trabalhado. Este é só um exemplo de ferramenta que necessita de um portal como camada de visão. E a forma com a qual esse e outros tipos de ferramentas mostram suas funcionalidades é através dos chamados *portlets*.

4.3 PORTLET

A internet fornece um conjunto de informações quase que ilimitado para diversos tipos de dispositivos. Com o advento da web foi criado um universo de padrões e protocolos de comunicação, recursos, e uma linguagem de marcação utilizada para apresentação (HTML). Mas a web e seus protocolos, são projetados para atender ao conceito de uma página como um pedaço estático único de informação, apresentado em um navegador, como uma entidade completa e inalterável. Para visualizar outra página, o usuário tem que chamar outra página.

Até o momento, diversos esforços têm sido feitos para superar esta limitação com o uso extensivo de código, como JavaScript/Ajax, ou alguma solução DHTML para trazer ao usuário uma experiência similar e até mesmo superior ao uso de uma aplicação desktop.

Portlet é uma maneira de superar a natureza “tudo ou nada” de uma página HTML. Para defini-los pode-se dizer que os portlets constituem o núcleo dos serviços de um portal, onde uma ferramenta de portal utiliza os portlets como uma interface de apresentação plugável, ou seja, aplicações que podem ser adicionadas em qualquer página em um ambiente de portal, com o objetivo de fornecer qualquer tipo de informação na camada de apresentação.

O conteúdo gerado por um portlet é chamado de fragmento, que na verdade é um pedaço de marcação (ex: HTML, XHTML, etc.) aderente a certas regras, de forma que possamos agregar pedaços de vários fragmentos para gerar um documento.

A página de um portal é composta por um ou mais portlets, que são normalmente agregados com o conteúdo de outros portlets para formar uma página. O ciclo de vida de um portlet é gerenciado pelo contêiner de portlet, conforme veremos logo a seguir. Na Figura 4 é mostrado uma página de exemplo do portal open source Liferay, com a disposição de vários portlets.

The screenshot shows a web portal with a dark navigation bar at the top containing links for HOME, PRODUCTS, BLOGS, WIKI, FORUMS, ABOUT US, and JAVA MAGAZINE, along with an 'Add Page' button. Below the navigation bar, the page is divided into four portlets:

- DICTIONARY:** Features a search input field, a 'Dictionary' dropdown menu, and a 'Find' button.
- CALENDAR:** Displays the date 'Sunday, May 24, 2009'. It includes a calendar grid for the month of May, with the 24th highlighted. A table below the grid shows the days of the week and their corresponding numbers. To the right of the calendar are buttons for 'Add Event' and 'Permissions', and a table with columns 'Time', 'Title', and 'Type' containing the message 'There are no events on this day.' Below this is the text 'Showing 0 results.'
- RSS:** Shows two RSS feeds. The first is titled 'Innovation' and includes articles from 5/22/09 and 5/24/09. The second is titled 'NYT > Technology' and includes an article from 5/24/09 by John Markoff about artificial intelligence.
- GOOGLE MAPS:** Displays a map of a city area in São Paulo, Brazil. A red pin is placed on a street, with a callout box showing the address 'Rua Tuiuti, 600 - Tatuapé - São Paulo - SP'. The map includes various street names and landmarks.

Figura 4. Exemplo de uma página de porta.

Olhando atentamente o conteúdo do navegador na Figura 4, verifica-se que a página é formada por diferentes janelas. Tem-se uma janela para um dicionário, outra para um RSS, uma terceira para um calendário, e por último uma para o Google Maps. Cada uma destas janelas representa um portlet. Ao analisar o detalhe de cada janela, percebe-se que cada uma delas contém uma barra de título e alguns botões, incluindo os botões de maximizar e minimizar.

Na verdade, estas janelas são aplicações diferentes, desenvolvidas independentemente uma das outras. O programador desenvolve o portlet como uma aplicação web e empacota em um arquivo “.war”, e o administrador do portal efetua o deploy deste arquivo “.war” no servidor de portal e adiciona o portlet recém instalado em uma ou mais páginas do portal.

Pelo fato de você poder colocar o portlet em qualquer página, faz com que você possa reutilizá-lo a todo o momento no portal, até mesmo em uma mesma página. No caso da Figura 5, é apresentado uma página com várias instâncias de um mesmo portlet (Locadora).

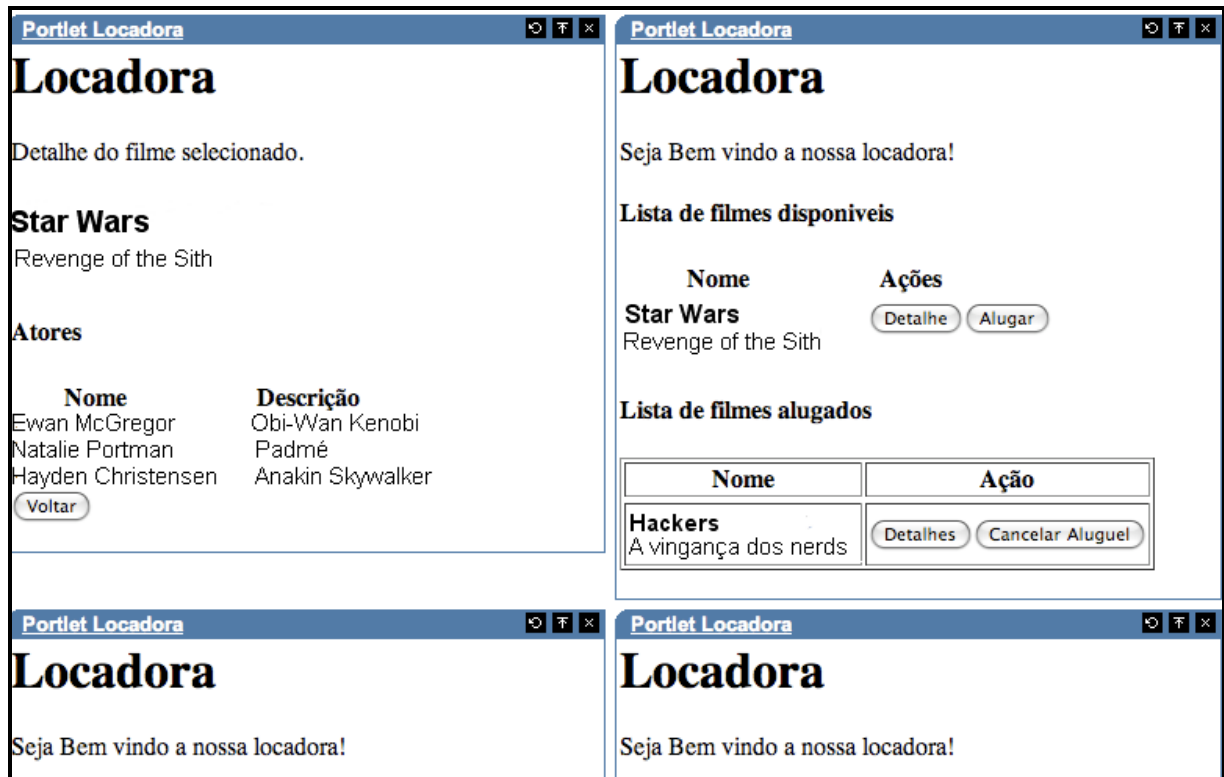


Figura 5. Várias instâncias do mesmo portlet em uma página.

Repare na figura 5 que se tem um mesmo portlet na página, mas com instâncias diferentes. A ferramenta de portal tem a responsabilidade de garantir sessões independentes da aplicação, mesmo estando todos na mesma página.

Portlets são executados em um contêiner de portlets. O contêiner fornece aos portlets o ambiente necessário para sua execução e gerenciam o seu ciclo de vida.

O contêiner de portlets recebe as requisições oriundas do portal para executar ações nos portlets em seu ambiente. É importante entender que contêiner de portlets não é responsável por agregar o conteúdo produzido pelo portlet que ele está hospedando, quem faz esta agregação é o próprio portal.

A Figura 6 apresenta como funciona a construção da página de um portal. Pode-se notar que os portlets rodam dentro do contêiner, que recebe o conteúdo gerado pelos portlets. Na seqüência, é mostrado que o contêiner retorna o conteúdo do portlet para o portal. Por último, o servidor de portal cria a página com o conteúdo gerado pelos portlets e envia para o dispositivo cliente, onde é renderizada (por exemplo, um navegador).

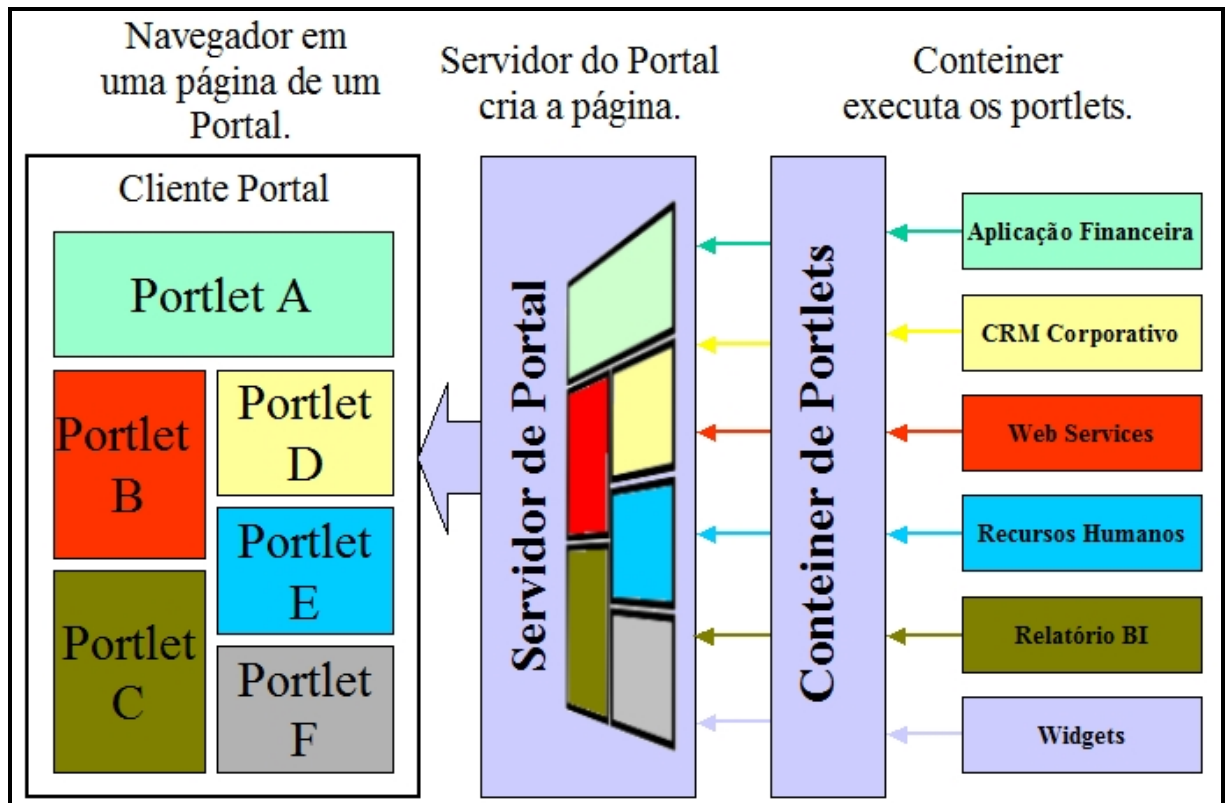


Figura 6. Fluxo de criação de uma página de portal.

4.3.1 Contêineres de Portlet e Contêineres de Servlets

Os portlets possuem várias similaridades com os servlets. Como ambos são componentes gerenciados por um contêiner especializado, os dois geram conteúdo dinâmico, por ambos interagirem com um cliente web através de request/response HTTP. Entretanto, os portlets não podem ser tratados como servlets por conta dos seguintes fatores:

Geram somente fragmentos de página no método de renderização, e não documentos completos;

- Podem somente ser invocados através de URLs construídas através da API de portlet;
- Clientes web interagem com os portlets através do sistema de portal;
- Possuem modos pré-definidos de portlet e estados de janela que indicam a função que o portlet está executando;
- Possuem um tratamento de requisição refinado para action, eventos, requisições de renderização e requisições a recursos;
- Não possuem acesso a certas funcionalidades fornecidas pelos servlets, como a URL de requisição do cliente ao Portal, ou atribuir o encoding de caracteres para a resposta ao cliente.

Por conta destas diferenças, o JCP decidiu criar para portlets um novo tipo de componente. Porém, ao formular a especificação de Portlets, o JCP procurou sempre que possível, potencializar o uso das funcionalidades fornecidas pela especificação de Servlets. Isto inclui *deployment*, *classloading*, aplicações web, gerenciamento de sessão e *request dispatching*. Por isso, diversos conceitos e partes da API de Portlets foram modelados a partir da API de Servlet.

Os objetos criados em uma aplicação de portal, como portlets, servlets e JSPs são empacotados como uma aplicação web comum (.war), e irão compartilhar o mesmo *classloader*, contexto de aplicação e sessão no ambiente de execução.

Request Dispatching: É o ato que efetuamos quando durante o processamento de um Servlet ou de um JSP necessitamos invocar outro servlet, JSP ou página HTML para um processamento adicional. Esta funcionalidade é fornecida pela API de Servlet mediante o uso da interface `javax.servlet.RequestDispatcher`.

4.4 JSRs 168/286

Duas JSRs, 168 e 286, atendem à criação de portlets. Em outubro de 2003 o JCP lançou a JSR-168, a primeira especificação que padroniza como os componentes devem ser desenvolvidos para os servidores de portal. Como as outras grandes JSRs, a JSR-168 possui o aval dos maiores fornecedores de portais. Esta especificação atende aos seguintes quesitos:

- O contrato de contêiner de portlet e o gerenciamento do ciclo de vida do portlet;
- A definição dos estados das janelas e os modos do portlet;
- Gerenciamento das preferências do Portlet;
- Informações do Usuário como nome, endereço, e-mail, que podem ser informadas no arquivo de *deployment descriptor* e que podem ser recuperadas por um objeto Map através da constante `USER_INFO` definida na interface *PortletRequest*;
- Empacotamento e Deployment;
- Segurança.

Mas, quando grandes fornecedores como a Apache, BEA, Borland, Oracle, IBM, SAP entre outros que fazem parte do *expert group* da JSR-168 começaram a criar as suas soluções de Portal, começaram a surgir também alguns problemas relacionados à limitação da especificação que antes não haviam sido detectados.

Conforme a adoção por soluções de portal baseadas na JSR 168 começou a aumentar, naturalmente o público consumidor destas soluções e o próprio negócio, passaram a demandar cada vez mais funcionalidades, que não haviam sido contempladas, fazendo com que soluções proprietárias fossem criadas pelos

fornecedores. Entre as limitações da primeira versão da especificação de Portlets (JSR-168), pode-se citar:

- Falta de filtros para portlets (similar aos Filtros nos Servlets);
- Ausência de funcionalidades para a intercomunicação entre os portlets;
- Limitação do Escopo de Sessão do Portlet, onde a sessão do Portlet na JSR 168 se restringe apenas ao ID individual da aplicação e não da sessão do usuário do Portal.

Essas limitações culminaram no lançamento da JSR 286, que define a segunda versão da API de Portlets. A JSR 286 incluiu novas funcionalidades como tratamento de eventos, envio e recebimento de parâmetros (permitindo a passagem de parâmetros entre os portlets), inclusão de filtros, suporte a Ajax e a disponibilidade de recursos do servidor, com o uso da interface *ResourceServingPortlet*.

Contrato do Contêiner e Gerenciamento do ciclo de vida do Portlet. A especificação da JSR-286 define um contrato para o contêiner de portlets. O contrato define quais métodos o contêiner deve chamar durante o ciclo de vida do portlet mediante o uso da interface *javax.portlet.Portlet*. O desenvolvedor pode implementar estes métodos para fornecer a funcionalidade desejada. Dessa forma, todo portlet tem que implementar esta interface, seja implementando-o diretamente ou estendendo uma classe existente que a implemente.

A especificação da JSR-286 ainda fornece interfaces opcionais para controle do ciclo de vida que o Portlet pode implementar como *EventPortlet*, que permite ao Portlet reagir às ações ou mudanças de estado oriundas da interação do usuário com o portlet ou ações originadas por outros portlets e *Portlet Modes* e *Window States* a interface *ResourceServingPortlet*, utilizada para disponibilizar recursos do servidor no Portlet.

Para cada portlet existem dois itens de estado gerenciado, que são o modo do portlet (*portlet mode*) e o estado da janela (*window state*). Cada portlet possui um “portlet mode” corrente que indica a função que o portlet está realizando.

Os *modes* (ou modos) definidos na especificação são VIEW, EDIT e HELP. Estes modos são utilizados pelo método de renderização *render()*, conforme vimos na Tabela 1, para decidir qual método de visualização deve ser chamado.

A classe *PortletMode* define constantes para estes *portlet modes*. A disponibilidade dos modos para o portlet pode ser restrito as roles (papéis) específicos de usuários do portal. Por exemplo, um usuário anônimo pode acessar o portlet apenas nos modos VIEW e HELP, enquanto que a role EDIT pode ser utilizada somente por usuários autenticados.

Outro tipo de estado gerenciado, o estado da janela (*window state*), apresenta a quantidade de informações que será disponibilizada para o portlet. A especificação define três tipos de estados para cada portlet, que são: normal, maximizado e minimizado. A classe *WindowState* possui constantes para estes estados da janela.

Para um maior entendimento, a Figura 5 apresenta estes conceitos de portlet mode e *window state* aplicados a um portlet.

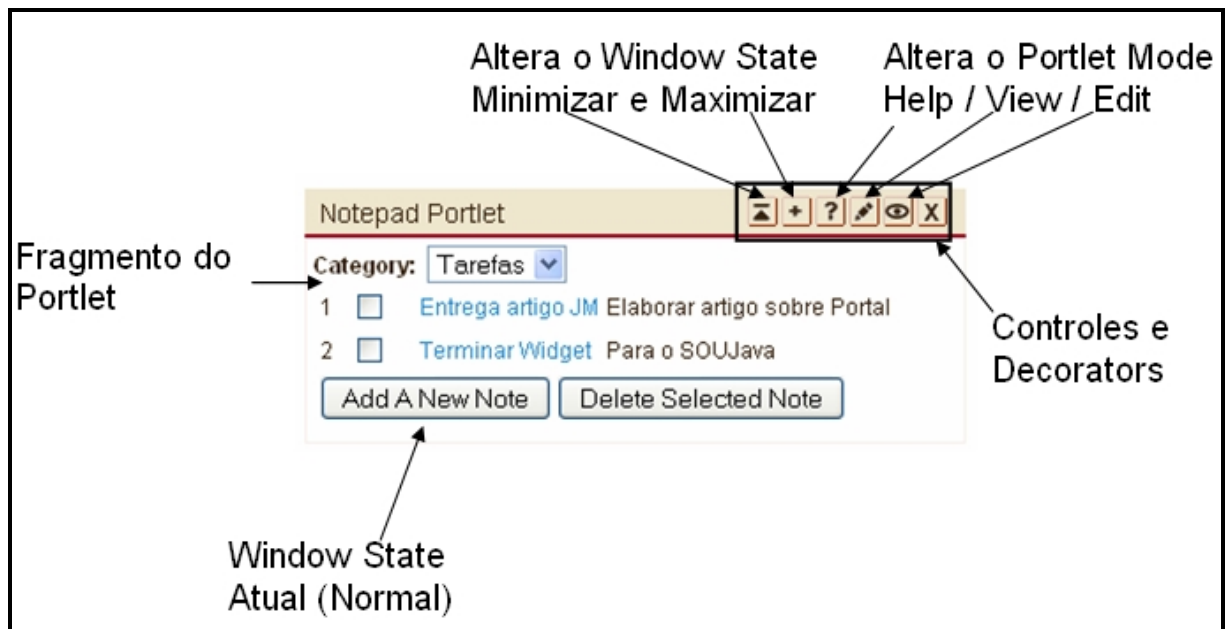


Figura 7. Anatomia de um portlet.

Opcionalmente o desenvolvedor pode especificar tipos customizados de modo e estado de janela, alterando o arquivo portlet.xml com uso da tag custom-window-state para estados de janela e custom-portlet-mode para modos.

4.5 JSF/Portlet Bridge

Como parte do projeto *OpenPortal* da Sun, foi criado o projeto *JSF/Portlet Bridge*, que tem como objetivo fornecer uma biblioteca de integração que permita que aplicações desenvolvidas com JSF possam ser executadas em um ambiente de Portlet.

Para atender à especificação de portlets, o JCP disponibilizou a JSR 301 para definir como um portlet 286 interage com os artefatos do JSF. A especificação foi para votação em Julho de 2006, mas somente em Janeiro de 2009 o draft da proposta final foi disponibilizada para o público.

Portlet Bridge é uma tecnologia utilizada por um portlet para fazer uma ponte (por isso o nome bridge) para um ambiente de execução, onde diferentes abstrações são utilizadas para processar as interações com o usuário ou para renderizar a interface gráfica.

O Portlet Bridge age como um *engine* tradutor entre o ambiente de portal e o sistema destino, neste caso, uma aplicação desenvolvida com o framework JSF. Ele ainda fornece uma abstração do ambiente de portal para o ambiente do bridge, deixando os desenvolvedores livres para executar aplicações Faces como portlets, sem a necessidade de conhecerem os detalhes das APIs de portlet e seu modelo de desenvolvimento.

A Figura 6 ilustra uma aplicação portlet utilizando o *JSF/Portlet Bridge* para executar páginas Faces como se fossem fragmentos de um portlet

5. PROPOSTA DE TRABALHO

A Figura 3 ilustra de forma geral o que se pretende fazer durante o desenvolvimento desse trabalho.

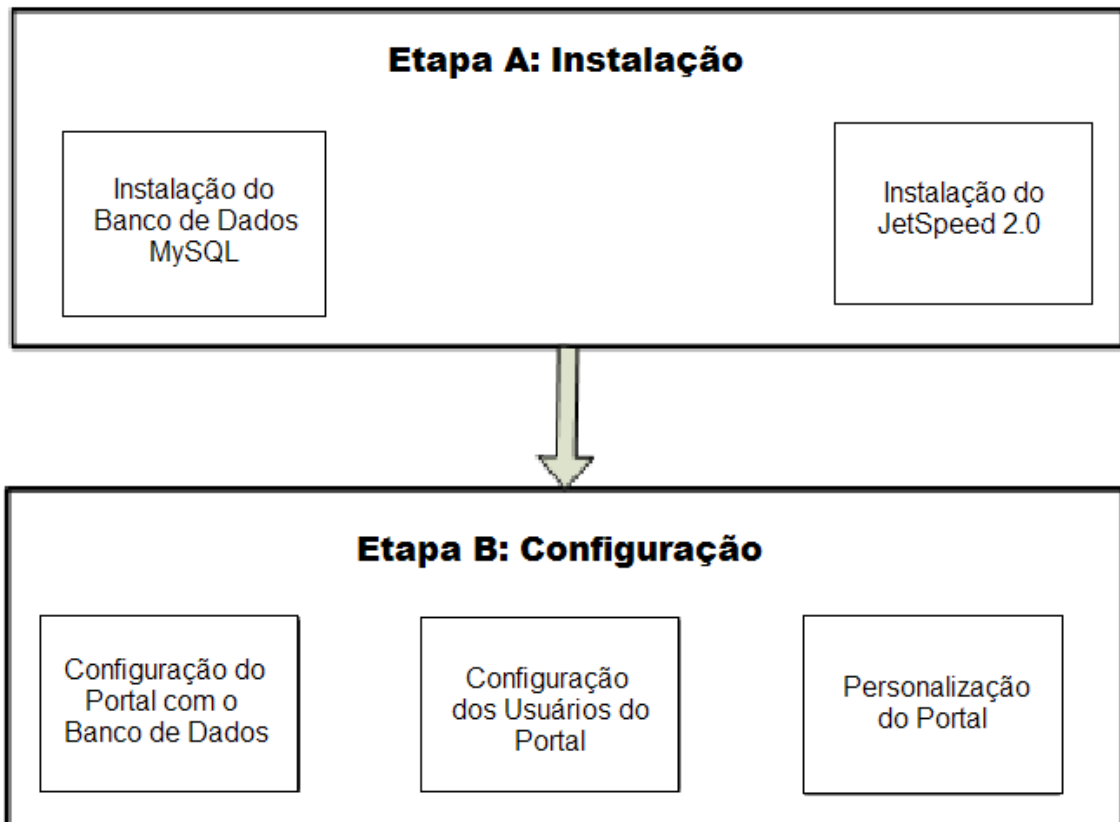


Figura 8. Proposta de Trabalho

Em suma a proposta consta de três etapas, sendo que a finalidade da Etapa A é preparar toda a infraestrutura de software; a Etapa B tem por finalidade realizar todas as configurações necessárias.

6. ESTUDO DE CASO

6.1 INSTALAÇÃO

Será explicada como é feita a instalação Framework JetSpeed 2, passando desde o download do *software* até a finalização de sua instalação. Qualquer versão dos programas citados serve. Para que toda essa aplicação funcione é necessário que tenha instalado antes de tudo, a Máquina Virtual Java (de preferência a versão 5 ou posterior). Os links para download dos softwares utilizados são:

<http://portals.apache.org/jetspeed-2/download.html> jetspeed-installer-2.2.1.jar

Jetspeed
Last Published: 27 April 2010

Applications | Portals | Jetspeed-2.1.3

Download Jetspeed-2 Distribution

Jetspeed-2 is distributed in several formats for your convenience and distributed under the [Apache License, version 2.0](#).

With version 2.2.1, there is now only one binary distribution: an installer. From the installation program, you can choose either the demo or minimal installation. The demo option is best for seeing more features, capabilities, and demo portlet applications. The minimal option could be useful for basing your own custom-portal distribution upon, or just seeing the basic feature set of Jetspeed.

1. Minimal Installer Option - only contains a Jetspeed Portal and the Jetspeed Administrative Portlets, with a minimal Site map and set of users
2. Full or Demo Option - contains a Jetspeed Portal, the Jetspeed Administrative Portlets, a full demo Site Map, 10 users. Additionally, there several portlet applications including an RSS application, a Database Browser application, a Web Content application, and a set of demo portlets including Google Maps, weather portlets, and lots of programming examples to get you started.

Jetspeed-2 2.2.1 Installer Distribution

Distribution	Mirrors	Checksum	Signature
Jetspeed-2 Standard Installer with both Minimal and Demo choices	jetspeed-installer-2.2.1.jar	here	here

The installer supports the following databases: Derby (default), DB2, MySQL, MSSQL, Oracle, PostgreSQL, SapDB, as well as manual (do it yourself) configuration of other databases.

An Ant script is provided for after installation, reinitializing or switching to another database in a one simple step operation.

Complete instructions for getting started using the installer is available [here](#).

Jetspeed-2 2.2.1 Portal Source Distribution

Figura 9. Pagina de download do JetSpeed - 2.

Para saber se possui uma Máquina virtual Java instalada é necessário entrar no prompt de comando e digitar a seguinte linha:

```
java -version
```

Após o comando deve mostrar algo parecido com o código abaixo:

```
java version "1.6.0_07"
```

```
Java(TM) SE Runtime Environment (build 1.6.0_07-b06)
```

Java HotSpot(TM) Client VM (build 10.0-b23, mixed mode, sharing)

Caso o código seja o mesmo apresentado, isto significa que possui uma máquina virtual Java instalada no seu computador. No caso de não ter a máquina virtual Java instalada é necessário entrar no site da Sun e fazer o seu download. O link é:

<http://java.sun.com/javase/downloads/index.jsp>

Para instalar o portal necessário seguir as seguintes etapas:

1ª etapa: Criar um banco de dados neste caso daremos o nome de portal ;

2ª etapa: Executar o arquivo `jetspeed-installer-2.2.1.jar` em um prompt de comando com a seguinte linha: `Java -jar jetspeed-installer-2.2.1.jar`;

Na Figura 10 ao executar *jetspeed-installer* será aberta uma janela de boas vindas.



Figura 10. Tela de Boas Vindas da instalação do JetSpeed - 2.

Na Figura 11 clique no botão *next* e será mostrada a tela com a licença do software.

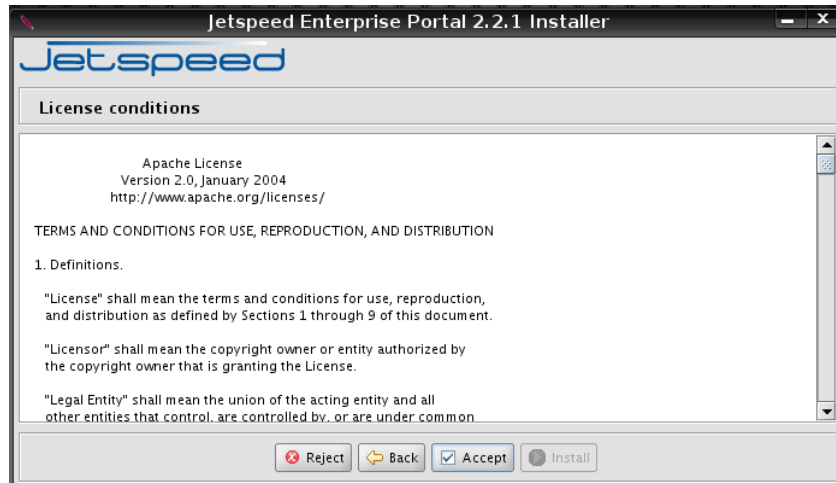


Figura 11. Tela com a licença do JetSpeed .

Na Figura 11 clique no botão *Accept* e será mostrada a tela onde será selecionada a opção a ser configurada.



Figura 12. Tela de seleção de opção .

Na Figura 12 selecione a opção *install JetSpeed Portal* e clique em *next*, será mostrada a tela de configuração do local a ser instalado o portal.

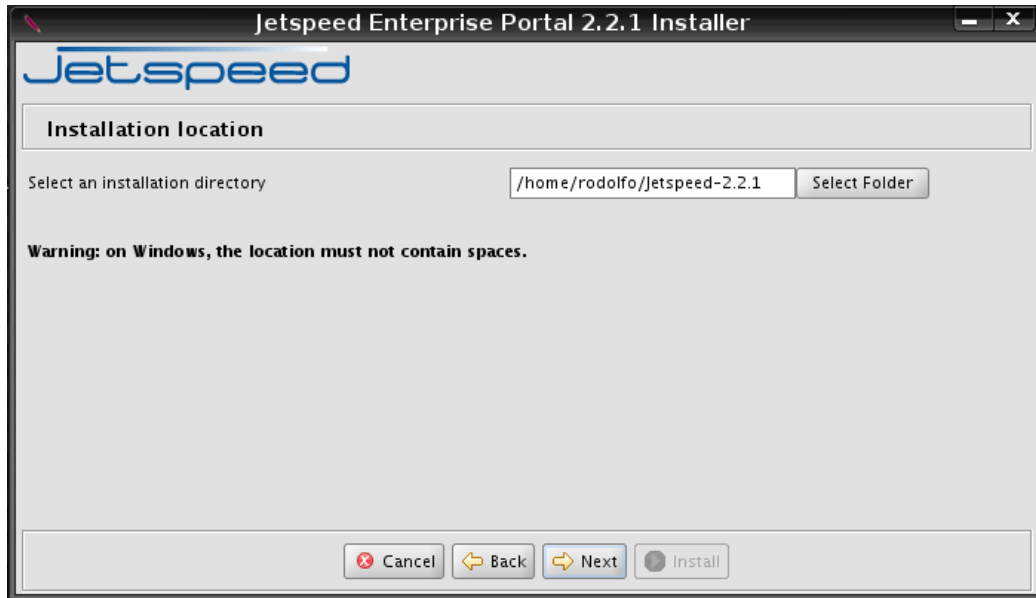


Figura 13. Tela de configuração do loção a ser instalado .

Na Figura13 selecione o local de sua preferência e clique em next se o diretório selecionada não existir será exibida uma mensagem dizendo que o diretório não existe e perguntando se pode criá-la cli Na Figura14 clique no botão Yes .

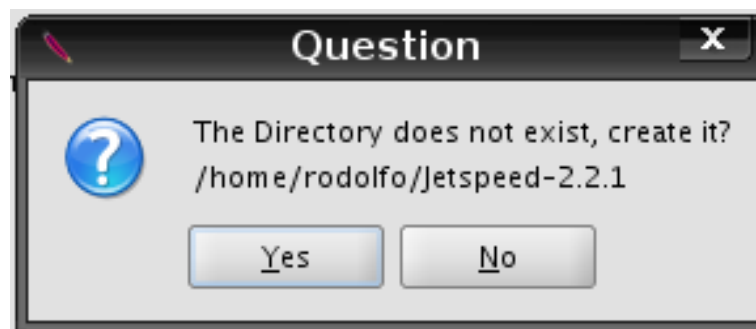


Figura 14. Tela de configuração do loção a ser instalado.

Na Figura15 Selecciona a opção Demo onde será instalada a engine do portal e alguns portlet de demonstração e a opção JetUI e clique no botão em next.

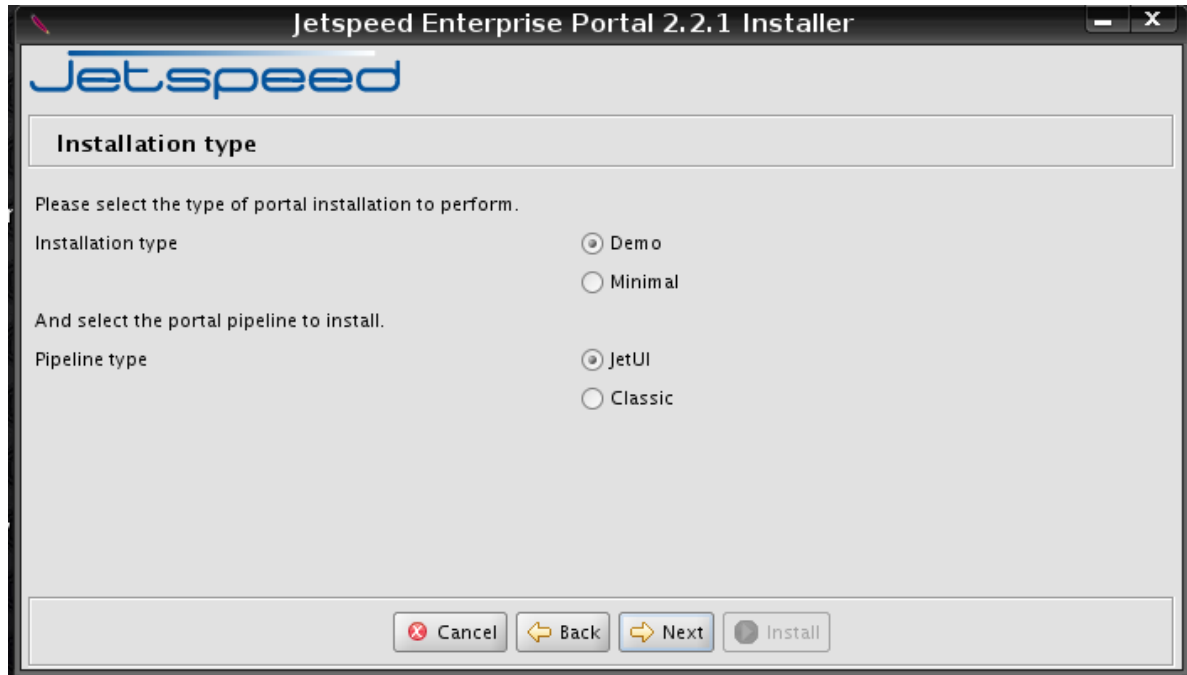


Figura 15. Tela de seleção do tipo a ser instalado.

Na Figura 16 não altere nada e clique no botão next .

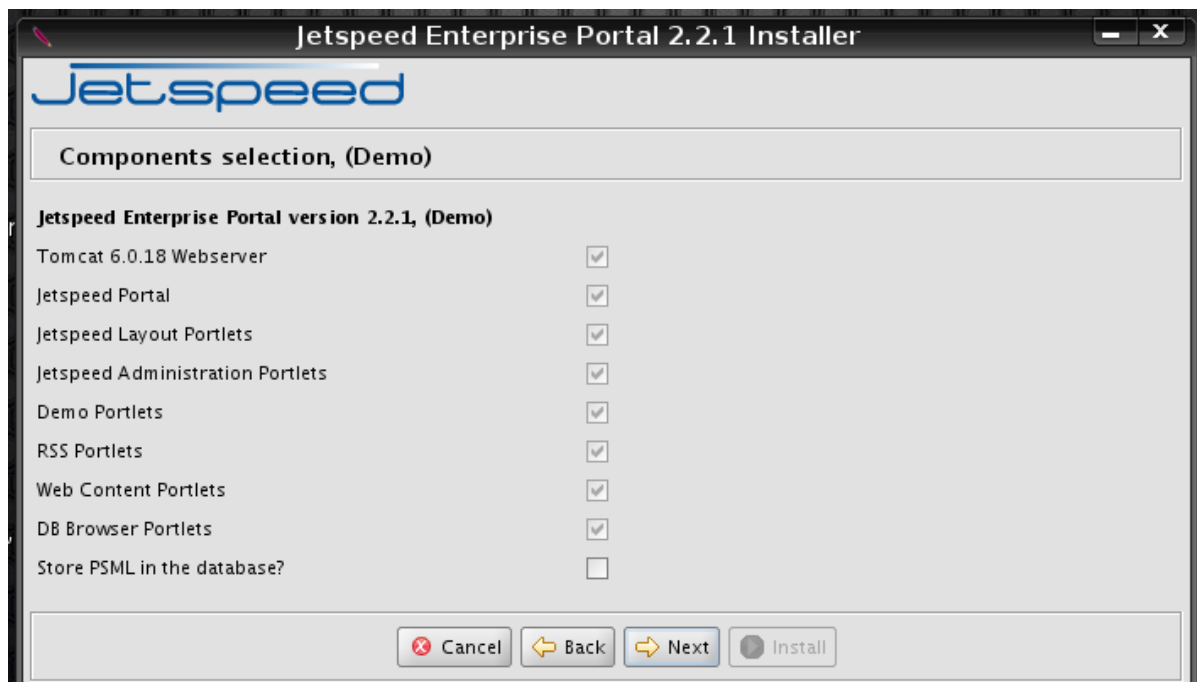


Figura 16. Tela de seleção dos componentes a ser instalado.

Na Figura 16 selecione a opção MySQL e clique no botão next.



Figura 17. Tela de seleção do SGBD.

Na Figura 17 no campo Database User Name coloque o nome do usuário do banco de dados, portal que foi criado anteriormente. No campo Database Password coloque a senha do banco de dados portal, no campo JDBC Connection String coloque jdbc:mysql://localhost:3606/portal e clique no botão Select File.

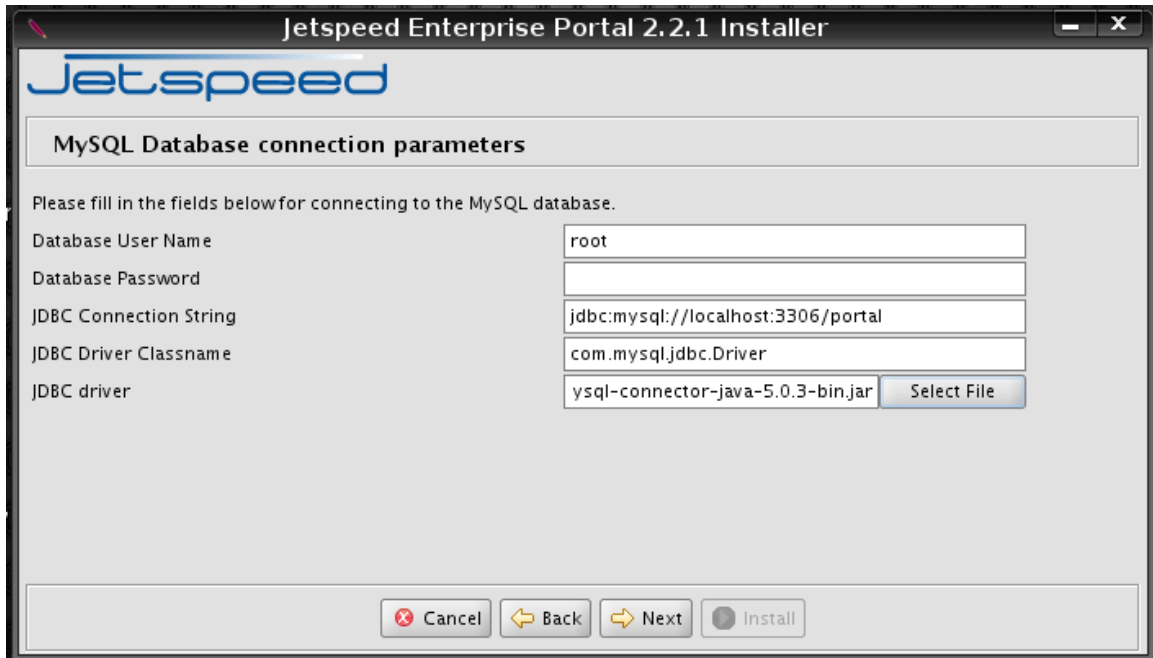


Figura 18. Tela de configuração da conexão com o SGBD.

Na Figura 19 selecione o arquivo `mysql-connector-java-5.0.3-bin.jar` (driver jdbc para o MySQL) e clique no botão *Select File* após na figura 18 clique no botão next.

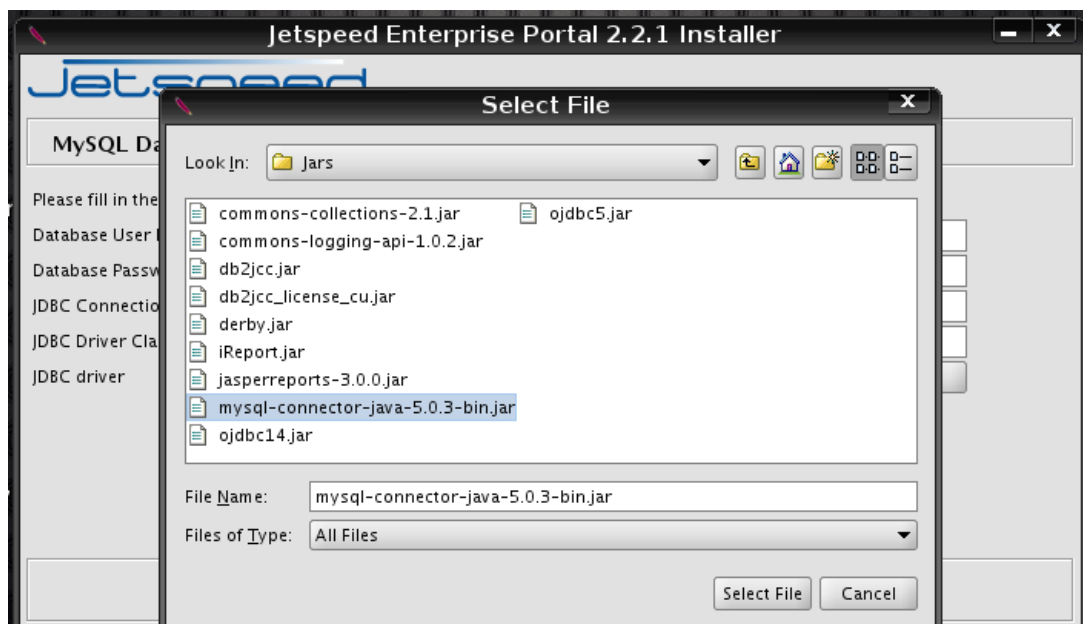


Figura 19. Tela de seleção do driver jdbc do SGBD.

Na Figura 20 clique no botão next para testar a conexão com o bando de dados.

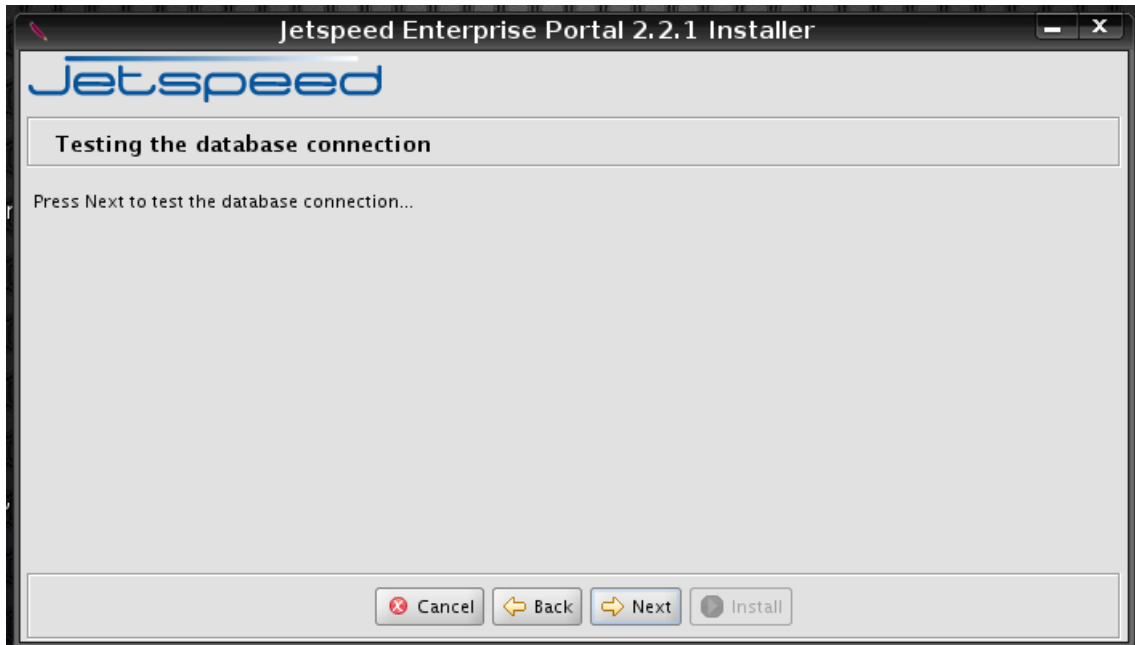


Figura 20. Tela de teste de conexão com o SGBD.

Se a conexão com o banco de dados não ocorrer erro, será exibida a tela informando os dados que foi configurado como local onde será instalado o portal e o banco de dados. Na Figura 21 clique no botão next.



Figura 21. Tela de informação.

Na Figura 22 clique no botão install para dar inicio a instalação do portal JetSpeed.

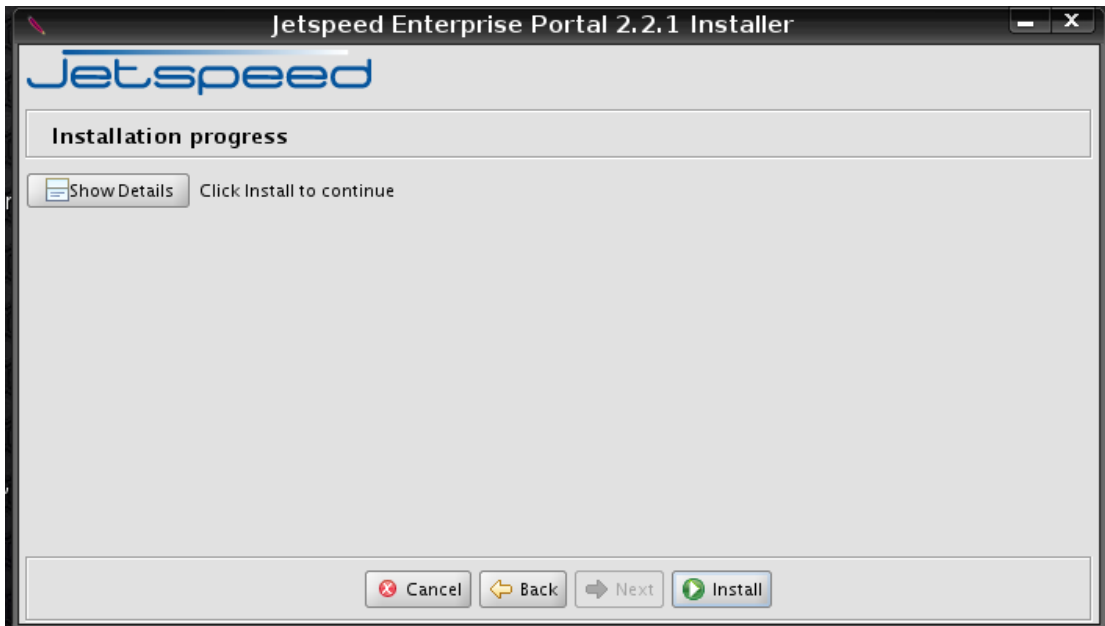


Figura 22. Tela de inicialização de instalação.

Na Figura 23 será exibida a mensagem de instalação completa clique no botão Ok

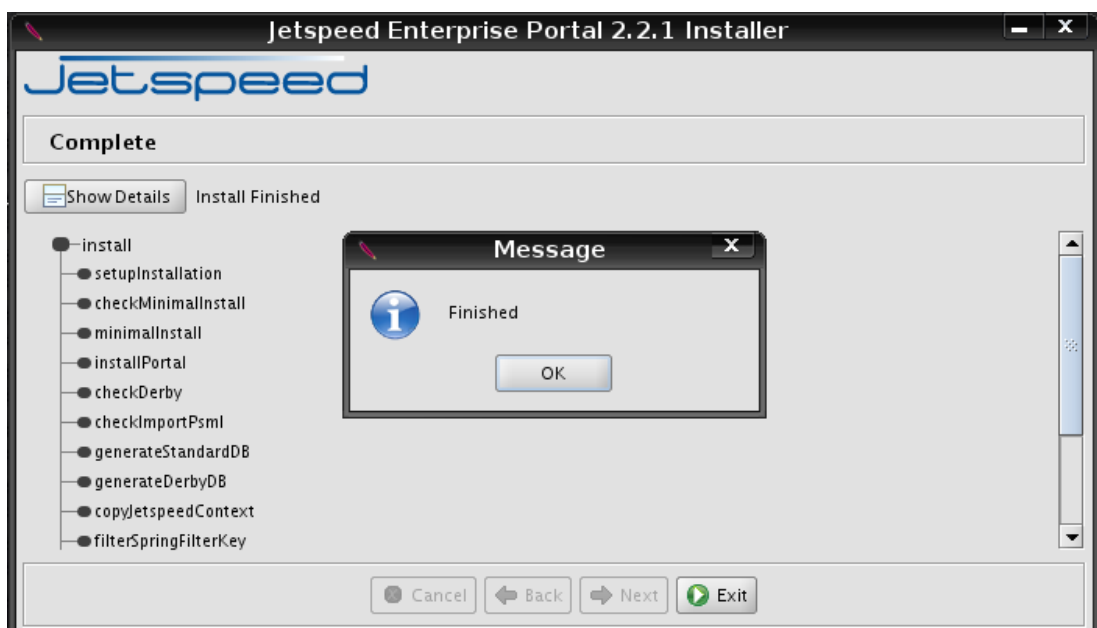


Figura 23. Tela instalação completa.

Com a instalação concluída é necessário testar o portal, para isso deve-se subir o servidor de aplicação pois a instalação do JetSpeed 2 instala o portal no servidor de aplicação Apache Tomcat. Para isso é preciso abrir o prompt de comando e através deste, acessar o diretório onde foi instalado o JetSpeed e executar o seguinte comando:

Linux = `./startup.sh`

Windows = `startup.bat`

Após, basta acessar o endereço <http://localhost:8080/> em um navegador de internet
Figura 24.

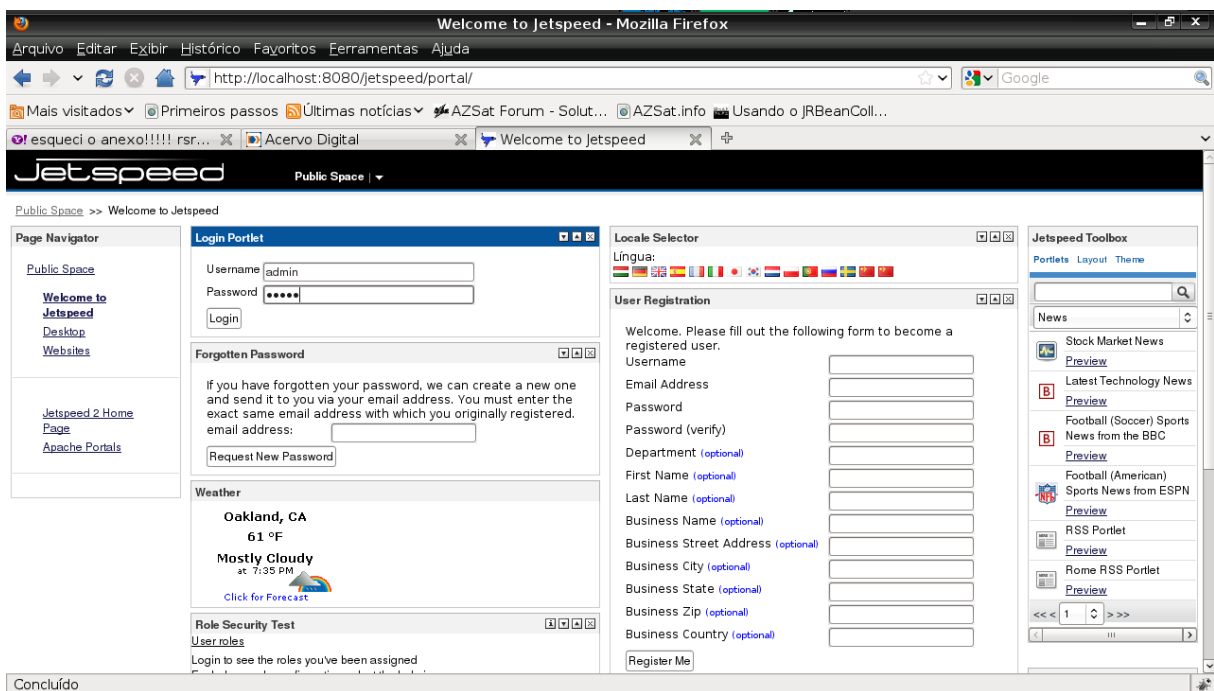


Figura 24. Tela inicial do portal instalado.

7. CONSIDERAÇÕES FINAIS

7.1 CONTRIBUIÇÕES

O desenvolvimento de software para integração de aplicações é um dos principais focos do mercado de trabalho nos dias de hoje. A construção de portais pode desenvolver esta tarefa suprimindo estas necessidades.

Este trabalho possibilitou a criação de uma base para o desenvolvimento de portais com a tecnologia de portlet. A principal etapa aqui foi o estudo de como funciona um portal e seus portlets com a aplicação do mesmo.

7.2 LIMITAÇÕES

Devido a dificuldade de encontrar material sobre o assunto não foi possível aprofundar em determinados conceitos sobre portal como Planejado no início do projeto. Este trabalho possui um levantamento de material sobre a tecnologia de desenvolvimento de portais baseados na JSR 168 o que auxilia o início de projetos futuros sobre o assunto.

REFERÊNCIAS

Jetspeed is an Open Source implementation of an Enterprise Information Portal
Disponível em: <http://portals.apache.org/jetspeed-1> Acesso em: 16 mar. 2010.

DINIZ, Vagner; CECONNI, Carlos. **Internet e web: passado, presente e futuro.**
Planeta web. São Paulo: W3C Escritório Brasil, jun/jul. 2008. Acesso em:
<http://www.w3c.br/palestras/internet-web-jun-jul-2008/internetWeb-02.html> Acesso
em maio. 2010.

COMPUTER HISTORY MUSEUM. Internet History. Acesso em:
http://www.computerhistory.org/internet_history Acesso em maio. 2010.

JAVA BOUTIQUE. Disponível em:
http://javaboutique.internet.com/reviews/Enterprise_Portals/. Acesso em: ago. 2010.

MANAGEABILITY - Open Source Portal Servers Written in Java. Disponível em:
http://www.manageability.org/blog/stuff/open_source_portal_servers_in_java. Acesso
em: ago. 2010.