UNIVERSIDADE DE SANTO AMARO

SISTEMA DE INFORMAÇÃO

ADMINISTRAÇÃO DE SERVIDORES VIA DISPOSITIVOS MÓVEIS

BRENO HENRIQUE DUARTE DE OLIVEIRA

LUCAS PEREIRA DA ROCHA

SÃO PAULO

2007

UNIVERSIDADE DE SANTO AMARO

SISTEMA DE INFORMAÇÃO

ADMINISTRAÇÃO DE SERVIDORES VIA DISPOSITIVOS MÓVEIS

BRENO HENRIQUE DUARTE DE OLIVEIRA

LUCAS PEREIRA DA ROCHA

“Trabalho de Conclusão de Curso apresentado para o titulo de bacharel em Sistema de Informação na Universidade de Santo Amaro, sob a orientação do professor Edílson Osório Junior. “

SÃO PAULO

2007

ADMINISTRAÇÃO DE SERVIDORES

VIA DISPOSITIVOS MÓVEIS

BRENO HENRIQUE DUARTE DE OLIVEIRA

LUCAS PEREIRA DA ROCHA

DATA __ /__/__

BANCA EXAMINADORA

_____

_____

_____

_____

CONCEITO FINAL: _____

DEDICATÓRIA

Este trabalho de conclusão de curso é dedicado a todos os educadores que têm a coragem de levar a Educação no Brasil.

O valor das coisas não está no tempo que elas duram, mas na intensidade com que acontecem. Por isso existem momentos inesquecíveis, coisas inexplicáveis e pessoas incomparáveis.

FERNANDO PESSOA

AGRADECIMENTOS

Gostaríamos de agradecer todos nossos familiares, professores do corpo acadêmico da Universidade de Santo Amaro pela paciência que tiveram e nos instruíram para chegar a conclusão desse trabalho.

SUMÁRIO

SUMÁRIO DE IMAGENS

RESUMO

O presente trabalho de conclusão de curso apresenta um sistema de administração de servidores via dispositivos móveis, implementando em PHP 5 com MySQL 5. O usuário tem o status do serviço, em forma de cores de semáforos, tornando o aplicativo intuitivo e de rápida percepção na visualização dos serviços.

ABSTRACT

The Objective of the current course conclusion work is to demonstrate a Serve Administration System that via mobile devices, implementing on PHP 5 with MySQL5. Which this system, users obtain the service status, displayed in color like traffic-lights, resulting in an intuitive applicative and a faster perception in order to visualize the service.

INTRODUÇÃO

Com a evolução dos dispositivos móveis sua capacidade de processamento de dados e armazenamento de informações é cada vez maior.

Sendo assim, será implementado um aplicativo para a administração de servidores através de dispositivos móveis que tenham acesso a internet.

Nos sistemas existentes, há muitos painéis administrativos robustos que permitem a administração quase que completa do servidor. Porém todos esses painéis são complexos e pesados demais para uso a partir de qualquer dispositivo móvel.

Objetivo

O objetivo deste trabalho é desenvolver um aplicativo para a administração de servidores através de qualquer dispositivo móvel, com uma interface rica, agradável e simples para administração de servidores com acesso através de qualquer dispositivo móvel que possa se conectar na internet.

O software irá realizar operações como:

- Parar, Iniciar e verificar estado de processos. Serviços como E-mail, Proxy, SSH, Bind, MySQL e outros;
- Verificar estado da memória;
- Verificar estado do Espaço em disco;
- Verificar o processamento do servidor.

Dessa forma, o administrador terá uma melhor visão do funcionamento do sistema.

Com acesso a algumas funções do servidor, através do aplicativo o administrador poderá resolver algumas falhas que esteja ocorrendo no servidor no exato momento. Para isso basta ele ter um dispositivo móvel com acesso a internet.

CAPÍTULO 1

EVOLUÇÃO DOS COMPUTADORES

A evolução dos computadores, conseqüentemente do hardware começou definitivamente com os computadores com válvulas, o mais conhecido foi ENIAC (Electronic Numeric Integrator and Calculator), criado em 1946 por John Mauchly e J. Presper Eckert, na Universidade da Pensilvânia utilizado para fazer cálculos matemáticos, que atingia o surpreendente calculo de cinco mil adições ou subtrações por segundo, para época era uma grande invenção, o ENIC pesava 30 toneladas e funcionava com 18.000 válvulas, 10.000 capacitores, além de milhares de resistores e relés, os quais consumiam cerca de 150.000 watts, que custou meio milhão de dólares.

Sua utilização era muito trabalhosa, com componentes muito frágeis e pouco duráveis. Para se ter uma idéia do processo de inserção de dados era através de programação através de cabos, mas esse trabalho todo compensava na hora dos resultados apresentados pelo o ENIC, reduziu para 30 segundos os cálculos de trajetórias de mísseis que antes levavam cerca de 1.000 segundos.

Esse é um ponto muito importante da evolução dos computadores, foi por causa da guerra mundial que começaram a desenvolver os computadores.

Em 1945 a 1950, foi feita uma descoberta que seria de extrema importância para a evolução dos computadores, que foi a lógica dos circuitos, as instruções podiam ser acessadas e manipuladas internamente tudo a partir de operações lógicas e números binários. O primeiro com o conceito de Von Neumann feito em escala comercial foi o

UNIV AC I (Universal Automated Computer), Em 1953 a IBM revolucionou a venda de computadores com o lançamento de dois modelos de computadores o IBM 701 (1953) e o IBM 650, em 1954 ela vendeu aproximadamente mais de 1000 unidades.

Logo após foi inventado o transistor, que tinham um tamanho de 100 vezes menor que uma válvula não precisava de tempo para aquecimento, consumia menos energia, era mais rápido e mais confiável. O seu potencial era tão grande que seu calculo podia chegar a microssegundos (milionésimos), quem sobre aproveitar esse grande potencial mais uma vez foi a IBM com os modelos 1401 e IBM 7094.

Com a substituição dos transistores pela tecnologia de circuitos integrados e outros componentes eletrônicos miniaturizados, montados num único chip, que já calculavam em nanossegundos (bilionésimos). Idéia era colocar num único chip todos os componentes eletrônicos do computador e os circuitos integrados eram confiáveis, não tendo partes moveis, são menores, com circuitos pequenos, portanto sua comunicação era mais rápida, baixo consumo de energia e principalmente com menor custo.

A IBM entrou na briga na venda de computadores em 16 de agosto de 1981 lançado um computador com a tecnologia de 16 bits que em pouco tempo se tomou um padrão (PC - Personal Computer).

Em 1984 a IBM já detinha mais da metade de todo o mercado de microcomputadores, principalmente pela sua atuação mercadológica que incluiu, nesse período. Em fevereiro de 1993, lança um novo modelo, o PC-XT com winchester opcional.

1. 1. Evolução dos Processadores

Atualmente temos duas empresas que produzem processadores, a Intel e AMD. Ambas vem travando uma briga há muito tempo para se tornar a melhor empresa de processadores.

A Intel foi criada em 1969, pioneira na fabricação de processadores, teve seu primeiro processador lançado em 1971, chamado de Intel 4004 e a partir dessa data a Intel foi criando vários processadores. A Intel desenvolvia seus processadores baseados em bit's, foi conquistando o mercado com seus processadores, ate se tornar líder de mercado. Entre suas criações podemos citar o Intel 386 lançado em (1988), esse foi o chip que começou tudo que vemos hoje. O 386 o primeiro processador de 32-bit para Pc's e também poderia usar cachê de 16 bytes, chegava de 12.5MHz até 33MHz. Todos os chips da família 386 eram compatíveis com o código binário com os seus antecessores, isso significa que o usuário não precisa adquirir um novo software para usá-lo. Além de tudo, esse foi um grande passo para o desenvolvimento de processadores. Ele deixou muitos padrões que são usados até hoje.

Os Intel i486 (também chamados 486 ou 80486) são uma família de microprocessadores CISC da Intel que fazem parte da família de processadores x86 e seu predecessor foi o processador 386. Entretanto, do ponto de vista do hardware, a arquitetura do i486 é um grande avanço. Ele vem com um cache de dados e instruções no chip, uma unidade de ponto flutuante (FPU) adicional, pela primeira vez dentro do chip (os modelos DX), e uma unidade de interface de barramento aprimorada. Em adição a isso, o núcleo do processador pode sustentar a taxa de execução de uma

instrução por ciclo de clock. Esses avanços dobraram o desempenho bruto em relação a um 386 de mesmo clock.

Depois desse importante processador a Intel criou Pentium I, Pentium II, Pentium III, Celeron, Pentium M, com sua estrutura, arquitetura, seu processamento interno e Chipset sempre evoluindo e sendo melhor que as versões anteriores.

Após isso ela lança a linha Pentium 4, Processador Intel® Pentium® 4, Processador Intel® Pentium® 4 com suporte para a tecnologia[1] HT, Processador Intel® Pentium® 4 com suporte para a tecnologia[1] HT Extreme Edition, Processador Intel® Pentium® D, Processador Intel® Pentium® Dual-Core, Processador Intel® Pentium® Extreme Edition, Processador Intel® Core™2 Duo, Processador Intel® Core™2 Quad e Processador Intel® Core™2 Extreme. Esses processadores possuem mais versões.

Fundada em 1969, a AMD (Advanced Micro Devices), teve seu começo fornecendo chips para a Intel, depois de algum tempo decidiu entrar no mercado de processadores, produzindo seus próprios processadores.

Seu diferencial de mercado era o preço, os processadores da AMD tinham um baixo custo, por isso começo a despertar curiosidades nos consumidores.

Sediada em Sunnyvale, Califórnia, hoje a AMD fornecedora mundial de circuitos integrados para os mercados de computadores pessoais e de rede, bem como para o mercado de comunicação, com fábricas nos Estados Unidos, Europa, Japão e Ásia.

Seu primeiro processador foi o AMD 286, Produzido em 1982, sua velocidade era de 12Mhz e 16Mhz, onde entrou no mercado de processadores para ser concorrente da Intel.

Depois a AMD produziu os processadores, AMD 386, AMD 486, AMD 586, AMD K5, AMD K6, AMD K6-2 AMD K6-3. Mas sua tecnologia sempre era baseada no que a Intel criava e por isso a AMD sempre esta um passo atrás da Intel em termos de processadores.

A AMD começou a mudar com o lançamento do processador Athlon que após algumas implementações como inclusão de cachê L1 e L2 conseguiu emparelhar sua tecnologia atingindo uma igualdade de processadores com a Intel.

CAPÍTULO 2

EVOLUÇÃO DOS DISPOSITIVOS MÓVEIS

A empresa americana Bell Company, em 1947, desenvolveu um sistema que permitia a utilização de telefonia móvel dentro de uma determinada área utilizando o conceito de células, ou áreas de cobertura, derivando deste, o nome celular. Ainda naquele ano, nos Estados Unidos, a AT&T e a Bell propuseram à FCC (*Federal Communication Commission*) a alocação de um número de freqüência de rádio especificamente para comunicação móvel, mas a FCC disponibilizou apenas poucas freqüências, possibilitando que somente 23 pessoas se conectassem simultaneamente ao sistema de uma determinada área de cobertura. Isto, na época, tornou a tecnologia inviável comercialmente.

Levou cerca de 20 anos para a FCC reconsiderar o número de freqüências destinadas à telefonia móvel e aumentar esse número para suportar mais usuários.

Em 1968, as empresas AT&T e Bell definiram o sistema de uso de torres para atender aos usuários por áreas, conforme seu deslocamento, desta forma, continuou a propagação do sistema até a cobertura atingida nos dias atuais.

A Bell, em 1973, já possuía um sistema de comunicação instalado em carros de polícia, mas foi a Motorola, naquele mesmo ano, a primeira a incorporar essa tecnologia a um dispositivo móvel de comunicação fora de um veículo, para uso pessoal. Contudo, em janeiro de 1979 o sistema foi realmente testado com 200 pessoas em Chicago.

Mas foi 10 anos depois, em 1983, que surgiu o primeiro celular aprovado pelo FCC, o DynaTAC 8000X, da Motorola - que junto com a empresa Ameritech iniciou o uso comercial da telefonia celular no Estados Unidos e no mundo.

Nesse momento a Motorola já havia investido cerca de US$ 100 milhões em 15 anos de pesquisas em tecnologia móvel celular.

O aparelho com cerca de 1kg, tinha capacidade para uma hora de conversação e oito de stand-by, memória para 30 números, além de *display* com LED (*Light Emitting Diodes*). "Consumidores ficaram tão impressionados com a idéia de estar sempre conectados que se dispunham a pagar US$ 3,995", lembra Rudy Krolopp, um dos primeiros participantes da equipe de desenvolvimento do aparelho. As listas de espera chegavam aos milhares, mesmo com o preço do DynaTAC 8000X; Hoje um aparelho de última geração está na faixa de U$250.

Passados 20 anos do início da telefonia móvel comercial, o conceito do uso desses dispositivos mudou; hoje os consumidores procuram maneiras de reforçar seus estilos de vida. O celular tornou-se, neste contexto, uma extensão da personalidade do usuário, uma peça capaz de enriquecer relacionamentos, divertir, aumentar a produtividade e expressar individualidade. Isso significa comunicar, dividir, criar e divertir com voz, textos, imagens, músicas e vídeos.

Com o barateamento da tecnologia, o número de usuários de celular no mundo passou de cerca de 300 mil, em 1984, para mais de 1,2 bilhão, atualmente. À medida que a indústria cresceu, as empresas anteciparam a demanda por tecnologias inovadoras, de acesso sem fio à internet, a jogos, músicas e imagens digitais.

A entrada do Brasil na era da telefonia celular ocorreu em novembro de 1990. Desta data até dezembro de 2003 passamos de 667 para mais de 43 milhões de acessos móveis. Uma trajetória impressionante de crescimento de uma tecnologia que marca presença em vários setores da sociedade.

“O telefone celular chegou a rincões do país, algumas vezes, antes mesmo que as populações locais dispusessem de água encanada ou saneamento básico”.

Pouco a pouco a tecnologia de comunicação móvel foi sendo incorporada ao cotidiano e às mais variadas situações. Desde novas formas de contato pessoal, passando por áreas de negócios, até salvamento de vidas, foram viabilizados através da tecnologia do celular. O celular deixou de ser apenas um objeto de desejo para se tornar uma necessidade, deixou de ser artigo de luxo para, em muitos casos, se tornar item básico. A telefonia móvel atingiu um patamar que permeia todo o tecido da sociedade brasileira moderna e que, assim como os computadores, criou um forte vínculo de dependência com essa tecnologia.

Shneiderman (2002) considera que, ao contrário da velha computação que valorizava os aspectos da máquina, a nova computação valoriza o que, efetivamente, os usuários podem fazer com as máquinas. Ainda segundo o pesquisador, a tecnologia não é o objetivo final e sim o meio pelo qual o usuário pode satisfazer diversas necessidades e enriquecer suas experiências.

O início da interação humano-computador está intrinsecamente relacionado ao desenvolvimento de novas tecnologias de interfaces para computadores e a história da

interface gráfica. Vários destes conceitos são empregados nos sistemas dos aparelhos celulares atuais.

As Interfaces dos primeiros celulares eram compostas, basicamente, por texto, e não permitiam uma riqueza de detalhes na apresentação das informações no visor do dispositivo, em razão da tecnologia imprópria.

A resolução, responsável pela melhor definição dos textos e formação de desenhos no visor, está gradualmente aumentando com a evolução dos aparelhos celulares, assim como a quantidade de cores. Com esta melhora na tecnologia de exibição de informações, os elementos gráficos na tela podem ser melhor representados, equiparando-se a pequenos computadores pessoais.

Assim como aconteceu como os assistentes pessoais digitais, a interface gráfica dos sistemas dos computadores pessoais foi usada como referência para apresentação dos elementos em alguns modelos de celulares, ao ponto de alguns telefones funcionarem com versão compacta do sistema Windows.

O som, que teve uma evolução com menos etapas do que a resolução da tela do celular, também gerou teve grande evolução. Hoje, inúmeros modelos de celulares, além de permitirem gravação de voz, grande qualidade de emissão de alertas sonoros, chamados sons polifônicos, também possibilitam mixagens e download de toques e músicas da Internet.

A evolução tecnológica busca cada vez mais criar dispositivos menores. Apesar de ser bastante atrativa aos consumidores, esta tendência tecnológica de priorizar a

miniaturização implica em problemas ergonômicos de manipulação do aparelho. Tais problemas aparecem na redução do tamanho das teclas e caracteres no visor, além da redução na quantidade de teclas presentes nos celulares, indo de encontro ao crescente aumento de funções nesses dispositivos.

CAPÍTULO 3

SISTEMA OPERACIONAL

O projeto GNU foi fundado em 1983, GNU significa (GNU's Not Unix, ou GNU não é Unix). Nasceu com a filosofia de desenvolver, disponibilizar softwares liberando seus códigos fontes para copia, podendo ser alterados e distribuídos, de forma que a pessoa necessitasse gratuitamente. Eles também iniciaram a ideologia de software livre, a palavra livre esta ligada a liberdade e não em custo.

Naquela época não existia um Sistema Operacional livre, (Um sistema operacional é um conjunto de programas básicos e utilitários que fazem seu computador funcionar) apenas software proprietário, tendo em vista isso o projeto GNU queria desenvolver um Sistema Operacional livre. O projeto GNU estava desenvolvendo software como compiladores, editores, formatadores de texto, software de e-mail, utilitários, bibliotecas, etc.

Esse desenvolvimento tomou vários anos, a idéia do projeto era tornar o sistema operacional compatível com o Unix, já que o Unix era um ótimo sistema operacional, o GNU queria conquistar esses usuários, para isso não queriam que houvesse impacto na mudança do sistema operacional Unix para GNU. Nos anos 90, o projeto GNU já tinha encontrado ou escrito todos os componentes principais. Mas ainda faltava o núcleo do Sistema Operacional.

3.1. Historia do Kernel

Desenvolvido pelo Linus Torvalds em 1991, era um sistema monolítico, um grande bloco de código que além do núcleo do sistema, continha drivers de dispositivos era compilado no Minix baseado em Unix, começava compilando o Kernel e em seguida algumas ferramentas básicas como o gerenciador de boot, o bash e o gcc. A partir de certo ponto podíamos iniciar no próprio Linux e compilar os demais programas a partir dele mesmo.

Torvalds notificou na internet sua criação, disponibilizou o código-fonte do que tinha desenvolvido a todos o,s interessados.

3.2. Estrutura do Kernel

Kernel é o núcleo do sistema operacional, responsável por gerenciar recursos do sistema operacional, ele esta na camada mais baixa da interfase com do hardware, é no Kernel que estão definidas funções para operação com periféricos (mouse, discos, impressoras, interface serial / interface paralela), gerenciamento de memória, entre outros.

Explicando o mais genericamente, o kernel vai trabalhar sobre o hardware para utilizar seus recursos, e assim os processos poderão usar os recursos de forma segura e padronizada.

A arquitetura do kernel pode ser monolítico que consistem em um bloco com todas as suas funcionalidades carregadas na memória, ou modular, com os módulos específicos para cada tarefa carregados opcionalmente, dinamicamente.

3.7. Kernel monolítico

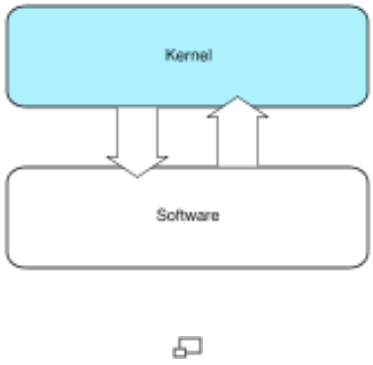

Fig. 01 - Diagrama de interação de um kernel monolítico (ou mono-bloco).

Kernel monolítico ou mono-bloco é um kernel que implementa uma interface de alto nível para possibilitar chamadas de sistema específicas para gestão de processos, concorrência e gestão de memória por parte de módulos dedicados que são executados com privilégios especiais.

Alguns exemplos deste tipo de kernel:

- BSD
- Linux

MS-DOS e derivados, incluindo Windows 95, Windows 98 e Windows ME

Solaris.

3.8. Integração do kernel e GNU

Com a integração dos softwares do projeto GNU e o Kernel do Linus Torvalds nascia o primeiro Sistema Operacional livre completo. Uma combinação perfeita, pois um completava a necessidade do outro.

Muitas pessoas conhecem e se referem o Sistema Operacional Livre como Linux, mas na verdade o certo é se dizer GNU/LINUX. Por essa junção dos dois projetos.

A partir daí programadores do mundo inteiro começaram a desenvolver o Linux conforme sua necessidade, disponibilizando sua alterações, melhorando o kernel cada vez mais, grandes empresas como a IBM e a Novell começaram a contribuir também com o desenvolvimento do Kernel, a fim de tornar o sistema mais robusto e adicionar recursos necessários para determinadas tarefas. A partir daí com o passar do tempo surgiram vários Sistemas Operacionais baseados em GNU/LINUX como Debian, OpenSusSe, Fedora, Slackware, Unbutu entre varias outras distribuições existente. Atualmente ambas estão evoluindo cada vez mais nos seus Sistemas Operacionais.

3.9. Direitos Autorais

O GNU/LINUX também tem possui direitos autorais.

A licença GPL

A GPL (GNU PUBLIC LICENSE) consistem em um lei copyright que garante que um software , uma vez livre, não pode se tornar proprietário.

Que consiste em:

- Você pode usar o software para a finalidade que você quiser.
- Pode modificar e adaptar as suas necessidades
- Pode distribuir quantas copias quiser gratuitamente ou não
- Uma vez alterado você pode disponibilizar seu código fonte para quem quiser usa-lo.

A distribuição Debian

O Debian é um sistema operacional (SO) livre para seu computador, baseado em GNU/Linux, utiliza no seu núcleo o kernel e ferramentas do sistema compiladores, editores, formatadores de texto, software de e-mail entre outros do projeto GNU. Por isso GNU/Linux. Ele vem com mais de 18733 pacotes contendo softwares pré-compilados e distribuídos em um bom formato, que torna fácil a instalação deles na sua máquina.

3.10. Interpretador de comandos

É o comunicador dos usuários com o sistema nosso caso o kernel, ele é responsável por captar os comandos digitados pelo usuário e levá-lo para o kernel. Existem duas formas de execução de comandos.

A interativa, o usuário digita os comandos pelo teclado para serem executados e Não interativa, é a forma de execução de comandos através de scripts, os scripts são criados em um arquivo para serem executados.

O GNU/Linux possui vários interpretadores de comandos que são ash, csh, tcsh, sh, o mais utilizado é o bash.

3.11. Terminal Virtual (console)

O terminal do GNU/Linux é baseado em multi-usuario que pode ser chamado de terminal virtual, um terminal trabalha é independente do outro. Assim cada terminal ser utilizado para tarefas distintas.

CAPÍTULO 4

LINGUAGEM DE PROGRAMAÇÃO PHP

O surgimento da linguagem PHP foi em 1994, como um subconjunto de scripts Perl criados por Rasmus Lerdof, e inicialmente chamado de Personal Home Page Tools. Em 1997 foi lançado o novo pacote da linguagem com o nome de PHP/FI, trazendo a ferramenta Forms Interpreter, que era na verdade um interpretador de comandos SQL. Talvez Rasmus Lerdorf não imaginasse que estaria criando o que hoje se tornou um fenômeno em termos de desenvolvimento para aplicações Web.

Mais tarde, Zeev Suraski desenvolveu o analisador do PHP 3 que contava com o primeiro recurso de Orientação à Objetos, que dava poder de alcançar alguns pacotes, tinha herança e dava aos desenvolvedores somente a possibilidade de implementar propriedades e métodos.

Pouco depois, Zeev e Andi Gutmans, escreveram o PHP 4, abandonando por completo o PHP 3, dando mais poder à máquina da linguagem e maior número de recursos de orientação a objetos. O problema sério que apresentou o PHP 4 foi a criação de cópias de objetos, pois a linguagem ainda não trabalhava com apontadores ou handlers, como é a linguagem JAVA.

O problema fora resolvido na versão atual do PHP, a versão 5, que já trabalha com handlers. Caso se copie um objeto, na verdade copiaremos um apontamento, pois, caso haja alguma mudança na versão original do objeto, todas as outras também sofrem a alteração, o que não acontecia na PHP 4.

Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores web. Diversos módulos são criados no repositório de extensões PECL (PHP Extension Community Library) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++. Pode ser, dependendo da configuração do servidor, embutida no código HTML. Existem versões do PHP disponíveis para os seguintes sistemas operacionais: Windows, Linux, FreeBSD, Mac OS, OS/2, AS/400, Novell Netware, RISC OS, IRIX e Solaris.

Construir uma página dinâmica baseada em bases de dados é simples, com PHP, este provê suporte a um grande número de bases de dados: Oracle, Sybase, PostgreSQL, InterBase, MySQL, SQLite, MSSQL, Firebird etc, podendo abstrair o banco com a biblioteca *ADOdb*, entre outras.

PHP tem suporte aos protocolos: IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP. É possível abrir sockets e interagir com outros protocolos. E as bibliotecas de terceiros expandem ainda mais estas funcionalidades.

O que deferência o PHP de linguagens como Javascript no lado do cliente é que o código é executado no servidor. Se você tivesse um script similar ao acima em seu servidor, o cliente receberia os resultados da execução desse script, sem nenhum modo de determinar como é o código fonte. Você pode inclusive configurar seu servidor para processar todos os seus arquivos HTML como PHP, e então não haverá nenhum modo dos usuários descobrirem está sendo utilizada, no caso o PHP.

Uma grande vantagem do uso do PHP está no fato de ele ser extremamente simples para um iniciante, mas oferece muitos recursos para o programador profissional. Não se preocupe em ler longas listas de funções do PHP. Você pode pular essa parte e começar a escrever scripts em poucas horas.

Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas fixos. A mais notável é a PHP-GTK. Trata-se de um conjunto do PHP com a biblioteca GTK, portada do C++, fazendo assim softwares inter-operacionais entre Windows e Linux. Na prática, essa extensão tem sido pouco utilizada para projetos reais.

Enfim, o PHP conquistou muito espaço nos últimos anos. As empresas perceberam esse rápido crescimento e cada vez mais estão considerando o PHP como melhor alternativa para o desenvolvimento de suas aplicações Web. Seu crescimento não foi somente quantitativo, mas também qualitativo.

4.1. O que é preciso?

Seu servidor tem suporte ao PHP ativado e que todos os arquivos terminam com a extensão .php são tratados pelo PHP. Na maioria dos servidores esta é a extensão padrão para os arquivos PHP, mas pergunte ao seu administrador só para ter certeza. Se o seu servidor suporta PHP então você não precisa fazer mais nada. Apenas crie seus arquivos .php e coloque-os no seu diretório web e o servidor irá com um passe de mágica mostrar suas páginas PHP. Não há nenhuma necessidade de compilação para qualquer ferramenta extra. Pense nesses arquivos PHP como se eles fossem páginas

HTML com algumas tags à mais que deixaram você fazer coisas mais interessantes do que somente páginas HTML estáticas.

Por exemplo você quer salvar sua preciosa conexão e desenvolver tudo localmente. Neste caso, você precisará instalar um servidor web, como o *Apache*, e claro o PHP. Você também irá querer instalar uma base de dados, como por exemplo o MySQL. Você pode instalá-los separadamente ou pelo jeito mais simples que é usar os pacotes pré-configurados. que irão instalar automaticamente todas as coisas com apenas alguns cliques. É simples configurar um servidor web com suporte ao PHP em qualquer sistema operacional, incluindo Linux e Windows.

CAPÍTULO 5

PROTOCOLO HTTP

A maioria dos sites da Web executam processos no servidor geralmente na porta 80 TCP aguardando conexões dos clientes(geralmente os navegadores). Depois de estabelecida a conexão, o cliente envia uma solicitação e o servidor envia uma resposta. A conexão é, então liberada. O protocolo define as solicitações e respostas válidas é chamado HTTP. No entanto um exemplo com o protocolo http pode dar uma idéia de como os servidores Web funcionam. Para esse exemplo o usuário acabou de dar um clique sobre um texto ou ícone que aponte para a página cujo nome é http://www.w3.org/hypertext/WWW/TheProject.html. Os procedimentos que ocorrem desde o momento que o usuário dá o clique e em que a página é apresentada são os seguintes:

1. O navegador determina o URL(verifica qual foi selecionado).
2. O navegador pergunta ao DNS qual é o endereço IP de www.w3.org.
3. A DNS responde 18.23.0.23.
4. O navegador estabelece uma conexão TCP com a porta 80 em 18.23.0.23.
5. Em seguida, o navegador envia um comando GET/hypertext/WWW/TheProject.html.
6. O servidor envia www.w3.org envia o arquivo TheProject.html.
7. A conexão TCP é liberada.
8. O navegador apresentado apresenta todo o conteúdo de TheProject.html.
9. O navegador busca e apresenta todas imagens de TheProject.html.

5.1. HTTP – HyperText Transfer Protocol

O protocolo de transferência da Web é o HTTP (HyperText Transfer Protocol). Cada interação consiste em uma solicitação ASCII, seguida de uma resposta RFC 822 do tipo fornecido pelo MIME. Embora o uso do TCP para conexão de transporte seja bem comum, essa não é uma exigência formal do padrão.

O HTTP está em constante evolução. Há várias versões em uso e outras tantas em desenvolvimento. O protocolo HTTP consiste em dois itens razoavelmente distintos: um conjunto de solicitações dos navegadores aos servidores e um conjunto de respostas que retornam no caminho inverso.

Todas novas versões do HTTP aceitam dois tipos de solicitações: simples e completas. Uma solicitação simples é a GET que identifica a página desejada, sem a versão do protocolo. A resposta é formada apenas pela a página, sem cabeçalhos, sem MIME e sem códigos.

As solicitações completas são indicadas pela presença da versão do protocolo na linha de solicitação GET. As solicitações podem constituir em várias linhas, seguidas de uma linha em branco para indicar o fim da solicitação. A primeira linha de uma solicitação completa contém o comando (onde GET é apenas uma das possibilidades), a página desejada e o protocolo/versão. As linhas subseqüentes contem cabeçalho RFC 822. O RFC 822 que descreve a sintaxe dos campos de cabeçalho do mail e define um conjunto de campos de cabeçalho e a respectiva interpretação.

Apesar de ter sido projetado para utilização na Web, o http foi, intencionalmente, criado para ser o mais genérico do que o necessário com vistas a futuras aplicações orientadas a objetos. Por esse motivo a primeira palavra na linha de solicitação completa é o nome do método (comando) a ser executado na página Web (ou objeto genérico). Quando você acessa objetos genéricos, outros métodos específicos dos objetos podem se tornar disponíveis. Os nomes fazem distinção entre maiúsculos/minúsculos; sendo assim; GET é um método válido mas get não é.

Método	**Descrição**
GET	Solicita a leitura de uma página da Web
HEAD	Solicita a leitura de um cabeçalho de página Web
PUT	Solicita o armazenamento de uma página da Web
POST	Acrescenta a um recurso (por exemplo, uma página da Web)
DELETE	Remove a página da Web
LINK	Conecta dois recursos existentes
UNLINK	Desfaz a conexão entre dois recursos

O método GET solicita ao servidor que envie a página (ou objeto, no caso mais genérico), codificando no MIME. Entretanto, a solicitação GET for seguida de um cabeçalho If-Modified-Since, o servidor só enviará os dados se eles tiverem sido alterados desde a data fornecida. Com esse mecanismo, um navegador pode, ao receber a solicitação de uma página armazenada no cachê, enviar uma solicitação condicional ao servidor, associando a data de alteração da página. Se a página no cachê ainda estiver válida, o servidor retornara uma linha de status anunciando esse fato, eliminando assim o overhead de transferir a página outra vez.

O método HEAD solicita apenas o cabeçalho da mensagem, sem página propriamente dita. Esse método pode ser utilizado para obter a data da última modificação feita na página, para obter informações para índices, ou apenas para testar a validade de um URL. Não existem solicitações HEAD condicionais.

O método PUT é o inverso de GET: em vez de ler, ele grava a página. Esse método possibilita a criação de um conjunto de páginas da Web em um servidor remoto. O corpo da solicitação contém a página. Ela pode ser codificada usando o MIME. Nesse caso, as linhas depois de PUT podem conter *Content-Type* e cabeçalhos de autenticação, para provar que o responsável pela chamada tem mesmo permissão para realização a operação solicitada.

Um pouco semelhante ao PUT é o método POST, que também transporta um URL. No entanto, em vez de substituir os dados existentes, os novos dados são acrescentados, em um sentido mais genérico. Enviar uma mensagem a um newsgroup ou incluir um arquivo em um BBS são dois exemplos de acrescentar nesse contexto. A intenção é fazer com que a Web adquira a funcionalidade do sistema news da USENET.

DELETE faz exatamente como proposto pelo nome: exclui a página. O exemplo de PUT, a permissão e a autenticação têm papel fundamental. Não há garantia que o DELETE tenha sido bem-sucedido, pois, mesmo que o servidor http remoto esteja pronto para excluir a página, o arquivo subjacente pode ter um modo que proíba o servidor HTTP de modificá-lo ou excluí-lo.

Os métodos LINK e UNLIK permitem que sejam estabelecidas conexões com páginas já existentes ou outros recursos.

Cada solicitação obtém uma resposta que consiste em uma linha de status e possivelmente informações adicionais (por exemplo, uma página da Web ou parte dela). A linha de status pode apresentar o código 200(OK), ou qualquer um dentre uma variedade de códigos de erro, por exemplo, 304 (não modificado), 400 (solicitação invalida), 403 (proibido) ou 404 (página inexistente).

Os padrões HTTP descrevem cabeçalhos e corpos de mensagens com considerável riqueza de detalhes. Basta dizer que elas são muito semelhantes às mensagens RFC 822 MIME.

CAPÍTULO 6

A INTEGRAÇÂO ENTRE SERVIDOR E SISTEMA

6.1. Shell Script

Na linha de comandos de um shell, podemos utilizar diversos comandos um após o outro, ou mesmo combiná-los numa mesma linha. Se colocarmos diversas linhas de comandos em um arquivo texto simples, teremos em mãos um Shell Script, ou um script em shell, já que Script é uma descrição geral de qualquer programa escrito em linguagem interpretada, ou seja, não compilada. Podemos então ter um script em php, um script perl e assim em diante.

Uma vez criado, um Shell Script pode ser reutilizado quantas vezes for necessário. Seu uso, portanto, é indicado na automação de tarefas que serão realizadas mais de uma vez. Todo sistema Unix e similares são repletos de scripts em shell para a realização das mais diversas atividades administrativas e de manutenção do sistema.

Os shell scripts são responsáveis pela parte mais importante do servidor, a execução dos serviços, ele será responsável por parar, iniciar os serviços.

O shell script é um arquivo executável que nele estão contidos todos os comandos do Linux, esse arquivo poderá ser executado se seu usuário tenha permissão de executá-lo.

A shell scripts manipulará os serviços do Postfix, Samba, *Apache* entre outros.

Os shell scripts vão ser executados pelo *apache*, que terá uma ajuda do sudo. O *apache* executa o script pelo sudo e assim o script é executado.

Sudo é responsável pelos usuários sem permissão de Super usuário, poder executar comandos de Super usuário, no nosso caso os script e manipulação dos serviços. O sudo possui um arquivo de configuração, que contem os arquivos que poderão ser aberto por ele, junto com as permissões e qual usuário do sistema pode executá-lo.

6.2. SUDO

Sudo foi desenvolvido por Bob Coggeshall e Cliff Spencer nos anos 80. A versão atual é mantida por Todd C. Miller e distribuída sob um tipo de licença BSD.

O comando sudo do sistema operacional Unix permite a usuários comuns obter privilégios de outro usuário, em geral o super usuário, para executar tarefas específicas dentro do sistema de maneira segura e controlável pelo administrador. O nome é uma forma abreviada de se fazer substituindo usuário ou fazer como super usuário.

Um super usuário precisa definir no arquivo /etc/sudoers quais usuários podem executar sudo, em quais computadores podem fazê-lo e quais comandos

podem executar através dele. Por ser uma tarefa delicada em termos de segurança a edição direta deste arquivo não é recomendada. Para isso foi criada a ferramenta denominada visudo que invoca um editor para uma cópia do arquivo /etc/sudoers e em seguida verifica o conteúdo do arquivo antes de substituir a configuração atual.

Um usuário que esteja inscrito no /etc/sudoers pode invocar o comando da seguinte forma:

sudo [-u usuário] comando

Onde comando é o comando que deseja executar. A opção -u usuário serve para especificar qual usuário deve ser utilizado para executar o comando, se omitida sudo considera o usuário root.

Antes de o comando ser executado, sudo confirma a identidade do usuário pedindo sua própria senha. Se o usuário tem permissão garantida na configuração, o comando é executado como super usuário e retorna. A linha de comando continua sendo a do usuário que executou sudo e não a do super usuário.

Caso sudo seja executado de forma não permitida pela configuração, um registro da ocorrência é feito no arquivo /var/log/auth.log.

6.3. Executando os comandos com o PHP

O PHP possui comandos nativos para realizar operações direto no sistema operacional, mas só podem executar comandos no qual o usuário *apache* tenha permissões. Um dos comandos que permite o PHP executar comandos do sistema operacional é o system().

system() executa o *comando* indicado e mostra o resultado. Se for dada uma variável como segundo argumento, então o código de status de retorno será escrito nesta variável.

A função system() também tenta automaticamente limpar o buffer de saída do servidor mandando os dados para o navegador após cada linha de saída se o PHP estiver sendo executado como módulo de servidor. Retorna a última linha da saída do comando em caso de sucesso ou um valor falso em caso de falha.

METODOLOGIA

Neste trabalho procuramos caracterizá-lo como um trabalho com possibilidades práticas para o desenvolvimento de um aplicativo para a administração de servidores via dispositivos móveis.

O servidor deve possuir o sistemas operacional Linux cujo desenvolvimento em código fonte aberto disponível sob licença GPL, de software livre e possuir um ambiente ideal para a atuação do software, devido sua fácil integração com a linguagem de programação escolhida.

O Debian foi escolhido como distribuição do sistema operacional pela facilidade da instalação dos pacotes e usabilidade do sistema no diagnóstico de problemas e resolução.

Os aplicativos necessários para o desenvolvimento do sistema são *Apache*, MySQL, PHP. Após a configuração dos itens básicos de funcionamento do sistema, serão instalados módulos adicionais para a administração via dispositivo móvel.

Na instalação do *Apache* é criado um usuário chamado “*apache*” este usuário é quem vai executar os comandos solicitados pelo aplicativo. Todos os arquivos de extensão .php que forem executar comandos dos Shell Scripts, devem ter o usuário apache como proprietário.

Para a instalação de outros pacotes como Postfix, Bind e SSH foi usado o comando apt-get.

O apt-get supriu as necessidades iniciais. Durante o desenvolvimento houve alguns problemas como a localização dos pacotes instalados e as versões de bibliotecas instaladas pelo apt-get e como adicionar módulos específicos para cada pacote.

Para obter maior autonomia sobre os pacotes foram compilá-los, depois de instalados os softwares, iniciamos a execução de testes para verificação de erros e desempenho.

Como o decorrer do desenvolvimento foram adicionados alguns pacotes ao servidor como o Sudo, considerado para o sistema um dos principais elementos para o seu funcionamento.

O pacote Sudo é necessário para que o usuário Apache que não possui comandos de super usuário possa executá-los através do dispositivo móvel.

Para evitar invasão do servidor selecionamos os comandos que o usuário Apache poderá executar. Foi realizado um estudo detalhado em cada comando de manipulação de serviços, para evitar uma falha de segurança e o aplicativo oferecer ao máximo de funcionamento. Para que haja um único usuário manipulando o Sudo, com execuções restritas a um único local para realizar manutenções necessárias.

Para o desenvolvimento da Interface utilizamos uma tela simples e de fácil memorização. Lembramos que o acesso dos usuários será feita através da internet para que a interface seja leve e simples.

Para facilitar o aprendizado aproveitamos o conhecimento já existente do usuário sobre as cores do semáforos, onde usuário verificará o estado atual do serviço.

Dentre os motivos que nos levaram a escolher a linguagem foi sua vasta biblioteca de funções nativas para executarmos rotinas administrativas no servidor.

Para facilitar e organizar em uma arquitetura MVC (“Model View Controller”) optamos por utilizar um framework, o CakePHP, é baseado no Ruby on Rails e utiliza padrões de projeto conhecidos, tais como *ActiveRecord*, *Association Data Mapping*, *Front Controller.*

O CakePHP é escrito em PHP que tem como principais objetivos oferecer uma estrutura que possibilite ao programador PHP de todos os níveis desenvolverem rapidamente aplicações complexas, sem perder flexibilidade.

7.1 O APLICATIVO

O sistema por si próprio não tem acesso direto a todas as funções do sistema operacional, por questões de segurança. Ele ainda conta com uma camada de shell scripts que vão realizar as operações no servidor e retornar ao aplicativo se

o shell script foi executado com sucesso ou não. A figura abaixo ilustra o funcionamento:

Fig. 02 – Camadas de atuação

Os arquivos de shell script para a administração dos serviços estão em um diretório especifico do sistema operacional. Esse diretório é configura no Sudo, para que o usuário apache possa executar somente os Shell Scripts desse diretório.

O Sudo irá dar permissões de Super usuário para os usuários comuns, isso torna possível a execução dos Shell Scripts que realizam operações de administração do servidor.

7.2 O SUDO

O sudo é configurado pelo comando visudo, onde é editado no /etc/sudoers, nesse local são configurados os usuários comuns que poderão executar um determinado comando de super usuário. No sudoers se coloca o usuário normal, o local do shell script, mais o caminho do destino que ele executará um determinado comando. Todos os comandos sudo são executados por um usuário comum necessita de uma senha por motivo de segurança, mais isso pode ser alterado no /etc/sudoers para não pedir senha.

Sendo assim foi adicionado o seguinte comando para o funcionamento do sistema através do script.

Apache ALL=”LOCAL DO SCRIPT” NOPASSWD:”LOCAL DOS SERVIÇOS“

7.3 CASO DE USO

Nosso caso de uso é ilustrado na figura abaixo:

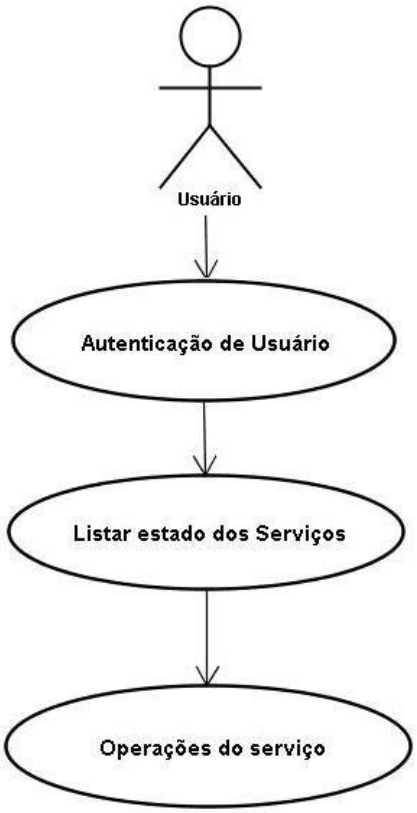

Fig.03 – Diagrama de Caso de Uso

O administrador do servidor através de seu dispositivo móvel irá acessar a URL do aplicativo. O aplicativo irá solicitar o usuário e sua respectiva senha para poder autenticar-se no aplicativo, caso seja informado corretamente é listado alguns serviços do servidor. Ele pode navegar entre e pode realizar operações do serviço.

7.4 A ESTRUTURA DO APLICATIVO

Foi modelado um componente de autenticação de usuários, esses usuários não precisão pertencer ao sistema operacional, são apenas usuários com senhas para o acesso ao aplicativo. Através desses usuários e suas respectivas senhas o administrador do sistema poderá acessar através de qualquer dispositivo móvel as operações que o aplicativo permite realizar no sistema operacional.

Para realizar qualquer operação é necessário autenticar-se, caso contrario o sistema irá retornar para a tela de autenticação, até que o usuário informe corretamente, isso torna o aplicativo mais seguro. A figura abaixo ilustra o diagrama de atividade do componente de autenticação:

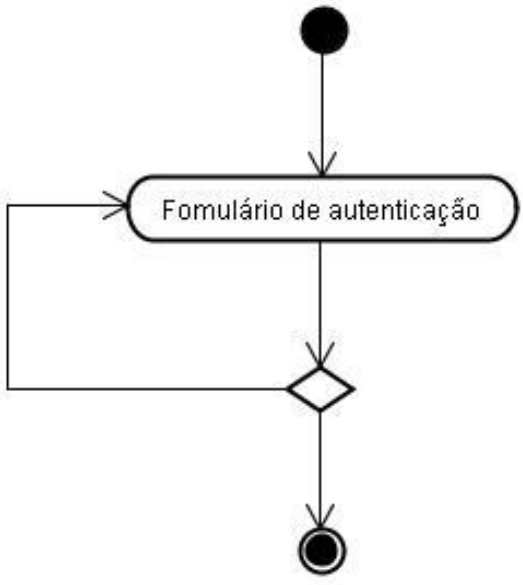

Fig. 04 - Diagrama de atividade para autenticação de usuários.

Após autenticado o aplicativo irá verificar o estado de cada serviço, cadastrado em sua base de dados, e exibirá a informação para o usuário através de cores do semáforo. Vermelho para serviço parado e Verde para serviço rodando. O diagrama de atividade abaixo ilustra o funcionamento:

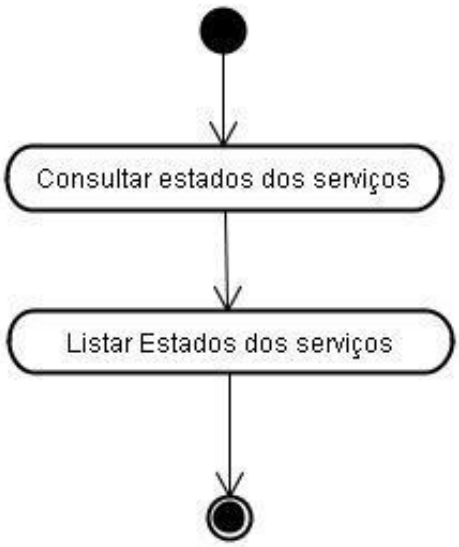

Fig. 05 – Diagrama de Atividade de Estado dos Serviços.

Nessa tela será possível ver os serviços agrupados por similaridade com outros serviços, facilitando a identificação. A Figura abaixo exibe a tela do aplicativo:

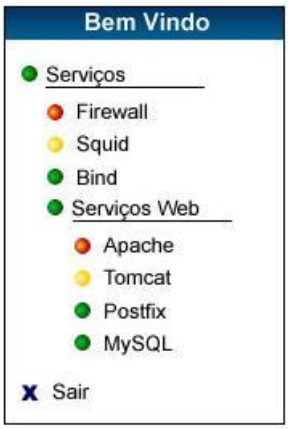

Fig. 06 - Tela que lista estado dos aplicativos.

Com os serviços listados na tela, o administrador pode navegar através do teclado do celular até o nome do serviço ao pressionar a tecla “OK” do dispositivo móvel. O aplicativo irá abrir uma outra somente com operações do serviço tais como:

- Iniciar – Caso o serviço esteja parado ele pode iniciar o serviço;
- Reiniciar – Caso o serviço esteja rodando ele pode reiniciar;
- Parar – Caso o serviço esteja rodando ele pode parar.

Caso o serviço esteja rodando a opção de iniciar estará desabilitada, nesse momento fica possível as opções de reiniciar e de parar. Caso o serviço esteja parado fica disponível a opção somente a opção de iniciar. O diagrama de atividade abaixo ilustra:

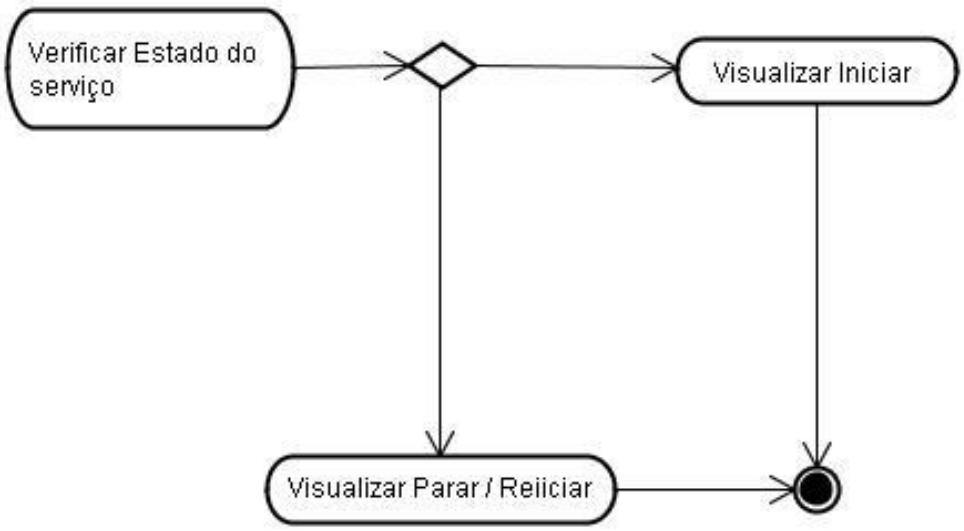

Fig. 06 – Diagrama de atividade de Operações do serviço.

No final de todas as telas, exceto a de autenticação, o administrador do sistema pode encerrar a sessão, basta navegar através do teclado do dispositivo móvel até o final do aplicativo, lá ele deve encontrar a opção “sair”. A opção encerra todas as sessões abertas no servidor com o aplicativo.

RESULTADOS

Com o resultado que obtivemos no presente trabalho, pudemos verificar que é possível, através de uma correta configuração do servidor e com o desenvolvimento de um aplicativo específico, administrar o servidor através de quaisquer dispositivos móveis que tenham acesso à internet.

É interessante observarmos a necessidade da instalação e configuração do servidor com o aplicativo Sudo e a criação dos Shell Script específicos para cada processo na interação com o aplicativo. Esses Shell Scripts que irão realizar as tarefas administrativas no servidor, portanto, torna-se necessário o correto desenvolvimento do mesmo.

Outra observação importante a ser feita no presente trabalho é a correta configuração do Sudo afinal de evitar falhas de segurança. Devemos especificar o local onde encontram-se os Shell Scripts e definir quais tarefas podem ser executadas por um determinado usuário, tornando assim o aplicativo mais seguro.

Podemos pensar que, o desenvolvimento do módulo do aplicativo que irá executar o Shell Script e retornar ao administrador com o resultado da execução do Shell Script, possibilitará uma completa administração do servidor via dispositivo móvel.

Como melhorias deixamos a implementação do modulo que irá da integração do sistemas com os Shell Scripts. Atualmente a implementação desse modulo não segue com o propósito inicial, em um desenvolvimento MVC (Model View Controller).

Também torna-se necessária a criação de outros Shell Script que irão realizar verificar estado da memória, espaço em disco e processamento do servidor.

CONCLUSÃO

O que nos interessou neste trabalho foi a possibilidade de administrar servidores via dispositivos móveis.

Neste sentido, desafiou-nos o desenvolvimento final do aplicativo que o trabalho possibilita apresentar. Buscamos então viabilizar o acesso as rotinas administrativas do servidor via dispositivo móvel.

Podemos também complementar que apesar do presente trabalho desenvolvido em PHP poderá ter seu desenvolvimento em quaisquer outras linguagens de plataforma Web executados em ambientes Linux.

A contribuição deste trabalho é trazer a facilidade de administrar o servidor a longas distâncias viabilizando possíveis resoluções de problemas.

Esperamos que as limitações deste trabalho não invalidem sua contribuição para a tecnologia de informação. Tentamos não delimitar, não definir uma área de desenvolvimento, digamos que o desenvolvimento em nosso trabalho talvez traga possibilidades da interação entre administrador e servidor.

Portanto, o desenvolvimento deste trabalho nos remete que a adequada configuração do servidor com o aplicativo possibilita uma interação entre ambos. Para que aconteça uma interação entre a administração do servidor e do dispositivo móvel, a Internet é o principal veículo.