

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

**Desenvolvimento de um Cliente de Webmail para Dispositivos
Móveis**

Leandro Boeing Vieira

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina
como parte dos requisitos para a obtenção do grau de Bacharel em Ciências da
Computação.

Florianópolis – SC

Outubro 2008

Leandro Boeing Vieira

Desenvolvimento de um Cliente de Webmail para Dispositivos Móveis

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina
como parte dos requisitos para a obtenção do grau de Bacharel em Ciências da
Computação.

Orientador: Professor Doutor João Bosco Manguiera Sobral

Banca Examinadora

Fernando Augusto da Silva Cruz

Lucas Guardalben

Agradecimentos

Agradeço aos meus pais Leonel e Maria pela oportunidade a mim dada e por nunca terem desacreditado que este momento chegaria. Também aos meus irmãos Léo Victor e Luiz Otávio pelo simples fato de existirem. À minha avó Nilza e aos meus tios Gilmara e Sérgio.

Aos professores que, cada um com sua didática, nos transmitiram seus conhecimentos e dividiram sua experiência ao longo desses semestres.

Aos bons amigos reconhecidos dentro do ambiente acadêmico, sejam eles professores, funcionários ou alunos; em especial a: Arthur F. Medeiros, George E. da Silva, Guilherme Ferreira, entre tantos outros que realizaram comigo esta grande conquista.

À minha namorada Natalia que, mesmo distante, está sempre presente.

Finalmente a Deus, por me presentear com as pessoas acima e com as oportunidades que apareceram em meu caminho.

Resumo

Para abrir as contas de e-mail do Departamento de Informática da UFSC (INE) era necessário um computador com acesso à Internet ou um dispositivo móvel recente que fosse capaz de interpretar HTML, abrindo o serviço de Webmail do departamento.

O presente trabalho refere-se ao desenvolvimento de um webmail para dispositivos móveis utilizando o protocolo WAP. O sistema foi desenvolvido utilizando a linguagem PHP do lado do servidor, e a linguagem WML no lado do cliente. Com esse webmail utilizando WML, agora é possível abrir as contas de e-mail tanto em dispositivos móveis mais antigos quanto nos mais recentes.

Por fim, o webmail foi instalado e testado nos servidores do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, tendo como resultado o fornecimento de mais uma opção para acesso à caixa de e-mail dos alunos e professores do departamento.

Palavras-chave: Desenvolvimento Web, Webmail, PHP, WAP, WML

Sumário

Lista de Figuras

Lista de Abreviaturas

1	Introdução	p. 9
1.1	Motivação e Justificativa	p. 10
1.2	Objetivos	p. 10
2	Internet	p. 12
2.1	História	p. 12
2.2	Protocolos da Internet	p. 13
2.2.1	Protocolos da Camada de Rede	p. 13
2.2.2	Protocolos da Camada de Transporte	p. 13
2.2.3	Protocolos da Camada de Aplicação	p. 14
	HTTP	p. 14
	POP	p. 15
	IMAP	p. 16
	SMTP	p. 16
2.3	Serviços da Internet	p. 16
2.3.1	Correio Eletrônico	p. 17
	Funcionamento	p. 17
2.3.2	World Wide Web	p. 18
	Funcionamento	p. 19

2.4	Webmail	p. 20
2.4.1	Protocolos da Internet	p. 20
2.4.2	Serviços da Internet	p. 21
3	WAP	p. 22
3.1	Histórico	p. 22
3.2	Arquiterura	p. 22
3.3	Segurança	p. 24
3.4	Limitações	p. 26
4	Projeto	p. 27
4.1	Requisitos	p. 30
4.2	Configuração	p. 31
4.3	Bibliotecas	p. 31
4.3.1	Output	p. 32
4.3.2	Mail	p. 32
4.3.3	User_Auth	p. 33
4.4	Implementação	p. 33
4.5	Implantação	p. 58
4.6	Usabilidade	p. 58
5	Conclusão	p. 61
	Referências	p. 62

Lista de Figuras

1	Transferência de Mensagens.	p. 18
2	Tela do Cliente de Webmail Squirrelmail	p. 21
3	Etapas da busca de uma URL por um Dispositivo Móvel	p. 23
4	Funcionalidades fornecidas pelo Warm Mail.	p. 28
5	Funcionalidades fornecidas pelo Wap Mail.	p. 29
6	Protocolos e respectivas portas	p. 31
7	Sistema no estado inicial.	p. 36
8	Sistema depois de efetuado login.	p. 38
9	Sistema depois de efetuado log-out.	p. 40
10	Formulário para enviar nova mensagem.	p. 43
11	Caixa de Entrada com a lista de mensagens do servidor.	p. 46
12	Mensagem sendo exibida.	p. 49
13	Respondendo a uma mensagem.	p. 52
14	Encaminhando uma mensagem.	p. 55
15	Apagando uma mensagem.	p. 57
16	Usabilidade do Wap Mail.	p. 60

Lista de Abreviaturas

URL: Uniform Resource Locator

URI: Uniform Resource Identifier

HTTP: Hypertext Transfer Protocol

POP: Post Office Protocol

SMTP: Simple Mail Transfer Protocol

WAP: Wireless Application Protocol

IMAP: Internet Message Access Protocol

HTML: HyperText Markup Language

WML: Wireless Markup Language

WWW: World Wide Web

1 *Introdução*

A Internet é um conjunto de redes em escala mundial com milhões de computadores interligados pelo Protocolo de Internet que permite o acesso a informações e transferência de dados.

Ultimamente, a Internet vem sendo amplamente utilizada como ambiente para sistemas devido à facilidade que ela fornece e aos serviços que oferece. Ou seja, o sistema está disponível em qualquer lugar e a qualquer momento.

Dois dos serviços mais utilizados, e principais responsáveis pelo sucesso e crescimento da Internet, são o serviço de Correio Eletrônico e a World Wide Web.

Um sistema de Correio Eletrônico é composto de programas de computador que suportam a funcionalidade de cliente de e-mail e de um ou mais servidores de e-mail que, através de um endereço de correio eletrônico, conseguem transferir uma mensagem de um usuário para outro. Assim é possível enviar e receber uma mensagem de e-mail através da Internet.

A World Wide Web (também conhecida como Web e WWW) é um sistema de documentos que são interligados e executados na Internet. Os documentos podem estar na forma de vídeos, sons, hipertextos e figuras. Para visualizar a informação, pode-se usar um programa de computador chamado navegador para visualizar informações (também chamadas "documentos" ou "páginas") de servidores web. O usuário pode então seguir as hiperligações na página para outros documentos ou mesmo enviar informações de volta para o servidor para interagir com ele.

Unindo os serviços de Correio Eletrônico e a World Wide Web surgiram os webmails. Nesses são utilizadas páginas web para a apresentação e utilização dos protocolos envolvidos no envio e recebimento de e-mail e o conteúdo pode ser acessado facilmente em qualquer lugar através de um sistema de autenticação pela WWW.

É no contexto da utilização da World Wide Web, ou simplesmente Web, como am-

biente de sistemas, que esse projeto de conclusão de curso se embasa. Neste caso, um navegador interpretará as páginas do sistema desenvolvido fornecendo uma interface entre o dispositivo sem fio (celular, pager, PDA, etc) e os servidores de e-mail.

1.1 Motivação e Justificativa

É com uma frequência cada vez maior que precisamos de informações atualizadas e facilmente disponibilizadas ao nosso alcance. Dentro dessa necessidade recebemos diariamente e-mails importantes às vidas pessoal, acadêmica e profissional.

Geralmente, para acessar uma conta de e-mail se faz necessária a instalação de um aplicativo para a leitura ou o envio de um correio eletrônico. São os chamados clientes de e-mail. Contudo, um webmail permite o acesso a uma conta de e-mail sem a precisão de um software cliente especializado para isso. Com esse sistema, dispendo de um computador com conexão à Internet e um navegador para exibir uma página da Web torna-se possível acessar o servidor e administrar as mensagens eletrônicas.

Entretanto, na falta de um computador era impossível, até então, visualizar a caixa de entrada de um e-mail ou enviar uma mensagem dentro do ambiente da Universidade Federal de Santa Catarina. O webmail desenvolvido e apresentado nesse trabalho supre essa carência, sendo possível que um usuário acesse sua conta de correio eletrônico através de um celular ou outro aparelho móvel que disponha de conexão à internet e um navegador WAP, também chamado de microbrowser.

1.2 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de um webmail para dispositivos móveis. Para implementar o sistema são necessários uma linguagem que manipule dados dinamicamente do lado do servidor Web, uma linguagem que seja interpretada e visualizada no aparelho móvel do lado do cliente e um servidor de correio eletrônico que forneça as funções de envio e recebimento de um e-mail.

O Departamento de Informática e Estatística da UFSC tem os serviços e servidores acima instalados e devidamente configurados para dar suporte ao sistema desenvolvido. Usando essas tecnologias disponíveis o webmail permitirá ao usuário:

- Conectar-se à uma conta no servidor de e-mail

- Ler correio
- Enviar correio
- Responder às mensagens
- Encaminhar mensagens
- Excluir correio

2 *Internet*

2.1 História

O medo iminente de que a Guerra Fria pudesse se tornar uma guerra nuclear com danos materiais e tecnológicos irreversíveis motivou o governo norte-americano a desenvolver um sistema para que seus computadores militares pudessem trocar informações entre si, de uma base militar para outra e que mesmo em caso de ataque os dados fossem preservados. Foi nesse âmbito que surgiu a ARPANet, antecessora da Internet, entre as décadas de 60 e 70. Era um projeto do Departamento de Defesa dos Estados Unidos que interconectou computadores, através de um sistema chamado de chaveamento de pacotes. Tal esquema transmitia dados em redes de computadores no qual as informações eram divididas em pacotes que continham trechos de dados, endereço do destinatário e informações que permitiam a concatenação dos trechos de forma a remontar a mensagem original.

Em seguida, o ambiente universitário se mostrou interessado na tecnologia recém-criada. Assim, várias universidades adotaram a ARPANet para trocar experiências e resultados de seus estudos e pesquisas de forma ágil, para a época, entre seus estudantes e professores. Graças a essa nova aplicação da ARPANet, a rede avançou consideravelmente em poucos anos funcionando sob um protocolo funcionalmente completo, o NPC (Network Control Protocol).

Entretanto, a ARPANet, até então, não tinha a interatividade que a Internet tem hoje. Com a criação da World Wide Web pelo cientista Tim Berners-Lee junto com o CERN, European Organization for Nuclear Research - Centro Europeu de Pesquisas Nucleares, era possível agora dar mais dinamismo ao conteúdo veiculado pela Internet, passando a exibir não somente texto mas também conteúdo multimídia através de páginas HTML sob o protocolo HTTP.

2.2 Protocolos da Internet

São necessárias três camadas de protocolos para que a Internet consiga estabelecer conexões e trocar informações de ponto a ponto na rede. Na camada mais baixa, a de rede, o IP (*Internet Protocol*) ou Protocolo de Internet transfere dados de um ponto a outro da rede. Na camada intermediária, a de transporte, o TCP (*Transmission Control Protocol*) realiza uma conexão virtual para a comunicação de dados em ambos os sentidos, envio e recebimento. Na última camada, a mais alta, estão os protocolos de aplicação que dão suporte para os serviços da Internet.

Os protocolos da Internet foram desenvolvidos para serem independentes do meio físico de transmissão. Qualquer rede de comunicação, seja através de cabos ou sem fio, que seja capaz de transportar dados digitais de duas vias é capaz de transportar tráfego da Internet. Por isso, os pacotes Internet podem ser transmitidos por uma variedade de meios de conexão tais como cabo coaxial, fibra ótica, redes sem fio ou por satélite.

2.2.1 Protocolos da Camada de Rede

São os protocolos que definem pacotes que carregam dados de um nó para outro nó da rede. Essa camada é importante pois independe de arquitetura e organização do computador para que dois sistemas se comuniquem e troquem informações entre si. Por exemplo, é possível que um dispositivo móvel se conecte a um servidor Web, ou que uma máquina rodando Sistema Operacional Windows se comunique diretamente com uma máquina rodando Linux. O IP, *Internet Protocol*, é um dos protocolos de rede mais utilizados.

2.2.2 Protocolos da Camada de Transporte

É a camada intermediária entre a de Rede e de Aplicação. É nessa camada que se garante que os dados cheguem íntegros ao receptor sem sofrer danos e na ordem correta que foram enviados. O TCP, *Transmission Control Protocol*, é um protocolo desta camada e é largamente utilizado, junto com o IP, para estabelecer conexões e transferências de dados via Internet.

2.2.3 Protocolos da Camada de Aplicação

As camadas anteriores existem para criar uma abstração para esta. São esses protocolos que definem mensagens específicas e formatos digitais (arquivos de imagem, arquivos de texto, etc) comunicados pelas aplicações. Alguns protocolos dessa camada podem ser destacados:

- POP (*Post Office Protocol*)
- IMAP (*Internet Message Access Protocol*)
- SMTP (*Simple Mail Transfer Protocol*)
- HTTP (*Hypertext Transfer Protocol*)

HTTP

O HyperText Transfer Protocol (HTTP) é um protocolo de aplicação responsável pelo tratamento de pedidos/respostas entre cliente e servidor na World Wide Web.

Em 1989, o CERN (European Particle Physics Laboratory) iniciou um projeto interno bem sucedido, que levou a um esforço para criar padrões para passar este novo tipo de informação através da Internet. Os componentes básicos consistiam de padrões para a criação de arquivos hipertexto multimídia e um padrão para servir estes arquivos padronizados quando requisitados. O padrão de arquivo é chamado HyperText Markup Language, ou HTML. O padrão servidor é chamado HyperText Transfer Protocol, ou HTTP. Um servidor HTTP enviará arquivos HTML através da Internet para outros clientes que os requisitem.

Este protocolo tem sido usado pela WWW desde 1990. A primeira versão de HTTP, chamada HTTP/0.9, era um protocolo simples para a transferência de dados no formato de texto ASCII pela Internet, através de um único método de requisição, chamado GET. A versão HTTP/1.0 foi desenvolvida entre 1992 e 1996 para suprir a necessidade de transferir não apenas texto. Com essa versão, o protocolo passou a transferir mensagens do tipo MIME (Multipurpose Internet Mail Extension) e foram implementados novos métodos de requisição, chamados POST e HEAD.

A versão atual do HTTP é a 1.1 definida pelo RFC 2616 em 1999 e se mantém constante desde então.

As etapas para uma conexão a um servidor Web via protocolo HTTP são:

- Um programa cliente estabelece uma conexão com um servidor
- O cliente envia uma requisição contendo basicamente: endereço, versão do protocolo e uma mensagem MIME
- O servidor envia para o cliente: versão de protocolo, código de operação (bem sucedida ou erro) e possível conteúdo no corpo da mensagem
- Encerra-se a conexão estabelecida.

POP

O Post Office Protocol é um protocolo utilizado no acesso remoto a uma caixa de correio eletrônico. Ele permite que as mensagens contidas numa caixa de correio eletrônico possam ser transferidas seqüencialmente para um computador local.

As etapas para leitura de um correio com este protocolo são as seguintes:

- É estabelecida uma conexão TCP entre o servidor de e-mail e o cliente
- O usuário autentica-se com login e senha
- As mensagens existentes na caixa de correio são transferidas para o computador cliente
- A conexão é encerrada

Depois que as mensagens são transferidas, é possível que as mesmas sejam apagadas do servidor, se assim o protocolo estiver configurado. Essa opção é útil quando deseja-se economizar espaço ou quando todas as mensagens são enviadas para o mesmo computador cliente, não havendo a necessidade, portanto, de visualizar as mensagens posteriormente em um outro meio.

Devido a essas etapas o POP é dito um protocolo *offline*, uma vez que depois de estabelecida a conexão, transferidas as mensagens e encerrada a conexão o usuário pode visualizar os correios sem a necessidade de estar conectado à Internet.

IMAP

O Internet Message Access Protocol é um protocolo de gerenciamento de correio eletrônico superior em recursos ao POP3. Uma vantagem deste protocolo é o compartilhamento de caixas postais entre usuários membros de um grupo de trabalho. Além disso, é possível efetuar pesquisas por mensagens diretamente no servidor, utilizando palavras-chaves.

SMTP

Simple Mail Transfer Protocol é o protocolo padrão para envio de e-mails através da Internet. Ele é um protocolo relativamente simples, baseado em texto, onde especifica-se um destinatário, ou vários, e então a mensagem é transferida.

Na década de 1980 se deu a utilização em escala do SMTP. Nessa época era um complemento ao UUCP, que era mais adequado para transferências de correio eletrônico entre máquinas sem ligação permanente. Por outro lado, o desempenho do SMTP aumenta se as máquinas envolvidas, emissor e receptor, se encontrarem ligadas permanentemente, como é o caso dos servidores de e-mail.

O SMTP é um protocolo de envio apenas, o que significa que ele não permite que um usuário descarregue as mensagens de um servidor. Para isso, é necessário um cliente de email, ou um webmail, com suporte ao protocolo POP3 ou IMAP.

2.3 Serviços da Internet

A Internet, através dos seus protocolos, oferece um conjunto de serviços para comunicação e transferências de arquivos entre dois computadores. Dentre os serviços suportados pode-se destacar:

- Correio Eletrônico
- World Wide Web
- Acesso Remoto

2.3.1 Correio Eletrônico

O correio eletrônico existe antes mesmo da Internet, como é conhecida hoje, e foi uma das ferramentas mais importantes para o surgimento desta.

O primeiro autor e a data correta do sistema de correio eletrônico são desconhecidos. Acredita-se que os primeiros sistemas de troca de mensagens criados tenham sido o Q32 da SDC e o CTSS do MIT no final da década de 1960.

Nessa época a rede ARPANET tinha grande sucesso entre o meio acadêmico e teve uma grande contribuição para a evolução do e-mail. Em 1971 o programador Ray Tomlinson iniciou o uso do sinal @ para separar o nome do usuário do endereço da máquina e se tornou um padrão utilizado até hoje. Ele também desenvolveu os programas SNDMSG e READMAIL funcionando sob a ARPANET, o que aumentou significativamente a popularidade do correio eletrônico.

Funcionamento

Para uma simples troca de mensagens entre dois usuários, pode ser necessária a utilização de vários protocolos e de várias aplicações, independente de ser uma rede local ou utilizar a Internet.

Um usuário que queira enviar uma mensagem para outro utilizará um aplicativo cliente de e-mail, também conhecido como MUA (*Mail User Agent*). Ao terminar de redigir a sua mensagem o MUA a enviará para um MTA (*Mail Transport Agent*) que se encarregará então de entregar a mensagem ao MTA do destinatário, caso ele se encontre em outra máquina ou simplesmente colocar a mensagem na caixa postal do destinatário, caso ele se encontre no mesmo servidor. A transferência da mensagem entre o MUA e o MTA se efetua utilizando-se o chamado SMTP. O protocolo SMTP será utilizado também entre o MTA do remetente e o MTA do destinatário.

Quando o servidor de e-mail do destinatário recebe uma mensagem ele a coloca na caixa postal deste usuário. Para ler as mensagens, então, o usuário deverá transferi-las para sua máquina com o seu cliente de e-mail. A transferência de mensagens recebidas entre o servidor e o cliente de e-mail requer a utilização de outros programas e protocolos.

Usualmente é utilizado para este fim o protocolo POP que guarda as mensagens dos usuários em caixas postais e aguarda que estes venham buscá-las, analogamente como ocorre com uma caixa postal dos Correios. Outro protocolo que pode ser utilizado para

este mesmo fim é o IMAP que implementa muitos outros recursos além das funcionalidades fornecidas pelo POP.

O recebimento de mensagens pelo cliente se dá através da solicitação do MUA do usuário ao seu servidor de e-mail, que após a autenticação do usuário vai informar se existem mensagens em sua caixa postal e quantas são. A seguir o MUA solicita a transferência das mensagens para a máquina local, finalizando assim o processo de troca de mensagens entre dois usuários. A figura, obtida em (WILEY, 2008), resume o processo:

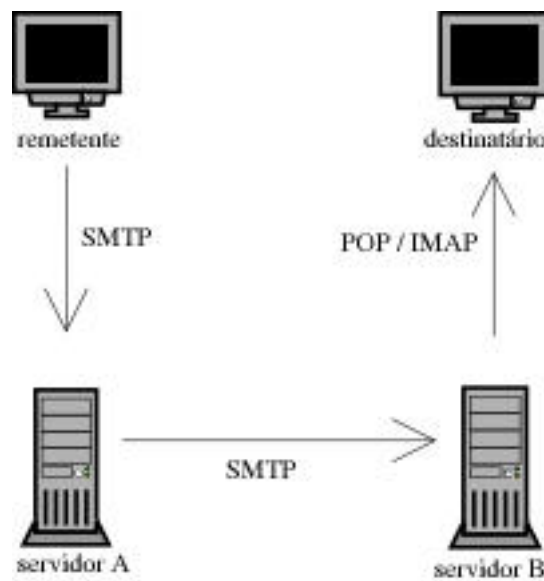


Figura 1: Transferência de Mensagens.

2.3.2 World Wide Web

A Web surgiu da necessidade que Tim Berners-Lee, que trabalhava no CERN (*Conseil Européen pour la Recherche Nucléaire*), tinha de tornar mais fácil o compartilhamento de documentos de pesquisa entre os colegas. Assim ele construiu o ENQUIRE que, ainda que diferente da Web atualmente, tinha os mesmos objetivos e idéias primordiais.

Em março de 1989, Berners-Lee escreveu uma proposta de gerenciamento da informação (*Information Management: A Proposal*) que descrevia um sistema de informação bem elaborado e referenciava seu projeto, o ENQUIRE. Em 1990, com a ajuda de Robert Cailliau, publicou uma segunda proposta (*WorldWideWeb: Proposal for a HyperText Project*) para a já então World Wide Web.

No final de 1990, Berners-Lee havia construído todas as ferramentas necessárias para a

World Wide Web: um navegador, um servidor que ele usou em um computador chamado NeXTcube e as primeiras páginas web que descreviam o próprio projeto Web. A estréia da Web como um serviço publicado na Internet data de 6 de agosto de 1991, quando ele postou um resumo do seu projeto no grupo de notícias alt.hypertext.

O conceito do hipertexto originou-se em projetos como Xanadu e NLS dos anos de 1960. Esses projetos previam a utilização de apontadores e hiperligações entre os documentos. Tim uniu o hipertexto e a Internet, já que ninguém havia dado crédito à sua idéia. No processo, ele desenvolveu um sistema de identificação global e único de recursos, o URI (*Uniform Resource Identifier*).

O grande sucesso da WWW começou quando a CERN anunciou, em 30 de abril de 1993, que o sistema não seria software proprietário tornando a criação de outros sistemas e extensões livres de licenciamento. O segundo marco foi com a invenção do Mosaic, no mesmo ano, que é um navegador gráfico possibilitando ter na mesma página elementos gráficos e de texto.

Funcionamento

Para visualizar uma página da Web é necessário um computador conectado à rede Internet e um navegador. Normalmente inicia-se a navegação digitando uma URL (*Uniform Resource Locator*) e a partir daí acessando as hiperligações.

A URL é composta pelo protocolo, máquina, caminho e recurso organizados na seguinte estrutura:

protocolo://máquina/caminho/recurso

Um exemplo de URL é a seguinte: `http://www.inf.ufsc.br/~admrede/index.html`, onde:

- HTTP é o protocolo
- `www.inf.ufsc.br` é o servidor
- `~admrede` é o caminho dentro de um sistema de arquivos
- `index.html` é o recurso requisitado

Primeiramente, a parte da URL referente ao servidor web é separada e transformada em um endereço IP, por um banco de dados da Internet chamado Domain Name System

(DNS). O navegador estabelece então uma conexão TCP-IP com o servidor Web localizado no endereço IP retornado.

Depois o navegador envia uma requisição de acordo com o protocolo, geralmente o HTTP, ao servidor para obter o recurso indicado pela parte restante da URL dentro do caminho especificado. No caso de uma página Web típica, o texto HTML é recebido e interpretado pelo navegador, que realiza então requisições adicionais para figuras, arquivos de formatação, arquivos de script e outros recursos que fazem parte da página.

2.4 Webmail

Webmail é uma interface da World Wide Web que permite ao usuário ler e escrever e-mail usando um navegador, interagindo assim com seu Correio Eletrônico.

A maior vantagem de um Webmail é o fato de não precisar de um programa especializado para a leitura e o envio de mensagens de correio eletrônico. Basta apenas um computador conectado à Internet e um navegador para interpretar as páginas do sistema. Outra vantagem é que as mensagens não são salvas no computador que acessou a caixa de entrada do correio, mas ficam armazenadas no servidor. Assim, não é necessário utilizar sempre o mesmo computador.

A desvantagem fica por conta do espaço de armazenamento do servidor que pode ficar comprometida. Eventualmente o usuário deverá apagar as mensagens mais antigas ou de menos importância julgadas pelo mesmo, já que este possui uma quota limite de armazenamento de correio.

2.4.1 Protocolos da Internet

Além dos protocolos TCP/IP utilizados por padrão para acesso à rede, um webmail utiliza alguns protocolos da camada de aplicação:

- HTTP (*Hypertext Transfer Protocol*)
- POP (*Post Office Protocol*)
- IMAP (*Internet Message Access Protocol*)
- SMTP (*Simple Mail Transfer Protocol*)

2.4.2 Serviços da Internet

Dentre os diversos serviços oferecidos pela Internet, em um Webmail utilizam-se o Correio Eletrônico e a World Wide Web. O primeiro serve para dar acesso à caixa de correio do usuário permitindo que esse leia, responda e apague, entre outras ações, os e-mails existentes. O segundo é para dar uma interface à essa comunicação, sem a necessidade de um aplicativo especializado instalado no computador. Com um simples navegador para acessar a Web o usuário pode interagir com o servidor de e-mail.

A figura a seguir mostra a tela de um cliente de Webmail instalado no servidor do Departamento de Informática e Estatística, o Squirrelmail, sendo aberto em um navegador chamado Mozilla Firefox:

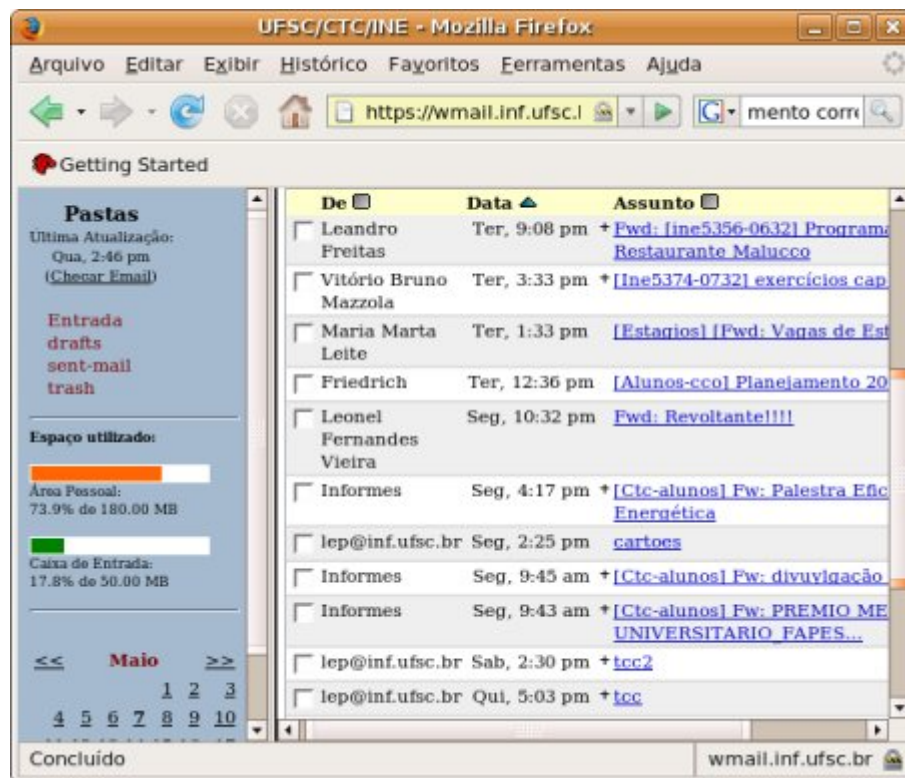


Figura 2: Tela do Cliente de Webmail Squirrelmail

Especificamente para um Webmail, ou qualquer site, que rode em dispositivos móveis, como celulares, é necessário também o protocolo WAP (*Wireless Application Protocol*). Este protocolo será objeto de estudo do próximo capítulo.

3 WAP

3.1 Histórico

O WAP, *Wireless Application Protocol*, não se resume a somente um protocolo. Ele pode ser entendido como um conjunto de compilação de protocolos que reunidos cobrem muitas necessidades das comunicações sem fio. Em seus protocolos, WAP fornece uma comunicação sem fio rápida, segura e eficaz.

Na década de 1990 as empresas Nokia, Ericsson, Motorola e Phone.com pesquisavam individualmente a tecnologia de comunicação sem fio até que resolveram, em 1997, se juntar para formar o WAP Forum. O objetivo dessa junção era de unir esforços para criar aplicações para a Internet incluindo os sistemas digitais que as mesmas produziam, como celulares, PDA's, pagers, etc. O resultado foi a criação do Protocolo de Aplicações Sem-fios.

Em cima do WAP, o WAP Forum se consolidou como um órgão independente, desvinculado dos seus criadores. Baseado nos padrões da Internet, como IP e XML, o protocolo foi adotado como um padrão para comunicação em redes wireless.

Em 2002 o WAP Forum, que tinha aproximadamente 200 companhias associadas, se tornou OMA (*Open Mobile Alliance*), que agora é a fonte oficial para WAP. O objetivo da OMA é consolidar em uma organização todas as especificações existentes nos serviços para redes sem-fio.

3.2 Arquiterura

A comunicação entre dispositivos WAP e a Internet é feita usando o mesmo padrão da WWW, utilizando URLs para localizar endereços e conteúdos nos servidores ou diretamente o endereço URI para acessar as páginas. A diferença está na maneira como é feita a conexão com a Internet.

O dispositivo sem-fio estabelece um circuito de troca de dados (CSD - Circuit Switch Data) junto ao hardware da operadora, estabelecendo uma linha de dados e não de voz. Agora, o dispositivo está pronto para fazer requisições à rede, entretanto ele não faz essa requisição diretamente. As requisições HTTP são feitas através de um servidor, chamado de WAP Gateway.

O WAP Gateway, quando recebe da URL consultada, faz uma verificação de sintaxe no código WML recebido, verificando se não está mal formatado e se seu tamanho não ultrapassa a capacidade suportada pelo celular. Depois compila tal código, substituindo as tags do WML por códigos binários que ocupam menos espaço, otimizando assim a banda de passagem, e enfim, remete o código compilado para o celular a fim de que ele seja renderizado e/ou executado pelo microbrowser.

O WAP Gateway traduz os pedidos da pilha de protocolos WAP para a pilha de protocolos WWW, conforme a figura extraída do livro *Desenvolvendo para WAP com WML*, (ARAÚJO, 2001):

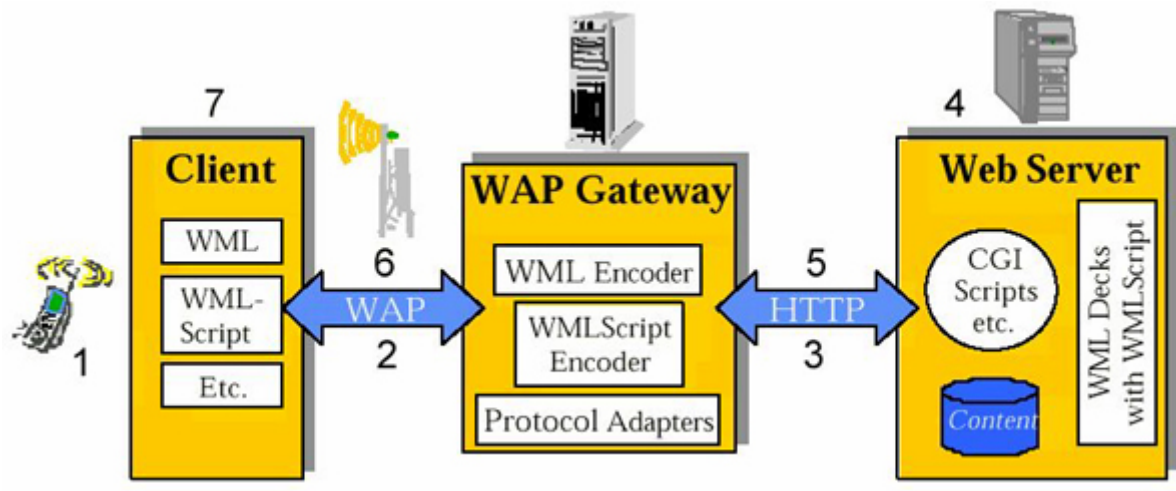


Figura 3: Etapas da busca de uma URL por um Dispositivo Móvel

1. Usuário pressiona uma tecla do aparelho celular para que ele faça a requisição de uma URL específica.
2. O dispositivo então manda a requisição de URL para o WAP Gateway, usando para isso a pilha de protocolos WAP.
3. O WAP Gateway cria então um HTTP convencional acessando o WEB Server (ou qualquer outro serviço disponível por HTTP) especificado pela URL.

4. No WEB Server, a requisição HTTP é processada, seja ela um cgi-bin, um Java Servlet, ou um script PHP, executando as instruções dadas pela URL.
5. O WEB Server retorna então um WML, informando no header da resposta que o content-type é 'text/vnd-wap-wml' em vez do habitual 'text/html'. Este código pode ser estático ou gerado dinamicamente por um script.
6. O WAP Gateway, recebe a resposta, então verifica o header da mesma e o conteúdo WML recebido, compilando-os em um formato binário. Depois disso, cria uma resposta WAP e envia de volta ao dispositivo sem fio.
7. O dispositivo recebe a resposta WAP, faz o parsing no WML mostrando as informações nele descritas.

Essa arquitetura projetada em cima da WWW busca maximizar a transferência de dados otimizando os protocolos já existentes para tal, como o HTTP e o TCP. Assim, a maciça troca de informações ocorre no ambiente da Internet, diminuindo a taxa de transmissão no ambiente sem-fio.

3.3 Segurança

Quando se fala em sistemas na Web, a principal preocupação é em relação às transações efetuadas. Informar dados de documentos, senhas e números de conta em banco deixam o usuário receoso antes de enviar esses dados. E no ambiente wireless a preocupação aumenta.

Na Internet, quando necessita-se fazer uma transação de forma segura, dois protocolos são mais utilizados: SSL e TLS. Ambos objetivam garantir privacidade e integridade dos dados, criptografando os dados quando são enviados, passando pela camada de transporte da rede, e descriptografando do outro lado no receptor.

O protocolo SSL é bastante utilizado na WWW para codificar o fluxo de dados entre o browser e o servidor Web, e é também utilizado para o ambiente WAP. Contudo, o SSL só é utilizado entre o servidor Web e o servidor WAP gateway. No transporte entre o WAP gateway e o dispositivo WAP é utilizado um sistema semelhante, chamado WTLS (*Wireless Transport Layer Security*). Esta camada foi especialmente desenvolvida para ser utilizada em ambiente wireless.

O principal objetivo da da camada WTLS, também chamada de protocolo WTLS, é oferecer privacidade, integridade dos dados e autenticação entre duas aplicações enquanto estas se comunicam, tal como acontece em SSL e TLS. A camada WTLS foi projetada para trabalhar com pacotes de dados em uma rede de alta latência e banda curta.

Algumas características do WTLS podem ser destacadas:

- Integridade de dados;
- Privacidade;
- Autenticação;
- Proteção contra Denial-of-Service.

O WTLS possui facilidades para assegurar que os dados enviados entre o terminal e o servidor de aplicação não sofram alterações, garantindo assim a integridade dos dados.

Possui, também, facilidades para assegurar que a informação transmitida através do terminal a um servidor de aplicação seja privada.

O WTLS facilita o estabelecimento da autenticidade do terminal e do servidor.

Por fim, o protocolo WTLS consegue detectar e rejeitar dados que foram duplicados ou não foram verificados com sucesso, fazendo com que ataques típicos de denial-of-service sejam mais difíceis de serem executados, protegendo as camadas de protocolos acima dele.

A camada WTLS é dividida em dois protocolos, que são o Protocolo de Handshake e o Protocolo de Gravação. O acompanhamento da comunicação entre estes protocolos é realizado através de primitivas de serviços. Estas primitivas representam, de maneira abstrata, a troca de informação e controle entre a camada de segurança e as camadas adjacentes.

Embora os protocolos SSL e WTLS ofereçam um nível de segurança em seus domínios, WWW e WAP respectivamente, há um problema em potencial onde os dois protocolos conversam: no WAP Gateway.

Isso ocorre porque o protocolo SSL não é diretamente compatível com WTLS. Portanto, quando um pacote de dados criptografados via SSL chega do servidor Web e encontra o WAP Gateway, essa informação é descriptografada. Então esse pacote é novamente criptografada segundo o modelo WTLS e enviado ao cliente.

Como o WAP Gateway geralmente é operado por um terceiro elemento, como uma operadora de telefonia celular, os dados ficam disponíveis por um período de tempo desprotegidos. Então, segundo (CABRAL; LEITE, 2008), a única maneira de garantir segurança total é utilizar SSL (ou TLS) no ambiente da Web, WTLS no ambiente wireless e ter um WAP Gateway protegido contra ataques de terceiros e de dentro da própria companhia.

3.4 Limitações

Os protocolos WAP são desenvolvidos em cima de padrões já existentes para otimizar transferências de dados. Mas isso não impede que haja limitações.

Além da lenta comunicação em dispositivos sem-fio, por volta de 9.600 kbps, destacam-se outras limitações quando estes são comparados a um computador:

- Processadores menos poderosos;
- Quantidade e velocidade de memória (RAM e ROM) menores;
- Consumo de potência restringido devido ao uso de baterias;
- Telas menores.

Ainda sobre as limitações, em celulares, a taxa de transferência cobrada varia de acordo com cada operadora. Dependendo do caso pode ser cobrada por minuto usado ou por KB transferido.

Portanto, quando se pensar em desenvolver uma aplicação WAP, deve-se levar em conta todas as limitações de processamento, memória e custo.

4 *Projeto*

Esse projeto se trata de um Webmail, baseado no sistema Warm Mail de autoria de Luke Welling e Laura Thomson , (WELLING; THOMSON, 2003). O Warm Mail foi projetado para ser utilizado em um computador com um navegador padrão e pode ser configurado para ler mais de uma conta de e-mail, onde as funções oferecidas são:

- Configurar contas de e-mail
- Visualizar conta
- Enviar correio
- Visualizar mensagem
- Responder mensagem
- Encaminhar mensagem
- Excluir mensagem

Um diagrama do fluxo geral do Warm Mail pode ser visto na figura 4.

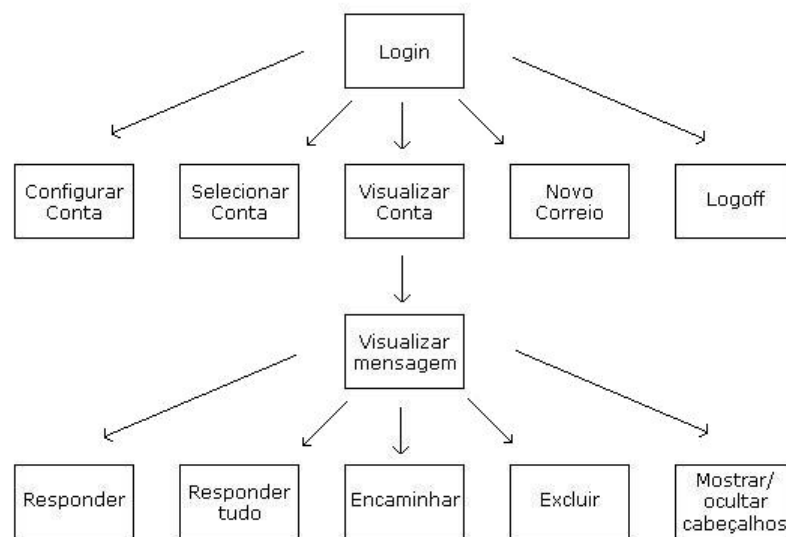


Figura 4: Funcionalidades fornecidas pelo Warm Mail.

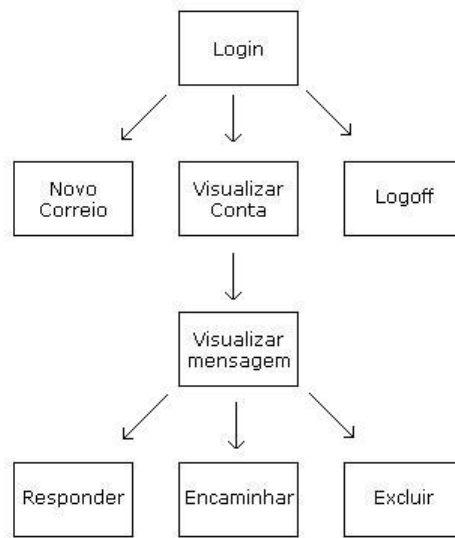


Figura 5: Funcionalidades fornecidas pelo Wap Mail.

As funções fornecidas pelo Webmail modificado deste projeto, que pode ser chamado de Wap Mail, são:

- Visualizar conta
- Enviar correio
- Visualizar mensagem
- Responder mensagem
- Encaminhar mensagem
- Excluir mensagem

A finalidade, desde o início do projeto, foi instalar o sistema nos servidores do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, sendo assim, apenas as contas de e-mail do departamento podem ser acessadas. Portanto a função para configurar contas de e-mail foi removida do projeto. Também foram removidas as funções de mostrar/ocultar cabeçalho e responder a todos.

O novo fluxo geral, feitas as modificações citadas, pode ser visto no diagrama da figura 5.

4.1 Requisitos

Os requisitos expressam as características e restrições do software/sistema no sentido de satisfazer as necessidades dos usuários. Ou seja, são objetivos ou restrições estabelecidas por clientes e usuários do sistema que definem as diversas propriedades do sistema.

Os requisitos descrevem as diversas funcionalidades que os usuários do sistema querem ou precisam que o mesmo atenda e as especificações de que o sistema necessita para o seu funcionamento. São os chamados requisitos funcionais e não-funcionais do sistema, respectivamente.

Um usuário de um sistema de Webmail espera que um software atenda a algumas necessidades, como acessar sua conta de e-mail e poder visualizar suas mensagens, entre outros. Nesse sentido, podemos destacar como requisitos funcionais:

- O sistema deve permitir que um usuário se conecte à sua conta de e-mail do INE;
- O usuário deve poder navegar pelas suas mensagens de correio;
- O sistema deve permitir que o usuário leia uma mensagem selecionada;
- O usuário deve poder apagar uma mensagem selecionada;
- O sistema deve permitir que o usuário responda uma mensagem selecionada;
- O usuário deve poder encaminhar uma mensagem selecionada;
- O sistema deve permitir que o usuário se desconecte da sua conta de e-mail;

Para poder ser utilizado o Wap Mail precisa que alguns requisitos sejam observados. São os requisitos não-funcionais:

- O Wap Mail deve ser hospedado em um servidor Web configurado para executar PHP
- É necessário um dispositivo móvel com acesso à Internet
- O sistema deve ser acessado via um browser WAP

4.2 Configuração

Além das bibliotecas de funções o sistema tem um arquivo de configuração, dentro do diretório *config*, chamado *config.php*. As seguintes variáveis podem ser alteradas na configuração:

- *server*
- *type*
- *port*

Onde:

server: é o endereço do servidor do correio eletrônico. No caso do Departamento de Informática e Estatística é *inf.ufsc.br*.

type: é o tipo do protocolo suportado pelo servidor, podendo ser *pop* ou *imap*.

port: é a porta utilizada pelo servidor de e-mail. Cada protocolo utiliza uma porta diferente, conforme a figura:

Protocolo	Porta
POP3	110
POP3-SSL	995
IMAP	143
IMAP-SSL	993
SMTP	25

Figura 6: Protocolos e respectivas portas

4.3 Bibliotecas

Para desenvolver o Wap Mail são utilizadas funções de saída para o navegador, funções de e-mail para conversar com o servidor de correio e funções de autorização de usuário para identificar se um determinado usuário está conectado ou não. Essas funções estão em arquivos localizados no diretório *fns* dentro da raiz. São as bibliotecas do sistema:

- *output.php*
- *mail.php*
- *user_auth.php*

4.3.1 Output

É a biblioteca que contém as funções que geram as saídas em linguagem WML para serem interpretadas pelo navegador. As funções dessa biblioteca são:

- do_wml_header
- do_wml_footer
- display_button
- display_form_button
- display_message
- display_new_message_form
- display_list
- display_login_form
- display_spacer
- display_toolbar
- format_action
- format_date
- escape
- pretty
- pretty_all

4.3.2 Mail

Essa é a biblioteca responsável por fornecer as funções de comunicação com o servidor de e-mail e manipular as mensagens recebidas e enviadas. São elas:

- add_quoting
- get_account_settings

- `get_list`
- `delete_message`
- `open_mailbox`
- `retrieve_message`
- `send_message`

4.3.3 User_Auth

Essa é uma biblioteca de funções que serve para autenticação. Fornece a função para login no sistema e função que verifica se o usuário ainda está conectado para poder navegar:

- `login`
- `check_auth_user`

4.4 Implementação

O sistema foi programado em PHP utilizando o paradigma estrutural. Essa técnica possibilitou a criação de estruturas simples, usando as subrotinas e as funções para obter o resultado final da ação desejada pelo usuário.

O Wap Mail utiliza as funções das bibliotecas *mail*, *auth_user* e *output* para processar e exibir essas informações requisitadas.

A biblioteca *mail* contém as funções para tratar dos eventos e ações para abrir uma conexão com o servidor de e-mail, obter a lista de mensagens, enviar um correio e apagar uma mensagem.

Na biblioteca *auth_user* estão as funções de autenticação do usuário.

Por fim a biblioteca *output* trata das saídas do sistema, implementadas em linguagem WML para serem interpretadas pelos browsers WAP.

No diretório raiz do sistema há dois tipos de arquivos: os arquivos que geram saídas WML para visualização do conteúdo e os arquivos com ações que, depois de realizadas, levam a alguma página que gera a saída WML.

As páginas que geram saídas WML diretamente são:

- index.php
- new-message.php
- view-mailbox.php
- view-message.php
- forward.php
- reply.php

Os arquivos que realizam alguma ação, e depois são direcionados para páginas com saídas WML são:

- login.php
- logout.php
- delete-message.php
- send-message.php

Além dos já citados, o sistema utiliza um arquivo para importar as configurações e bibliotecas necessárias à aplicação:

Código Fonte 4.1: include.php

```
1 <?php
2
3     include_once( './config/config.php' );
4
5     include_once( './fns/output.php' );
6     include_once( './fns/mail.php' );
7     include_once( './fns/user_auth.php' );
8
9 ?>
```

A página principal do sistema, `index.php`, é responsável por verificar se um usuário está conectado ou não. É também essa página que vai receber as respostas das ações, como login, logout, apagar mensagem e enviar uma mensagem.

Código Fonte 4.2: `index.php`

```
1 <?php
2
3     session_start ();
4
5     include ('include.php');
6
7     header("Content-type: text/vnd.wap.wml");
8
9     $status = '';
10    if( isset( $_GET['status'] ) )
11        $status = $_GET['status'];
12
13    $buttons = array ();
14
15    if(check_auth_user())
16    {
17        $buttons[0] = 'view-mailbox';
18        $buttons[1] = 'new-message';
19        $buttons[2] = 'log-out';
20    }
21
22    do_wml_header("WAP Mail", $status);
23
24    $nButtons = sizeof( $buttons );
25    display_toolbar($buttons, '', $nButtons);
26
27    if(!check_auth_user())
28    {
29        display_login_form ();
30    }
31
32    do_wml_footer ();
33
34    ?>
```

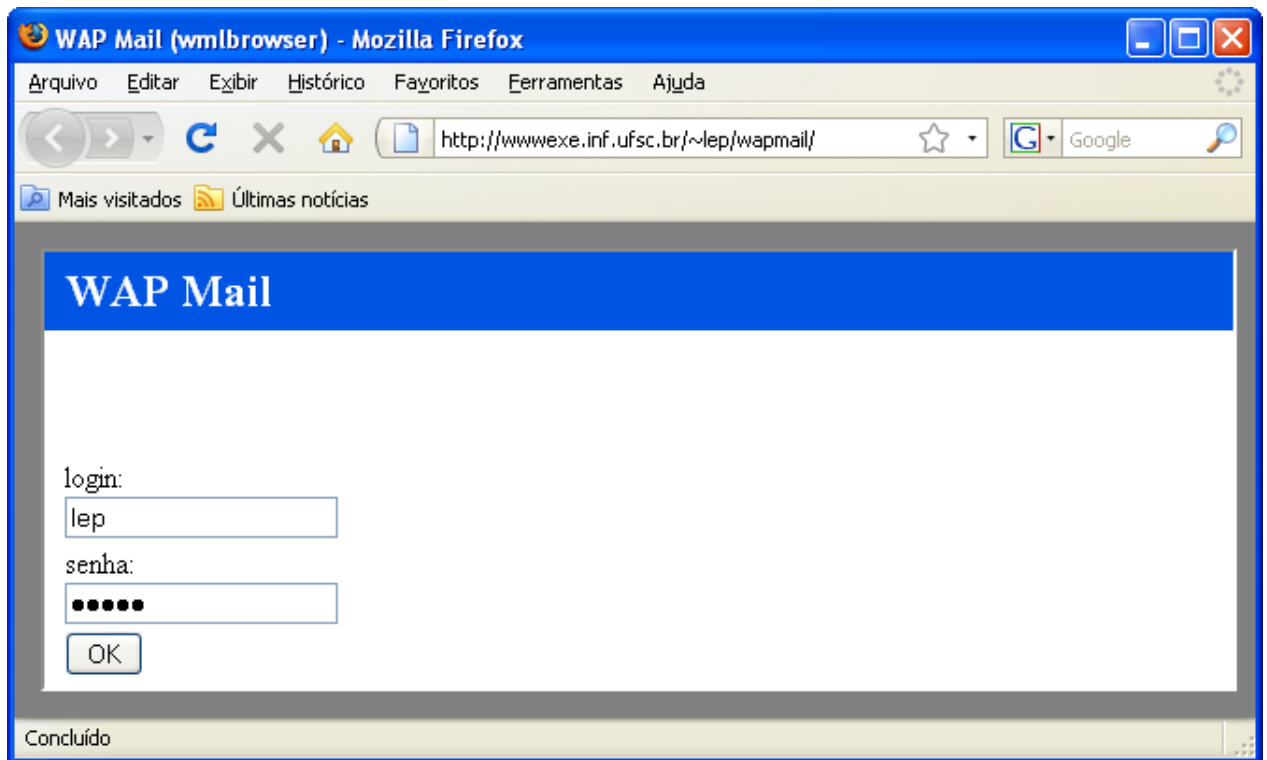


Figura 7: Sistema no estado inicial.

A figura mostra a primeira página do sistema em seu estado inicial. Ou seja, é o primeiro acesso e o usuário ainda não efetuou login. Essa autenticação é verificada com a função *check_auth_user*, da biblioteca *Auth_User*, que foi chamada nas linhas 15 e 27.

Como não há um usuário conectado, um formulário para login é exibido, através da chamada da função *display_login_form*, da biblioteca *Output*, na linha 29.

Para a visualização do sistema, nesse e em todos os screenshots o navegador utilizado para exibir as páginas WML foi o Firefox com a extensão *wmlbrowser* instalada.

O formulário da página inicial leva ao arquivo login.php que tenta efetuar a conexão com o servidor de e-mail do usuário com as credenciais fornecidas. Depois de realizar a ação a página inicial é chamada novamente.

Código Fonte 4.3: login.php

```
1 <?php
2
3 session_start();
4
5 include ('include.php');
6
7 $username = '';
8 $passwd = '';
9
10 if( isset( $_POST['username'] ) )
11     $username = $_POST['username'];
12
13 if( isset( $_POST['passwd'] ) )
14     $passwd = $_POST['passwd'];
15
16 if($username != '' || $passwd != '')
17 {
18     if(login($username, $passwd))
19     {
20         $_SESSION['auth_user'] = $username;
21         $_SESSION['password'] = $passwd;
22     }
23     else
24     {
25         $status = 'Erro ao efetuar login.';
26     }
27 }
28
29 $host = $_SERVER['HTTP_HOST'];
30 $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\ ');
31 $extra = 'index.php?status='.$status;
32 header("Location: http://$host$uri/$extra");
33 exit;
34
35 ?>
```

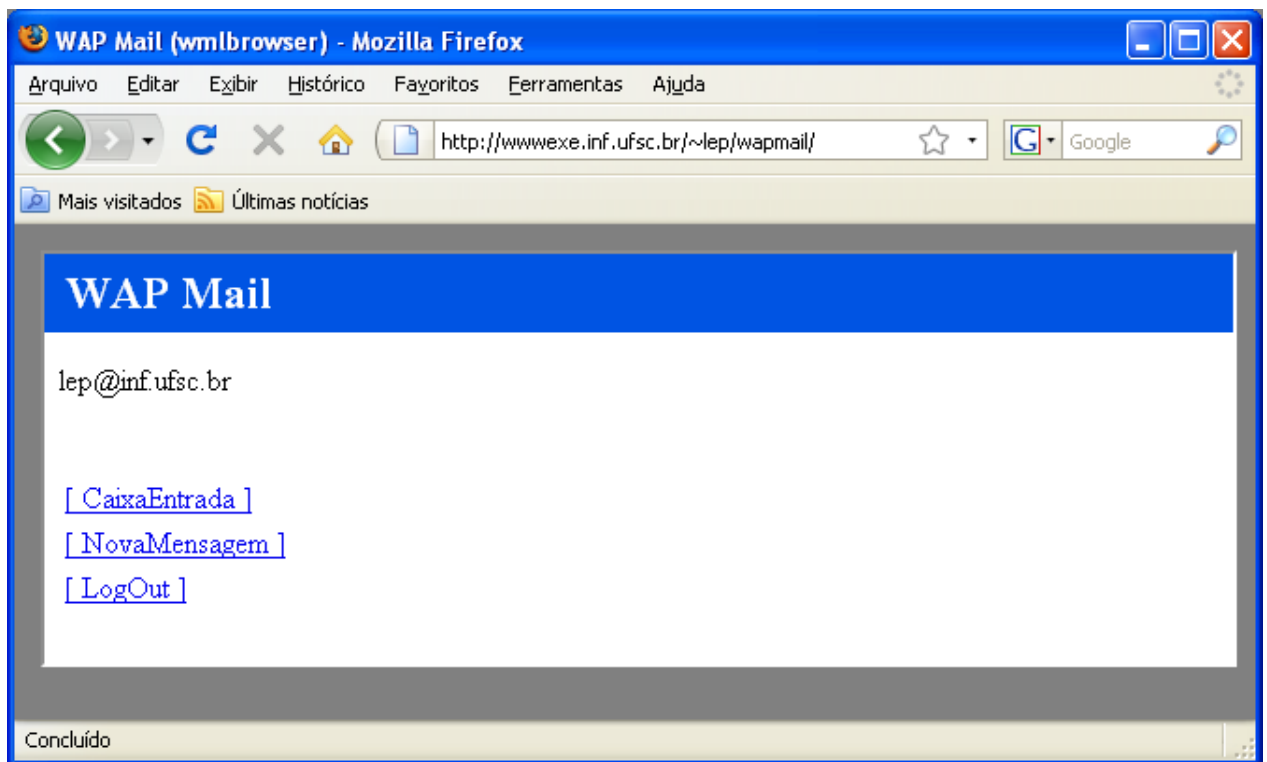


Figura 8: Sistema depois de efetuado login.

Depois de efetuado o login, seguindo o fluxograma definido na figura 4, o usuário agora está conectado ao sistema e pode abrir sua Caixa de Entrada, escrever uma Nova Mensagem ou, ainda, desconectar do sistema efetuando logout, como mostra a figura acima.

Se o usuário optar por desconectar do sistema ele será levado à página log-out.php, e esta, depois de realizar a ação requisitada, volta à página inicial da aplicação.

Código Fonte 4.4: log-out.php

```
1 <?php
2
3     session_start ();
4
5     include ('include.php');
6
7     session_destroy ();
8
9     $status = 'Voce efetuou logout.';
10
11     $host  = $_SERVER['HTTP_HOST'];
12     $uri   = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
13     $extra = 'index.php?status='.$status;
14     header("Location: http://$host$uri/$extra");
15     exit;
16
17 ?>
```

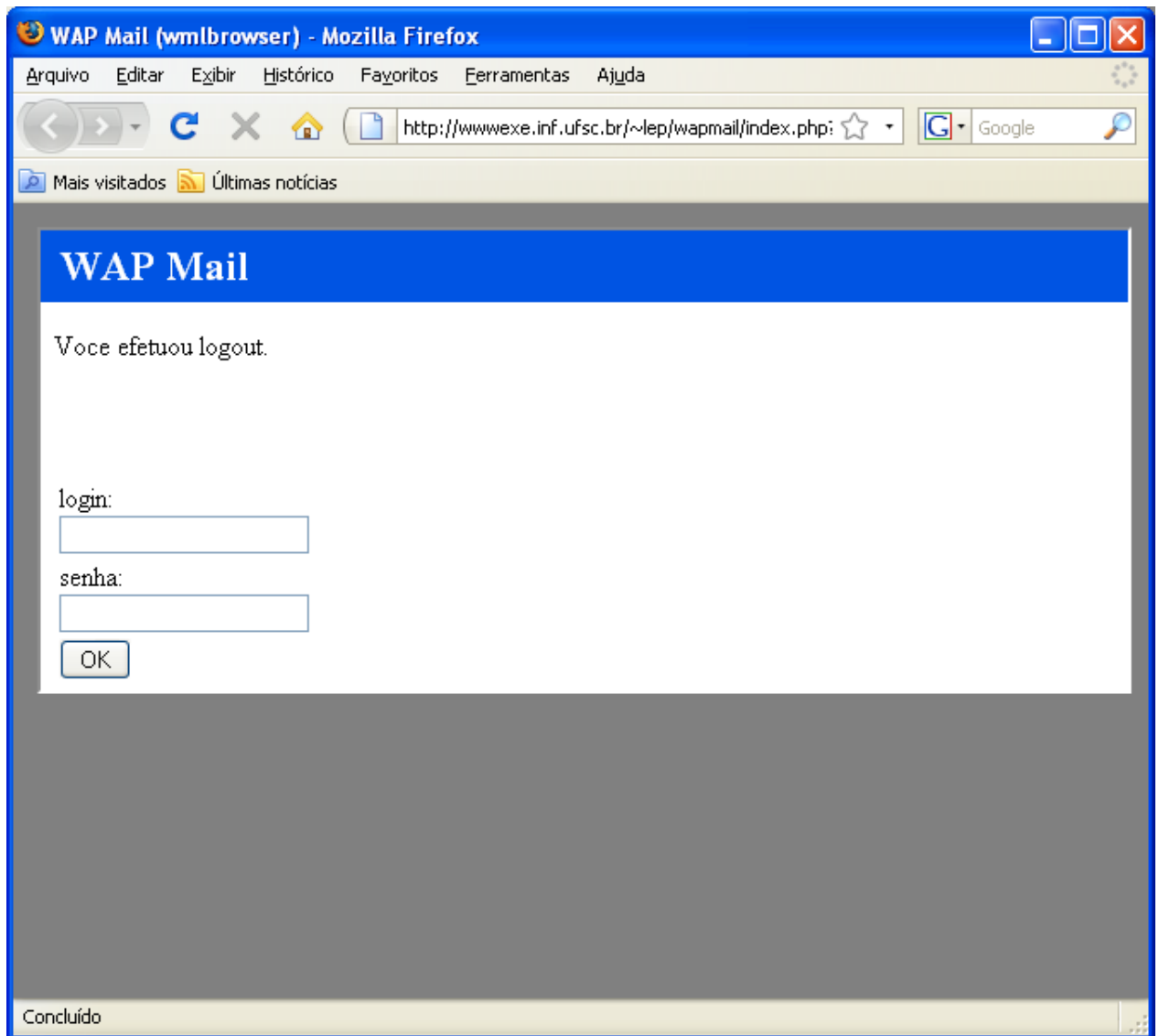


Figura 9: Sistema depois de efetuado log-out.

De volta ao fluxo inicial, se o usuário escolher enviar uma Nova Mensagem, será chamada a página new-message.php.

Código Fonte 4.5: new-message.php

```
1 <?php
2
3 session_start();
4
5 include ('include.php');
6
7 header("Content-type: text/vnd.wap.wml");
8
9 $to = '';
10 if( isset( $_GET['to'] ) )
11     $to = $_GET['to'];
12
13 $cc = '';
14 if( isset( $_GET['cc'] ) )
15     $cc = $_GET['cc'];
16
17 $subject = '';
18 if( isset( $_GET['subject'] ) )
19     $subject = $_GET['subject'];
20
21 $message = '';
22 if( isset( $_GET['message'] ) )
23     $message = $_GET['message'];
24
25 $buttons = array();
26 if(check_auth_user())
27 {
28     $buttons[0] = 'view-mailbox';
29     $buttons[1] = 'new-message';
30     $buttons[2] = 'log-out';
31 }
32
33 do_wml_header("WAP Mail", $status);
34
35 $nButtons = sizeof( $buttons );
36 display_toolbar($buttons, '', $nButtons);
37
38 if(!check_auth_user())
39 {
```

```
40     display_login_form($action);
41 }
42 else
43 {
44     display_new_message_form($to, $cc, $subject, $body, $append);
45 }
46
47 do_wml_footer();
48
49 ?>
```

Figura 10: Formulário para enviar nova mensagem.

Esse formulário envia uma nova mensagem, chamando a ação da página `send-message.php`. Se enviar com sucesso retorna à página inicial uma mensagem confirmando o envio. Caso contrário retorna uma mensagem de erro que será exibida na página inicial.

Código Fonte 4.6: `send-message.php`

```

1 <?php
2
3     session_start ();
4
5     include ( 'include.php' );
6
7     $to = '';
8     if( isset( $_POST[ 'to' ] ) )

```

```
9      $to = $_POST['to'];
10
11     $cc = '';
12     if( isset( $_POST['cc'] ) )
13         $cc = $_POST['cc'];
14
15     $subject = '';
16     if( isset( $_POST['subject'] ) )
17         $subject = $_POST['subject'];
18
19     $message = '';
20     if( isset( $_POST['message'] ) )
21         $message = $_POST['message'];
22
23     $append = '';
24     if( isset( $_POST['append'] ) )
25         $append = $_POST['append'];
26
27     if( $append != '' )
28     {
29         $append = str_replace("> ", "\n> ",imap_qprint($append));
30         $message .= "\n\n".$append;
31     }
32
33     if( check_auth_user() )
34     {
35         if( send_message($to, $cc, $subject, $message) )
36             $status = 'A mensagem foi enviada.';
37         else
38             $status = 'Erro ao enviar a mensagem.';
39     }
40
41     $host = $_SERVER['HTTP_HOST'];
42     $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\ ');
43     $extra = 'index.php?status='.$status;
44     header("Location: http://$host$uri/$extra");
45     exit;
46
47 ?>
```

Novamente no fluxo inicial, se o usuário escolher abrir sua Caixa de Entrada ele visualizará uma lista com as mensagens extraídas do servidor de e-mail.

Código Fonte 4.7: view-mailbox.php

```
1 <?php
2
3     session_start ();
4
5     include ('include.php');
6
7     header("Content-type: text/vnd.wap.wml");
8
9     $buttons = array();
10
11     if(check_auth_user())
12     {
13         $buttons[0] = 'view-mailbox';
14         $buttons[1] = 'new-message';
15         $buttons[2] = 'log-out';
16     }
17
18     do_wml_header("WAP Mail", $status);
19
20     $nButtons = sizeof( $buttons );
21     display_toolbar($buttons, '', $nButtons);
22
23     if(!check_auth_user())
24     {
25         display_login_form($action);
26     }
27     else
28     {
29         if( $_GET['pg'] )
30             $pg = $_GET['pg'];
31         else
32             $pg = 1;
33
34         display_list($pg);
35     }
36
37     do_wml_footer();
38
39 ?>
```

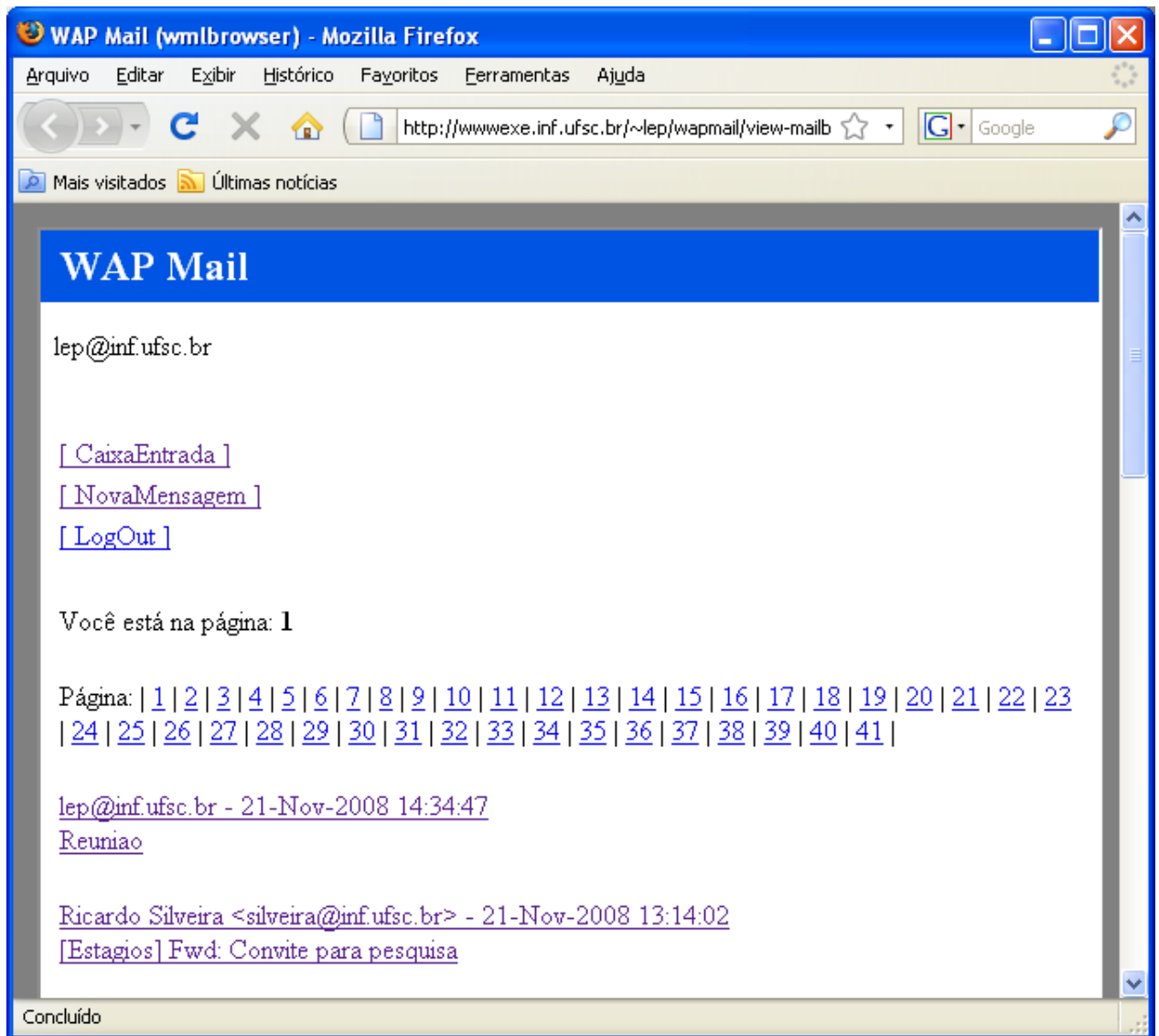


Figura 11: Caixa de Entrada com a lista de mensagens do servidor.

Quando selecionada uma mensagem, o seu conteúdo é exibido. Além do conteúdo da mensagem, três novos botões ficam disponíveis para o usuário: Responder, Encaminhar e Apagar.

Código Fonte 4.8: view-message.php

```
1 <?php
2
3 session_start();
4
5 include ('include.php');
6
7 header("Content-type: text/vnd.wap.wml");
8
9 $status = '';
10
11 if( isset( $_GET['status'] ) )
12     $status = $_GET['status'];
13
14 $buttons = array();
15 if(check_auth_user())
16 {
17     $buttons[0] = 'view-mailbox';
18     $buttons[1] = 'new-message';
19     $buttons[2] = 'log-out';
20 }
21
22 do_wml_header("WAP Mail", $status);
23
24 $nButtons = sizeof( $buttons );
25 display_toolbar($buttons, '', $nButtons);
26
27 if(!check_auth_user())
28 {
29     display_login_form($action);
30 }
31 else
32 {
33     if( isset( $_GET['messageid'] ) )
34     {
35         $messageid = $_GET['messageid'];
36         display_message($messageid);
37     }
38 }
```

39

40 `do_wml_footer();`

41

42 `?>`

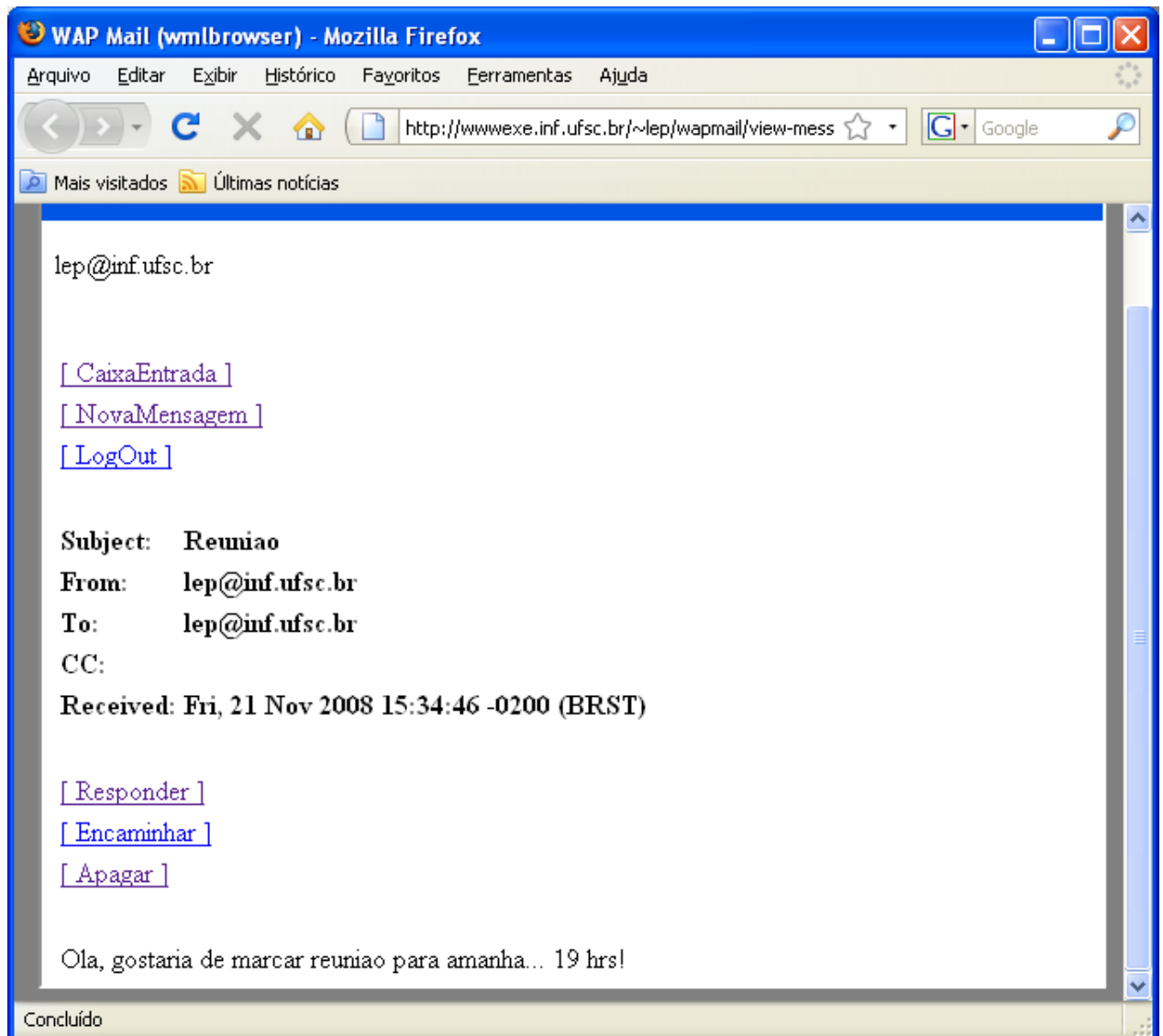


Figura 12: Mensagem sendo exibida.

Se o usuário optar por responder a uma mensagem, ele é levado a um formulário preenchido com os dados da mensagem anteriormente selecionada, como endereço remetente, assunto e corpo.

Código Fonte 4.9: reply.php

```
1 <?php
2
3  session_start ();
4
5  include ( 'include.php' );
6
7  header("Content-type: text/vnd.wap.wml");
8
9  $buttons = array ();
10 if( check_auth_user () )
11 {
12     $buttons [0] = 'view-mailbox';
13     $buttons [1] = 'new-message';
14     $buttons [2] = 'log-out';
15 }
16
17 do_wml_header("WAP Mail", $status);
18
19 $nButtons = sizeof( $buttons );
20 display_toolbar($buttons, '', $nButtons);
21
22 if(!check_auth_user ())
23 {
24     display_login_form($action);
25 }
26 else
27 {
28     if( isset( $_GET['messageid'] ) )
29     {
30         $messageid = $_GET['messageid'];
31         if (!$imap)
32             $imap = open_mailbox ();
33
34         if ($imap)
35         {
36             $header = imap_header($imap, $messageid);
37             if ($header->reply_toaddress)
38                 $to = htmlspecialchars($header->reply_toaddress);
```

```
39     else
40         $to = $header->fromaddress;
41
42         $subject = htmlspecialchars('Re: ' . $header->subject);
43         $append = stripslashes(imap_body($imap, $messageid));
44         imap_close($imap);
45
46         display_new_message_form($to, $cc, $subject, $body, $append);
47     }
48 }
49 }
50
51 do_wml_footer();
52
53 ?>
```

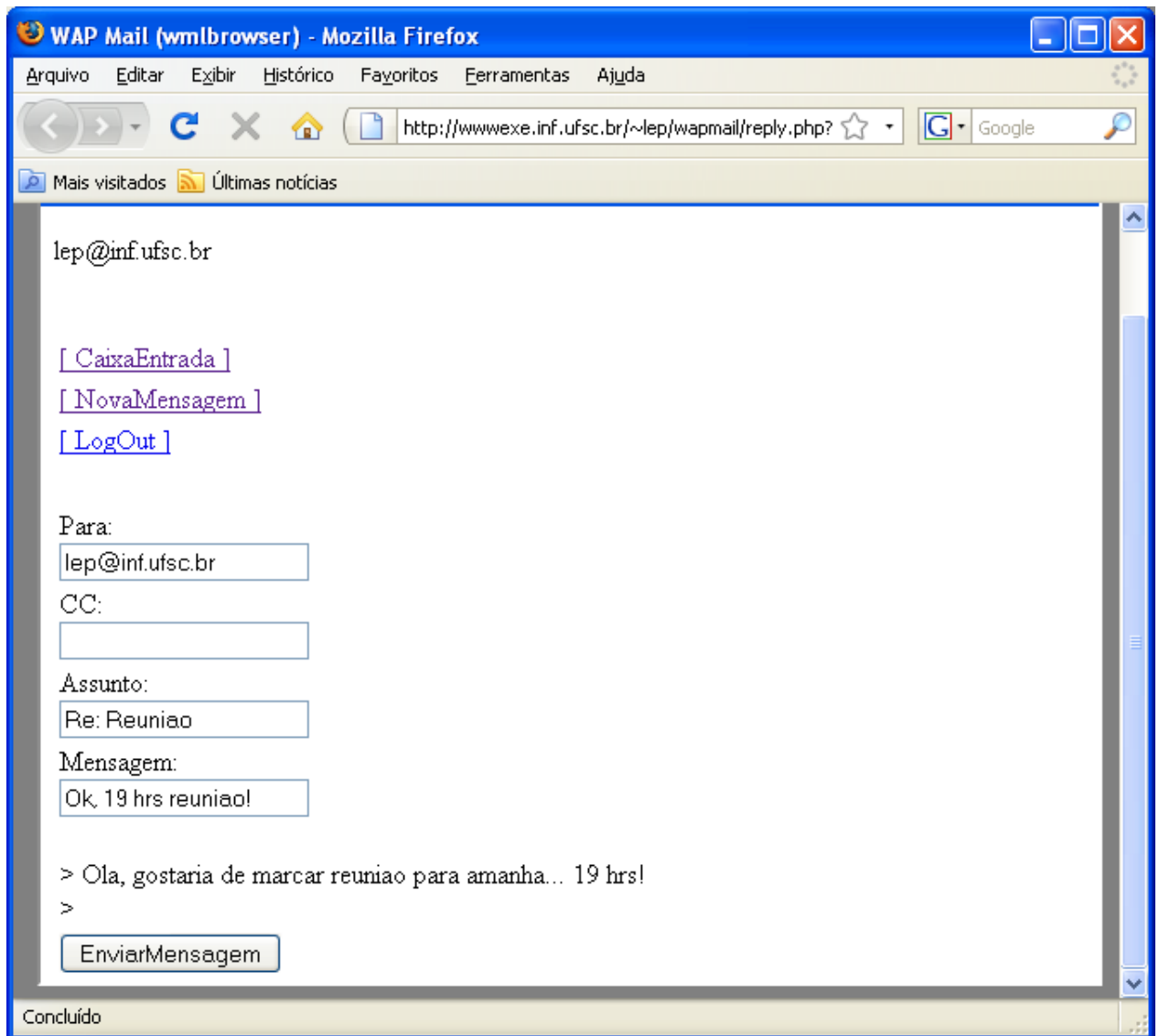


Figura 13: Respondendo a uma mensagem.

Já se o usuário optar por encaminhar a mensagem, ele é levado a um outro formulário preenchido com os dados da mensagem anteriormente selecionada, como endereço remetente, assunto e corpo.

Código Fonte 4.10: forward.php

```
1 <?php
2
3     session_start ();
4
5     include ('include.php');
6
7     header("Content-type: text/vnd.wap.wml");
8
9     $buttons = array ();
10
11     if(check_auth_user())
12     {
13         $buttons [0] = 'view-mailbox';
14         $buttons [1] = 'new-message';
15         $buttons [2] = 'log-out';
16     }
17
18     do_wml_header("WAP Mail", $status);
19
20     $nButtons = sizeof( $buttons );
21     display_toolbar($buttons, '', $nButtons);
22
23     if(!check_auth_user())
24     {
25         display_login_form($action);
26     }
27     else
28     {
29         if( isset( $_GET['messageid'] ) )
30         {
31             $messageid = $_GET['messageid'];
32
33             if(!$imap)
34                 $imap = open_mailbox();
35
36             if($imap)
37             {
38                 $header = imap_header($imap, $messageid);
```

```
39
40     $subject = htmlspecialchars('Fwd: '. $header->subject);
41     $append = stripslashes(imap_body($imap, $messageid));
42     imap_close($imap);
43
44     display_new_message_form($to, $cc, $subject, $body, $append);
45     }
46 }
47 }
48
49 do_wml_footer();
50
51 ?>
```

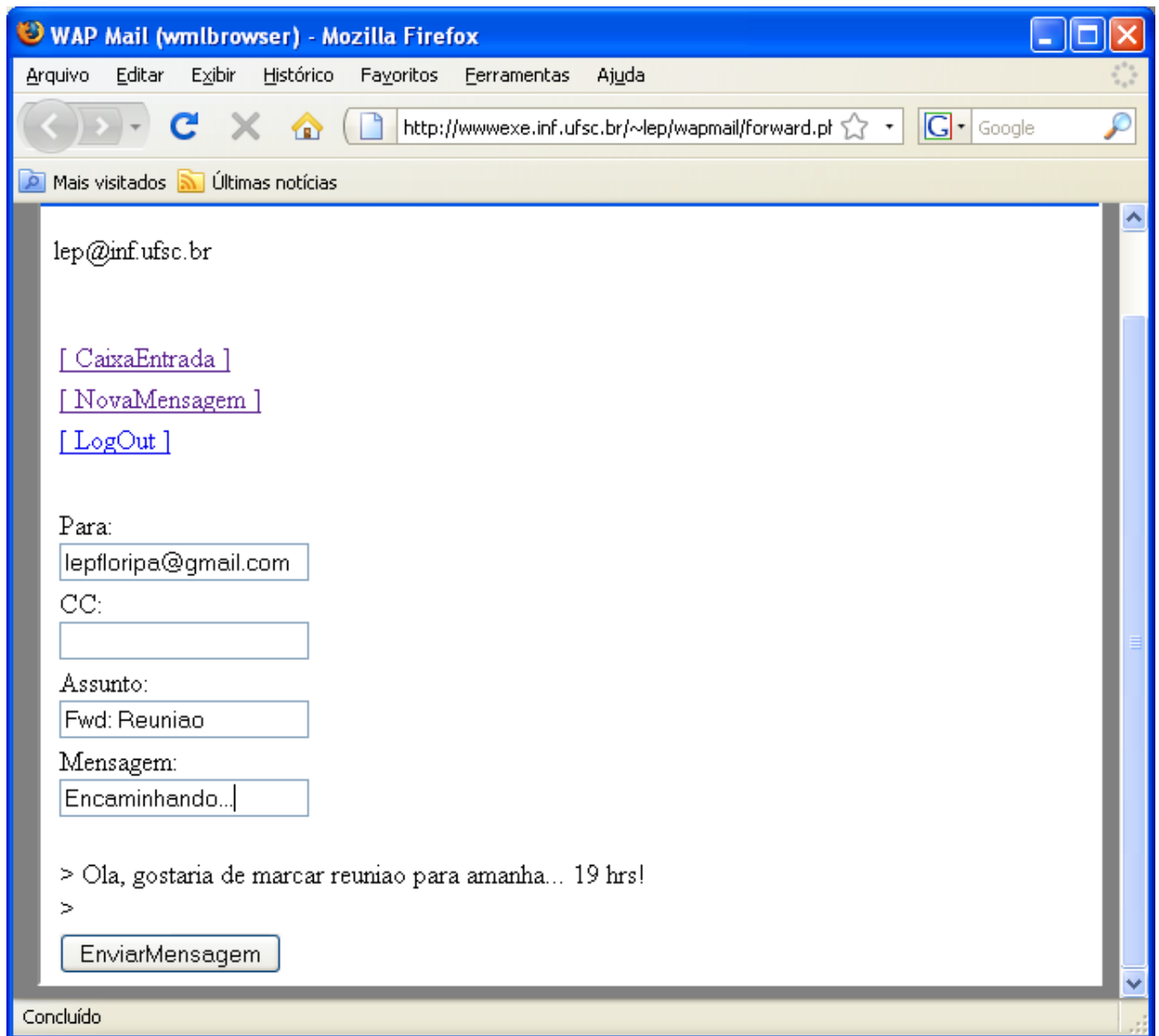


Figura 14: Encaminhando uma mensagem.

Em ambos os casos, seja responder ou encaminhar uma mensagem, os dados são enviados para a página `send-message.php` que envia mensagens conforme visto anteriormente.

Por fim, se o usuário desejar apagar a mensagem, essa requisição é enviada para o arquivo delete-message.php.

Código Fonte 4.11: delete-message.php

```
1 <?php
2
3     session_start ();
4
5     include ( 'include.php' );
6
7     if( check_auth_user() )
8     {
9         if( isset( $_GET['messageid'] ) )
10        {
11            $messageid = $_GET['messageid'];
12            if( delete_message($messageid) )
13                $status = 'A mensagem foi apagada.';
14            else
15                $status = 'Erro ao apagar a mensagem.';
16        }
17    }
18
19    $host  = $_SERVER['HTTP_HOST'];
20    $uri   = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
21    $extra = 'index.php?status='.$status;
22    header("Location: http://$host$uri/$extra");
23    exit;
24
25 ?>
```

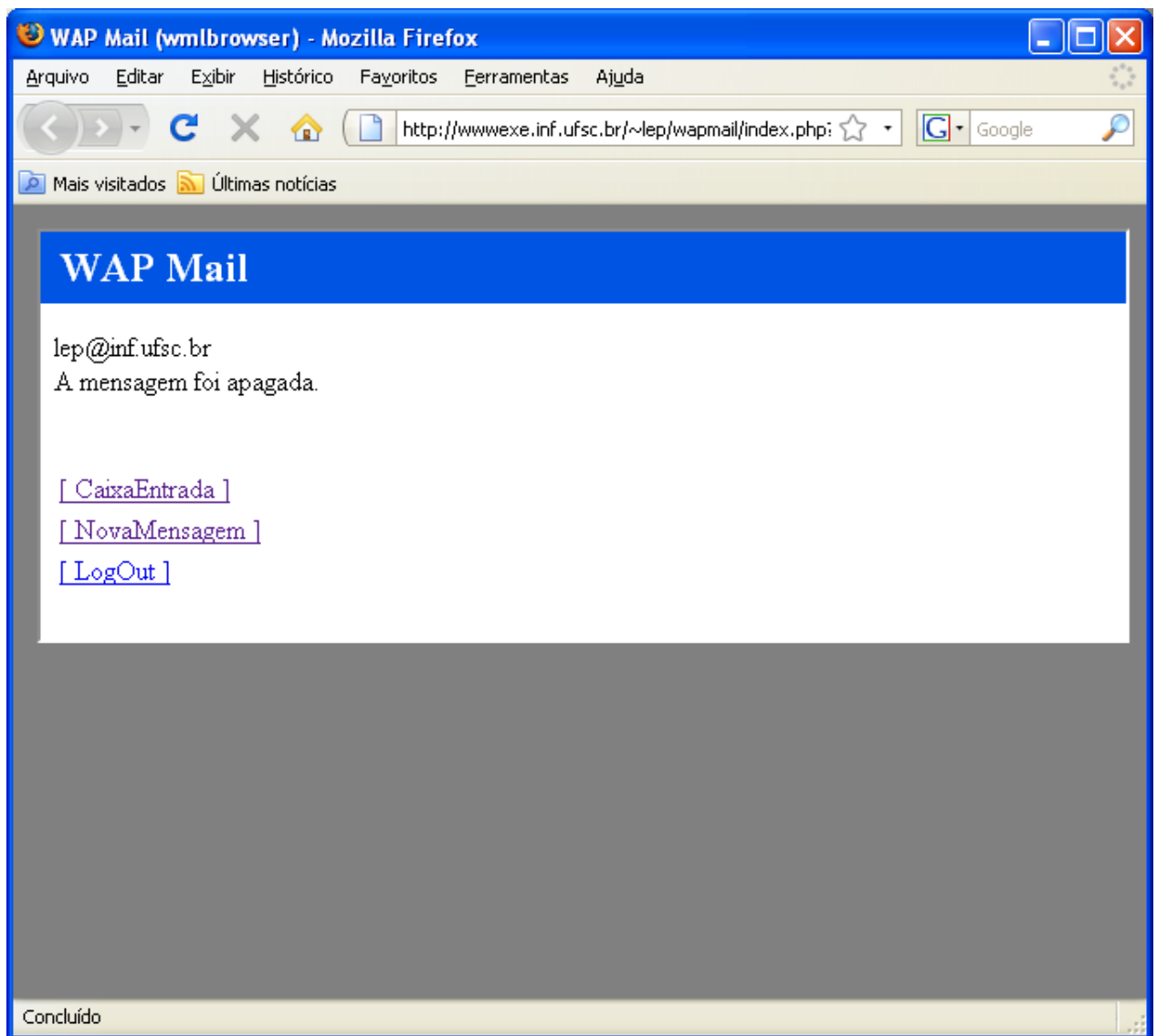



Figura 15: Apagando uma mensagem.

A página delete-message.php apaga a mensagem do servidor de e-mail e retorna à página inicial do sistema informando se a ação foi bem sucedida ou não.

4.5 Implantação

O sistema foi implantado no servidor <http://wwwexe.inf.ufsc.br/> do INE que é disponível para todos os alunos e professores do departamento. É nesse endereço que está instalado o servidor Web Apache configurado para executar PHP.

Cada aluno e professor do departamento tem, portanto, uma conta que serve para utilizar os serviços oferecidos. Entre eles, destacam-se as contas de e-mail e os espaços em disco disponíveis no servidor Web. Acessando <http://wwwexe.inf.ufsc.br/~lep/> por exemplo, chega-se à página dinâmica do aluno com a conta *lep*.

O Wap Mail foi instalado em <http://wwwexe.inf.ufsc.br/~lep/wapmail/> e acessando esse endereço através de um dispositivo móvel, é possível que um usuário do INE se conecte à sua conta de e-mail e visualize, escreva ou apague as suas mensagens eletrônicas.

4.6 Usabilidade

A interface de usuário é uma parte fundamental em um sistema Web, pois é a responsável direta em fazer com que o usuário consiga realizar suas tarefas de maneira fácil, rápida e satisfatória. Em ambientes móveis, uma boa usabilidade se torna ainda mais imprescindível.

Em (RUIZ, 2008), José Julio Ruiz traz um guia para o correto design de websites para PDAs, que pode ser estendido para qualquer dispositivo sem-fio. Abaixo, seguem as análises, item por item, de cada heurística levantada pelo autor e adaptadas para dispositivos móveis em geral:

1. Ser consciente das limitações dos dispositivos

Como os dispositivos em questão têm uma tela reduzida, menos memória e menos capacidade de processamento que os computadores convencionais, essas limitações têm que ser prioritárias na concepção do site WAP.

2. Definir cuidadosamente a estrutura do site

O site tem que ser objetivo, evitando telas com boas-vindas, apresentações, seleção de preferências, etc, visto que a tarifação ainda é um grande problema na transferência WAP. O fluxograma definido para o Wap Mail prevê poucas mudanças de estado para se chegar à ação pretendida, portanto, o sistema cumpre esse quesito.

3. Evitar o uso de tabelas

No guia, o autor sugere que não sejam utilizadas tabelas. Se for necessário utilizá-las, não especificar a largura em tamanhos fixos para evitar rolamento horizontal. No Wap Mail as tabelas são usadas para separar botões e conteúdo do webmail, e nenhuma tem tamanho fixo determinado, sendo adequadas aos diferentes tamanhos de telas.

4. Não utilizar molduras (frames)

Como o site foi desenvolvido em WML, não há frames, pois a especificação da linguagem não fornece essa tag.

5. Publicar conteúdos concisos

O artigo diz que deve-se escolher somente a informação essencial para exibir ao usuário, devido à limitação de memória e de tamanho de tela. O sistema cumpre esse quesito já que fornece botões e conteúdo pertinentes ao estado atual do usuário.

6. Organizar a informação cuidadosamente

Distribuir de maneira ótima os links de navegação do site é uma maneira de organizar a informação. Os botões do Wap Mail estão todos já à primeira vista do usuário, e seu conteúdo logo abaixo, evitando rolagem desnecessária da barra de navegação. Como o conteúdo do webmail é dinâmico, variando de mensagem a mensagem, pode ocorrer uma maximização da longitude do texto, sendo necessária mais de uma tela de rolagem para visualização. Nesse sentido, esse quesito é parcialmente coberto pelo sistema.

7. Otimizar os gráficos

Levar em consideração as dimensões de tela e a paleta de cores disponíveis nos mais variados dispositivos. Nesse sentido, o Wap Mail não utiliza gráficos e cores, exibindo toda a informação necessária em modo texto.

8. Incluir texto alternativo em todas as imagens

O Wap Mail não utiliza imagens para um carregamento mais rápido de suas páginas.

9. Adicionar a tag handheldfriendly ao começo de todas as páginas

Em páginas escritas em HTML para dispositivos móveis, a adição dessa tag é importante para informar que o conteúdo seguinte está otimizado para dispositivos móveis. Utilizando WML, e o cabeçalho correto, essa tag é obsoleta.

10. Utilizar convenientemente o caching das páginas

Não salvar em cache as páginas que se atualizam diariamente ou com alguma frequência aumenta a velocidade do carregamento, já que o micro-browser não precisará verificar se houve mudança de conteúdo ou não. No Wap Mail não há salvamento de cache, ou seja, o browser sempre interpretará as páginas como se fosse o primeiro acesso.

Pode-se dizer, então, que o Wap Mail apresenta uma usabilidade que cumpre em 90% os quesitos levantados pelo guia, conforme o gráfico abaixo:

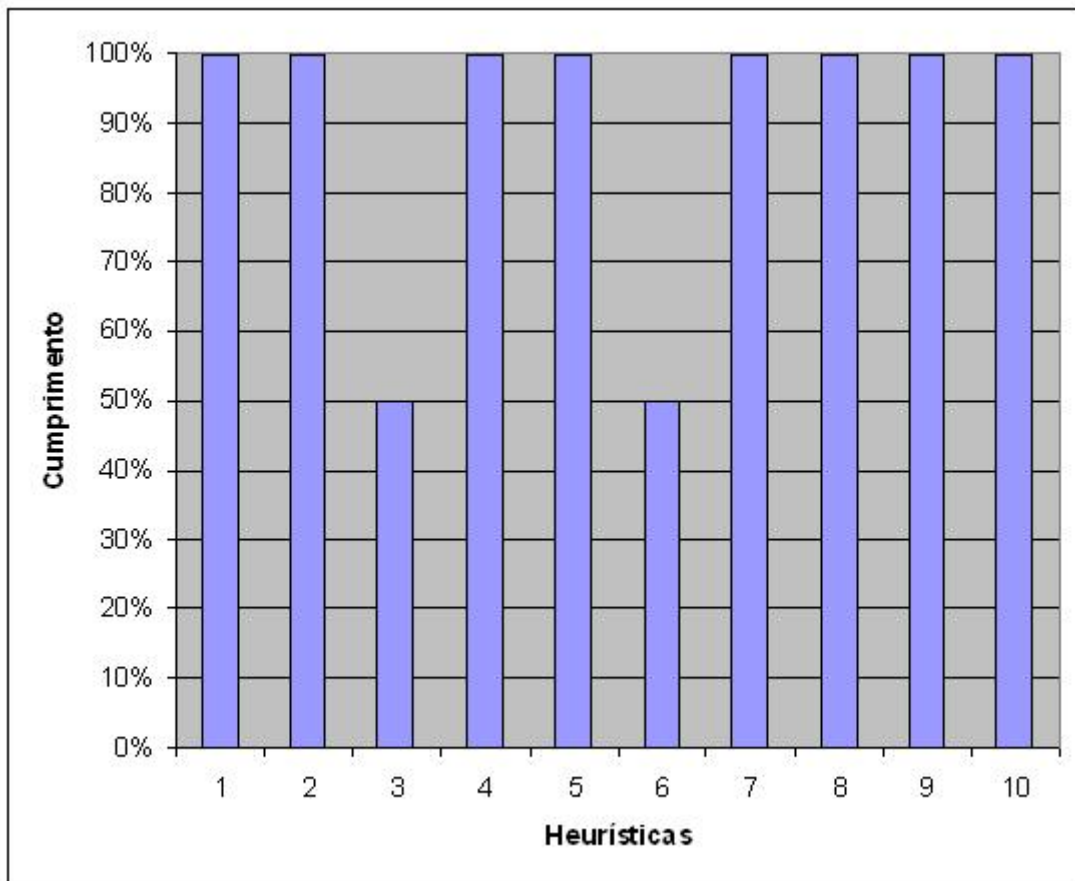


Figura 16: Usabilidade do Wap Mail.

5 *Conclusão*

Neste trabalho foi projetado um webmail para dispositivos móveis que fornece as principais funcionalidades que um sistema como esse deve oferecer. Essas funcionalidades são: conexão a uma conta de e-mail, ler um correio, responder e encaminhar mensagens, e excluir um determinado correio.

O trabalho foi desenvolvido utilizando as linguagens de programação PHP e WML. A primeira para ser executada no lado do servidor e fazer todo o processamento necessário para, então, fornecer uma interface ao usuário do dispositivo móvel, usando a segunda linguagem, no lado do cliente.

Utilizando essas ferramentas foi possível implementar as funcionalidades básicas de um webmail propostas neste trabalho.

Esse webmail não pretende substituir os já instalados nos servidores do Departamento de Informática e Estatística, mas complementar os mesmos. Sendo assim, esse trabalho tem como intenção ser uma alternativa, via dispositivo móvel, para suprir a necessidade de se acessar as contas de e-mail, visualizar as mensagens e respondê-las ou encaminhá-las, bem como eventualmente apagar algum correio.

Algumas alterações podem ser previstas para um futuro trabalho. Uma implementação futura poderia ser utilizar um banco de dados para guardar preferências dos usuários, como: agenda, assinatura, número de e-mails exibidos por páginas, entre outros.

Referências

ARAÚJO, J. *Desenvolvendo para WAP com WML*. 1. ed. [S.l.]: Editora Ciência Moderna, 2001.

CABRAL, J. L. M.; LEITE, L. M. *Segurança em Transações e Aplicações WAP*. [S.l.], nov. 2008. Disponível em: <http://www.wirelessbrasil.org/wirelessbr/colaboradores/cabral_leite/seg_wap_01.html>.

RUIZ, J. J. *10 puntos de usabilidad y diseño web para PDAs*. [S.l.], nov. 2008. Disponível em: <http://www.pdaexpertos.com/Tutoriales/Internet_movil/10_puntos_de_usabilidad_y_diseno_web_para_PDAs.shtml>.

WELLING, L.; THOMSON, L. *PHP and MySQL - Web Development*. 2. ed. [S.l.]: Sams Publishing, 2003.

WILEY, D. *Correio Eletrônico*. [S.l.], nov. 2008. Disponível em: <http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch11.html>.