

# Apache + PHP + MySQL

Fernando Lozano

Consultor Independente

Prof. Faculdade Metodista Bennett

Webmaster da Free Software Foundation

[fernando@lozano.eti.br](mailto:fernando@lozano.eti.br)

**SEPAI 2001**

# O Que São Estes Softwares

Apache:

Servidor web

[www.apache.org](http://www.apache.org)

PHP:

Interpretador de Server-Side Scripts

[www.php.net](http://www.php.net)

MySQL:

Servidor de Bancos de Dados Relacionais

[www.mysql.org](http://www.mysql.org)

# Para Que Servem?

Para a criação de aplicações web

Comércio Eletrônico

Intranets

Portais

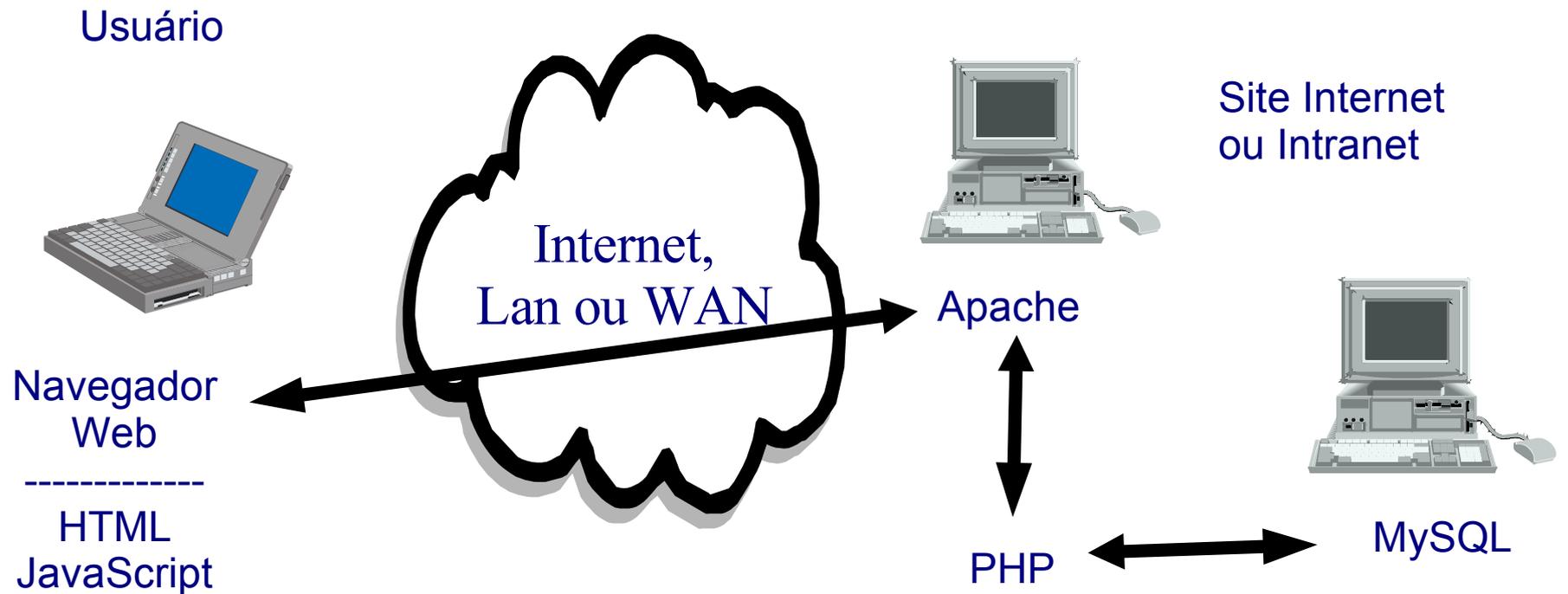
Webmail

Workflow / Groupware

Sistemas de Informação

...

# Como Funcionam?



# Qual o Papel de Cada Um?

Navegador Web:  
Interface com o usuário

Apache:  
Conectividade, segurança e auditoria

PHP:  
Regras de negócio, lógica da aplicação

MySQL:  
Acesso e armazenamento de Informações

# Porque Utilizar o Apache

É o líder do mercado, com mais de 60% dos sites de toda a Internet

É o mais seguro, não foi afetado por Code Red, Ninda, Sircam, ...

Criado para atender às demandas dos maiores provedores e sites web

Não tem concorrentes quanto à riqueza de recursos, estabilidade, performance, ...

# Sobre o Apache

Surgiu quando grandes provedores, insatisfeitos com os servidores web disponíveis, modificaram o NCSA httpd

“A PAtCHEd web server”

Suporta centenas de domínios virtuais simultâneos, criptografia (SSL), certificados digitais, controle de acesso, proxy e redirecionamento, entre outros features

Extensível pela Module API

# Porque Utilizar o PHP

Foi criado do zero para a Internet em vez de ser a adaptação de uma tecnologia anterior

Forncece controles sobre o uso de memória, arquivos, CPU, uploads, etc, de modo que uma aplicação com bugs não poderá derrubar o servidor web

Fácil aprendido e com recursos de OO

Grande quantidade de módulos e bibliotecas prontas disponíveis

# Sobre o PHP

Utiliza uma linguagem que é uma simplificação e extensão do Perl

Respeita as permissões do sistema de arquivos mesmo rodando como módulo

Interpretado, mas podemos obter performance de código compilado utilizando add-ons como o Zend Optimizer, o PHP Accelerator e outros

Extensível por módulos escritos em C

# Módulos do PHP

Suporte a protocolos como SNMP, SMTP, POP, IMAP, LDAP, ...

Geração de documentos XML, WML, PDF, SWF (Flash), GIF, PNG

Acesso nativo a bancos de dados

Criptografia e compressão de dados

# Porque Utilizar o MySQL

Baixa demanda de recursos de hardware

Simplicidade de administração

Protocolo de rede enxuto

É o mais veloz do mercado

Otimizado para as aplicações típicas da web, onde ocorrem mais consultas do que atualizações

# Sobre o MySQL

O MySQL foi criado para ser extremamente ágil, por isso decidiu não implementar recursos que geram muito overhead como triggers, locks, subselects e níveis de isolamento

A arquitetura baseada em múltiplos threads garante performance sob acesso concorrente

# Porque Utilizar os Três?

Porque fornecem tudo o que você necessita para construir aplicações e sites web sofisticados, e de maneira mais simples do que outras alternativas

Porque não trazem elevados custos de licença de *run-time*, por CPU ou por usuário

Porque fornecem produtividade, performance, escalabilidade, segurança e confiabilidade

Principalmente porque são **software livre!**

# Tenho Que Migrar Para GNU/Linux?

Não, pois tanto o Apache quanto o MySQL e o PHP funcionam nativamente em qualquer versão Windows, desde o 95 até o XP

Você pode utilizar outros sistemas Unix como o Solaris, AIX, HP-UX, FreeBSD, ...

Você também pode utilizar em OS/2 e MacOS/X

# Posso Desenvolver Aplicações Comerciais?

Não existe nada contra a comercialização de software livre, o que você não pode fazer é restringir a cópia e uso do software livre  
Software Comercial  $\neq$  Software Proprietário

Programas gerados com ferramentas livres não tem que ser distribuídos como software livre -- você pode criar e vender soluções proprietárias com Apache, PHP e MySQL

# Tenho Que Utilizar os Três Em Conjunto ?

Não, esta é apenas uma combinação popular na Internet. Outras alternativas seriam:

Apache com Perl, Python, ASP, Java, ...

PHP com IIS, PWS, Xitami, ...

PHP com PostgreSQL, Interbase, Oracle, DB2, MS SQL Server ...

MySQL com Perl, Python, ASP, Java, ...

# Quando Não Utilizar o Apache

Quando você sentir falta das paradas causadas por vírus

Quando você precisar justificar upgrade de hardware

Quando você tiver que gastar o orçamento anual do departamento

# Quando Não Utilizar o PHP

Quando necessitar de performance a todo custo -- utilizar aplicações compiladas ou construir aplicações distribuídas

Quando necessitar de recursos OO mais sofisticados para organizar projetos de grande envergadura

Quando necessitar de *verdadeiros* projetos em três camadas -- mas o PHP pode ser integrado com um *middle tier* em C ou em Java

# Quando Não Utilizar o MySQL

Quando necessitar de recursos SQL avançados, como stored procedures e views

Em aplicações transacionais (muitas inclusões / atualizações)

Em aplicações OLAP (DataMarts, BI, ...)

Quando utilizar bancos de dados muito grandes, que necessitem ser espalhados em vários discos ou mesmo em vários servidores (bancos distribuídos)

# Falsos Mitos Sobre o MySQL

Não suporta transações

(este recurso pode ser habilitado ou não para cada tabela)

Não permite replicação

(a capacidade de replicação é fornecida por programas externos, mas específicos para o MySQL, de acordo com a cultura Unix)

# Foco no PHP

O desenvolvedor não se preocupa com o Apache, que é responsabilidade do administrador de rede

O desenvolvedor não se preocupa com o MySQL, pois ele pode ser trocado por qualquer banco SQL

Aplicações PHP funcionam com poucas ou nenhuma modificação com outros bancos, e não são afetadas pelo servidor web

# Demo do PHP

Vamos ver algumas características do desenvolvimento em PHP através de exemplos