

Uma plataforma para pequenos e médios clientes utilizando a plataforma LADP em substituição ao LAMP

Igor Vitório Custódio <igorvc@gmail.com>

Introdução:

Sabe-se que a maioria das empresas de pequeno e médio porte com negócios baseados na internet procuram soluções simples, porém eficientes para que fazerem seus projetos de maneira fácil, rápida, com menor custo possível.

E nesta via que esta sendo utilizada a plataforma denominada LAMP que significa:

- L - Linux como Sistema Operacional base, devido a sua robustez, segurança e estabilidade, além de ser grátis;
- A - Apache como servidor para aplicações Web, pelos mesmos motivos do uso de Linux;
- M - MySQL, pela facilidade de uso, pela disponibilidade e por ser grátis;
- P - Linguagem de programação, em muitos casos utiliza-se PHP, pelo suporte nativo a MySQL, ou outras, também iniciadas por P, a saber Perl e Python.

Esta plataforma esta enraizada em quase todos os servidores de hospedagens disponíveis, sendo que ela se tornou um sinônimo de desenvolvimento rápido e com poucos recursos.

O intuito deste artigo não e mostrar que esta plataforma não e eficiente, ou que ela e a melhor, o intuito e mostrar que existem opções, que em determinados aspectos podem ser muito mais interessantes quando se tem em vista questões como escalabilidade, estabilidade suporte entre outros.

E com esta idéia que propomos a analise de uma nova plataforma a LABP com LAP tendo o mesmo significado da plataforma anterior, formando a base do sistema, trocando o M por B, que em suma significa:

B - Plataforma DB2 Express C da IBM: uma plataforma de desenvolvimento e um banco de dados relacional com suporte a XML nativo (Versão Viper), maiores informações visite...

Muitas pessoas podem se perguntar o porque migrar de um Banco de dados Livre para um proprietário que garante a licença de distribuição?

As respostas podem ser medidas em termos de confiança/responsabilidade, isto não quer dizer que os bancos de dados livres não possuem, porém ter um background de uma empresa e um conhecimento de dezenas de anos na área com certeza é um diferencial.

Por exemplo, quando pensamos em transações pela Internet nos vem sempre a cabeça um tal de XML (Extensible Markup Language), porém pensar em um banco de dados livre que suporte este tipo de funcionalidade é complicado, uma vez que grandes bancos pagos, como o Oracle, também não o tem!

Muitas pessoas podem pensar de uma maneira alternativa e conseguir utilizar um banco de dados relacional e "simular" o suporte a XML. Este e o caso de possuímos em uma tabela um campo "nome de arquivo" que aponta para um arquivo XML real.

Assim, para fazer acesso ao arquivo XML bastaria fazer a query que estamos acostumados e depois abrir o arquivo XML resultante.

Porém pense no caso que você tenha que pesquisar algo em todos os arquivos XML, ou seja, você quer que sejam retornados os valores de determinada condição dentro do arquivo XML. Muitos diriam que e trivial, basta fazer o select * e tratar arquivo por arquivo XML montando uma lista com os dados que necessitamos.

Pois é, trivial, porém computacionalmente inviável! Com certeza sua aplicação pode funcionar, porém, falhará em conceitos como desempenho/escalabilidade/flexibilidade, uma vez que estamos tratando de arquivos fisicamente armazenados em disco, além de que o "tratamento" deve ser um código fixo, se mudar a query, devemos reprogramar todas as funções.

E, até os mais críticos devem dar o braço a torcer. E aí que entra o DB2, o primeiro banco de dados a implementar suporte nativo a XML. Com ele conseguimos fazer desde tabelas com campos Varchar até campos XML, que contém um arquivo XML!

Além disto, conseguimos indexar estes tipos de arquivos e fazer buscas como se fossem selects normais.

Poderíamos ficar aqui páginas e páginas mostrando as funcionalidades deste banco de dados, porém caso queira maiores informações visite o site da IBM sobre o assunto.

Convém citar o tipo de licença que a IBM lançou este Banco de dados. Ela fez uma licença que garante aos desenvolvedores e empresas a não necessidade de pagamento da licença base (DB2 Express C), que possui todas as funcionalidades necessárias às pequenas e medias empresas, sendo somente necessário o registro de tal distribuição via formulário de download, muito parecido com as versões grátis que possuímos hoje, porem com algumas outras vantagens:

1) Caso queira escalabilidade, basta "comprar" uma versão mais ampla do sistema, o que não significa que a versão "livre" seja limitada, o que significa e que ela não possui todas as ferramentas disponíveis nas versões pagas, já que o núcleo da aplicação e o

mesmo, diferenciando, além do preço, funcionalidades agregadas, como clusterização, disponibilidade, etc.

2) Suporte, seguindo a filosofia da comunidade livre. A IBM vendo o grande avanço das comunidades lançou um fórum livre, do qual participa toda a comunidade de desenvolvedores e de utilizadores deste banco de dados, o que garante um suporte eficiente e até surpreendente em alguns casos. O interessante é que participam deles os empregados da IBM, que pode significar uma ajuda de alto nível até então gratuita.

Claro, quando necessitar de alguma coisa mais dirigida, basta comprar uma versão superior à utilizada neste artigo e terá toda a infra-estrutura da IBM a seu dispor.

3) Anos de concorrência: Sabemos que a concorrência é o motor do capitalismo, assim, para uma empresa estar ativa a tantos anos no mercado e porque ela é boa este é a idéia por trás de confiar no produto desta empresa.

4) Ferramentas gráficas de altíssimo nível: Sabe-se que estas empresas investem muito em ferramentas visuais, e a IBM não fica para trás no assunto, possuindo diversos aplicativos que facilitam e muito a vida do programador/administrador do banco de dados.

Muitos podem pensar o porque da IBM adotar esta postura agora, liberando uma versão plenamente utilizável de um de seus sistemas mais importantes. A reflexão é profunda, mas vejamos, não há como avançar num mercado existente sem ter uma solução parecida, um ambiente parecido, e aí que entra a licença! Já que os bancos de dados livres imperam neste seguimento.

Modificações:

Bem, como grande parte da plataforma se manteve, reservamos o direito à não explicar como se instala/configura o LA. Seguindo somente as operações necessárias para a adequação das outras duas letras ao nosso objetivo.

B - Instalação e configuração do DB2:

Existem diversos tutoriais pela internet ensinando como se instalar.

Porém, convém citar que o programa de instalação gráfico é auto-explicativo e muito fácil de executar!

Maiores informações em:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.doc/doc/t0024287.htm>

P - Configuração do PHP para suporte a linguagem.

Todas as operações necessárias para adequação do PHP ao novo banco de dados estão também presentes no site da IBM em:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.apdv.php.doc/doc/t0011926.htm>

Estudo de caso:

Sistema de avisos do Portal @Comp (<http://www.comp.ufscar.br>).

Este sistema tem as seguintes funcionalidades:

- Permitir login e logout;
- Permitir a inclusão e edição de usuários;
- Permitir a inclusão de avisos;
- Permitir a remoção lógica de avisos, aos respectivos donos
- Permitir a edição de avisos apenas aos donos dos avisos;
- Permitir a visualização de avisos ativos.

Estão presentes no Apêndice os algoritmos do sistema.

Este sistema foi implementado e está atualmente rodando com base em MySQL (Antes). O objetivo foi ver se é possível, de uma forma simples, converter a aplicação atual para o DB2 Express C (Depois).

Lembrando que somente foram mostradas as partes do código modificadas!

Convém lembrar que este não é um tutorial de conversão, porém dá uma idéia de que ela é possível e dá dicas de como realiza-la.

São utilizadas poucas das funções principais oferecidas pela biblioteca, porém foram os suficientes para que uma aplicação real fosse migrada.

A conclusão que podemos chegar depois deste exemplo é que é sim possível, veja o resultado da migração e as modificações necessárias:

Arquivo config.php	
Antes:	Depois:
<pre><code>\$usuario_db = "usuario"; \$senha_db = "senha"; \$servidor_db = "localhost"; \$base_db = "avisos"; \$conexao = mysql_pconnect(\$servidor_db, \$usuario_db, \$senha_db) or die("Erro!");</code></pre>	<pre><code>\$usuario_db = ""; \$senha_db = ""; \$base_db = "PROXY"; //Realiza conexão e deixa ela em \$conexao \$conexao = db2_connect(\$base_db, \$usuario_db, \$senha_db);</code></pre>

mysql_select_db(\$base_db, \$conexao) or die("Erro!");	if (!\$conexao) { echo "Erro: " . db2_conn_error() . " " . db2_conn_errormsg() ; }
---	---

Verificamos uma modificação mínima no arquivo que será utilizado como base para todos os outros que requerem conexões, para tanto devem incluir este arquivo.

Arquivo login_vai.php	
Antes:	Depois:
<pre> \$SQL = "SELECT id, nome, login, senha FROM usuario WHERE login = " . \$login . """; \$result_id = @mysql_query(\$SQL) or die("Erro no banco de dados!"); \$total = @mysql_num_rows(\$result_id); // Caso o usuário tenha digitado um login válido o número de linhas será 1.. if(\$total) { // Obtém os dados do usuário, para poder verificar a senha e passar os demais dados para a sessão \$dados = @mysql_fetch_array(\$result_id); // Agora verifica a senha if(!strcmp(\$senha, \$dados["senha"])) { // TUDO OK! Agora, passa os dados para a sessão e redireciona o usuário \$_SESSION["id_usuario"] = \$dados["id"]; \$_SESSION["nome_usuario"] = stripslashes(\$dados["nome"]); header("Location: admin.php"); exit; } // Senha inválida else { header("Location: login.php?msg=1"); exit; } } // Login inválido else { header("Location: login.php?msg=1"); exit; } </pre>	<pre> \$SQL = "SELECT * FROM "SYSTEM".USUARIOS AS USUARIOS WHERE USUARIOS.LOGIN = " . \$login . """; //Verificando se o usuário é válido! \$stmt = db2_exec(\$conexao, \$SQL, array('cursor' => DB2_SCROLLABLE)); while (\$dados = db2_fetch_array(\$stmt)) { if(!strcmp(\$senha, \$dados[3])) { // TUDO OK! Agora, passa os dados para a sessão e redireciona o usuário \$_SESSION["id_usuario"] = \$dados[0]; \$_SESSION["nome_usuario"] = stripslashes(\$dados[1]); header("Location: admin.php"); } // Senha inválida else { header("Location: login.php?msg=1"); exit; } } </pre>

Verificamos que novamente as modificações foram mínimas.

Arquivo avisos.php	
Antes:	Depois:
\$SQL = "SELECT avisos.id as aid, avisos.titulo,	\$SQL = "SELECT AVISOS.ID,

<pre> avisos.conteudo, avisos.data, avisos.autor_id, avisos.ativo, usuario.login, usuario.nome, usuario.id FROM avisos, usuario WHERE usuario.id = avisos.autor_id ORDER BY data DESC"; \$result_id = mysql_query(\$SQL) or die(mysql_error()); \$total = mysql_num_rows(\$result_id); if(\$total) { // Efetua o loop no banco de dados while(\$dados = mysql_fetch_array(\$result_id)) { if (\$dados["ativo"] == 0){ // só mostra se o aviso estiver ativo! // Abre tabela HTML echo "<table border=1 cellpadding=3 cellspacing=0 width='100%' >\n"; echo "<tr> <td>Autor</td>"; echo "<td>". \$dados["nome"] . "</td></tr>"; echo "<tr> <td>Data</td>"; echo "<td>". date("d/m/Y à\s H:i:s", \$dados["data"]) . "</td></tr>"; echo "<tr> <td>Título</td>"; echo "<td>". stripslashes(\$dados["titulo"]) ."</td>"; echo "</tr>"; echo "<tr> <td>Aviso</td>"; echo "<td>". \$dados["conteudo"] . "</td>"; echo "</tr>"; // fecha tabela echo "</table>\n"; echo "

"; } } else { echo "Nenhum aviso cadastrado!\n"; } </pre>	<pre> AVISOS.TITULO, AVISOS.AUTOR_ID, AVISOS.\\"DATA\", AVISOS.ATIVO, AVISOS.CONTEUDO, USUARIOS.NOME, USUARIOS.ID, USUARIOS.LOGIN FROM \\"SYSTEM\".AVISOS AS AVISOS, \\"SYSTEM\".USUARIOS AS USUARIOS WHERE AVISOS.AUTOR_ID = USUARIOS.ID ORDER BY AVISOS.\\"DATA\" DESC"; \$sachou = FALSE; \$stmt = db2_exec(\$conexao, \$SQL, array('cursor' => DB2_SCROLLABLE)); // Efetua o loop no banco de dados while (\$dados = db2_fetch_array(\$stmt)) { if (\$dados[4] == 0){ // só mostra se o aviso estiver ativo! // Abre tabela HTML echo "<table border=1 cellpadding=3 cellspacing=0 width='100%' >\n"; echo "<tr> <td>Autor</td>"; echo "<td>". \$dados[6] ."</td></tr>"; echo "<tr> <td>Data</td>"; echo "<td>". date("d/m/Y à\s H:i:s", \$dados[3]) . "</td></tr>"; echo "<tr> <td>Título</td>"; echo "<td>". stripslashes(\$dados[1]) ."</td>"; echo "</tr>"; echo "<tr> <td>Aviso</td>"; echo "<td>". \$dados[5] . "</td>"; echo "</tr>"; // fecha tabela echo "</table>\n"; echo "

"; \$sachou = TRUE; } } if (!\$sachou) { echo "Nenhum aviso cadastrado!\n"; } </pre>
--	---

Arquivo admin.php

Antes:	Depois:
<pre> // Verifica e imprime quantidade de notícias no nome do usuário \$SQL = "SELECT id FROM avisos WHERE autor_id = " . \$_SESSION["id_usuario"]; \$result_id = mysql_query(\$SQL) or die(mysql_error()); \$total = mysql_num_rows(\$result_id); </pre>	<pre> // Verifica e imprime quantidade de notícias no nome do usuário \$SQL = "SELECT id FROM avisos WHERE autor_id = " . \$_SESSION["id_usuario"]; //Varrendo o resultado e contando: </pre>

<pre> if(\$total) { echo "Há um total de " . \$total . " aviso(s) de sua autoria!\n"; } else { echo "Não há nenhum aviso de sua autoria!\n"; } </pre>	<pre> \$total = 0; \$stmt = db2_exec(\$conexao, \$SQL, array('cursor' => DB2_SCROLLABLE)); while (\$dados = db2_fetch_array(\$stmt)) { \$total++; } if(\$total) { echo "Há um total de " . \$total . " aviso(s) de sua autoria!\n"; } else { echo "Não há nenhum aviso de sua autoria!\n"; } </pre>
<pre> \$id_pes = \$_SESSION["id_usuario"]; // so pode editar o usuário dele! \$SQL = "SELECT * FROM usuario where id=" . \$id_pes . ";"; \$result_id = mysql_query(\$SQL) or die(mysql_error()); \$total = mysql_num_rows(\$result_id); if(\$total) { // Efetua o loop no banco de dados while(\$dados = mysql_fetch_array(\$result_id)) { echo "<form method='post' action='admin.php?opc=7' name='salvaUsuario'OnSubmit='JavaScript:return checkdata()' >"; echo "Edição do usuário
"; echo "<center>"; echo "<table>"; echo "<tr>"; echo " <td> Nome Completo: </td>"; echo " <td> <input type='text' name='nome' size='43' value='\" . \$dados["nome"] . "\"> </td>"; echo "</tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr>"; echo " <td> Login: </td>"; echo " <td> <input type='text' name='login' size='20' value='\" . \$dados["login"] . "\"> </td>"; echo "</tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr>"; echo " <td> Senha: </td>"; echo " <td> <input type='password' name='senha' size='20' > </td>"; </pre>	<pre> \$id_pes = \$_SESSION["id_usuario"]; // so pode editar o usuário dele! \$SQL = "SELECT * FROM \"SYSTEM\".USUARIOS WHERE id = " . \$id_pes ; //Varrendo o resultado e contando: \$sachou = FALSE; \$stmt = db2_exec(\$conexao, \$SQL, array('cursor' => DB2_SCROLLABLE)); // Efetua o loop no banco de dados while (\$dados = db2_fetch_array(\$stmt)) { echo "<form method='post' action='admin.php?opc=7' name='salvaUsuario'OnSubmit='JavaScript:return checkdata()' >"; echo "Edição do usuário
"; echo "<center>"; echo "<table>"; echo "<tr>"; echo " <td> Nome Completo: </td>"; echo " <td> <input type='text' name='nome' size='43' value='\" . \$dados[1] . "\"> </td>"; echo "</tr>"; echo "<tr> <td> </td> <td> </td>"; </tr>"; echo "<tr>"; echo " <td> Login: </td>"; echo " <td> <input type='text' name='login' size='20' value='\" . \$dados[2] . " "\"> </td>"; echo "</tr>"; echo "<tr> <td> </td> <td> </td>"; </tr>"; echo "<tr>"; </pre>

<pre> echo " </tr>"; echo " <tr> <td> </td> <td> </td> </tr>"; echo " <tr> <td> </td> <td> <center> <input type=\"submit\" name=\"Salvar\" value=\"Salvar\" </center> </td> </tr>"; echo " </table>"; echo " </center>"; echo " </form>"; } // fim while } // fim if do SQL </pre>	<pre> echo " <td> Senha: </td>"; echo " <td> <input type=\"password\" name=\"senha\" size=\"20\" > </td>"; echo " </tr>"; echo " <tr> <td> </td> <td> </td> </tr>"; echo " <tr> <td> </td> <td> <center> <input type=\"submit\" name=\"Salvar\" value=\"Salvar\" > </center> </td> </tr>"; echo " </table>"; echo " </center>"; echo " </form>"; \$achou = TRUE; } // fim while </pre>
<pre> // Só o autor pode editar o seu texto! \$id_avisos = \$_GET["avisos"]; \$SQL = "SELECT * FROM avisos where id=" . \$id_avisos . ";"; \$result_id = mysql_query(\$SQL) or die(mysql_error()); \$total = mysql_num_rows(\$result_id); if(\$total) { // Efetua o loop no banco de dados while(\$dados = mysql_fetch_array(\$result_id)) { if(\$_SESSION["id_usuario"] == \$dados["autor_id"]){ echo " <form method=\"post\" action=\"admin.php?opc=5\" name=\"salvaAviso\" >"; echo " <input type=\"hidden\" name=\"id_ori\" value=\"". \$dados["id"] . "\>"; echo " <center>"; echo " <table>"; echo " <tr>"; echo " <td> Aviso de Número: </td>"; echo " <td> ". \$dados["id"] . " "; </td>"; echo " </tr>"; echo " <tr> <td> </td> <td> </td> </tr>"; echo " <tr>"; echo " <td> Título: </td>"; echo " <td> <input type=\"text\" name=\"titulo\" size=\"46\" value=\"". \$dados["titulo"] . "\" > </td>"; echo " </tr>"; echo " <tr> <td> </td> <td> </td> </tr>"; echo " <tr>"; echo " <td> Aviso: </td>"; echo " <td> <textarea name=\"campoAviso\" cols=\"44\" rows=\"5\" wrap=\"VIRTUAL\" id=\"campoAviso\"> . \$dados["conteudo"] . " </textarea> </td>"; echo " </tr>"; </pre>	<pre> // Só o autor pode editar o seu texto! \$id_avisos = \$_GET["avisos"]; \$SQL = "SELECT * FROM \"SYSTEM\".AVISOS WHERE id = " . \$id_avisos ; \$achou = FALSE; \$stmt = db2_exec(\$conexao, \$SQL, array('cursor' => DB2_SCROLLABLE)); // Efetua o loop no banco de dados while (\$dados = db2_fetch_array(\$stmt)) { if(\$_SESSION["id_usuario"] == \$dados[3]){ echo " <form method=\"post\" action=\"admin.php?opc=5\" name=\"salvaAviso\" >"; echo " <input type=\"hidden\" name=\"id_ori\" value=\"". \$dados[0] . "\>"; echo " <center>"; echo " <table>"; echo " <tr>"; echo " <td> Aviso de Número: </td>"; echo " <td> ". \$dados[0] . " </td>"; echo " </tr>"; echo " <tr> <td> </td> <td> </td> </tr>"; echo " <tr>"; echo " <td> Título: </td>"; echo " <td> <input type=\"text\" name=\"titulo\" size=\"46\" value=\"". \$dados[1] . "\" > </td>"; echo " </tr>"; echo " <tr> <td> </td> <td> </td> </tr>"; echo " <tr>"; echo " <td> Aviso: </td>"; echo " <td> <textarea name=\"campoAviso\" cols=\"44\" rows=\"5\" wrap=\"VIRTUAL\" id=\"campoAviso\"> . \$dados[2] . </pre>

<pre> echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr> <td> </td> </tr>"; <td><center><input type="submit" name="Salvar" value="Salvar"></center></form> </td> </tr>"; echo "<tr> <td><form method="post" action="admin.php?opc=9" name="deletaAviso" ></td>"; echo "<td><input type="hidden" name="id_ori" value="". \$dados["id"] . "\">"; echo "<center><input type="submit" name="Apagar" value="Apagar"></center> "; echo "</form>"; echo "</td> </tr>"; echo "</table>"; echo "</center>"; } else { echo "<h2> Visualização do Aviso: </h2>"; echo "<center><table>"; echo "<tr><td>ID do aviso:</td><td> " . \$dados["id"] . "</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr><td>Título do Aviso:</td><td> " . stripslashes(\$dados["titulo"]) ."</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr><td>Aviso:</td><td> " . \$dados["conteudo"] . "</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr><td>ID do autor: </td><td> " . \$dados["autor_id"] . "</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "</table> </center>

"; } // fim if } // fim while </pre>	<pre> "</textarea> </td>"; echo "</tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr> <td> </td> </tr>"; <td><center><input type="submit" name="Salvar" value="Salvar"></center></form> </td> </tr>"; echo "<tr> <td><form method="post" action="admin.php?opc=9" name="deletaAviso" ></td>"; echo "<td><input type="hidden" name="id_ori" value="". \$dados[0] . "\">"; echo "<center><input type="submit" name="Apagar" value="Apagar"></center> "; echo "</form>"; echo "</td> </tr>"; echo "</table>"; echo "</center>"; \$sachou = TRUE; } else { echo "<h2> Visualização do Aviso: </h2>"; echo "<center><table>"; echo "<tr><td>ID do aviso:</td><td> " . \$dados[0] . "</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr><td>Título do Aviso:</td><td> " . stripslashes(\$dados[1]) ."</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr><td>Aviso:</td><td> " . \$dados[2] . "</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; echo "<tr><td>ID do autor: </td><td> " . \$dados[3] . "</td></tr>"; echo "<tr> <td> </td> <td> </td> </tr>"; \$sachou = TRUE; echo "</table> </center>

"; } // fim if } // fim while </pre>
<pre> \$id = \$_REQUEST["id_ori"]; \$title = \$_REQUEST["titulo"]; \$aviso = \$_REQUEST["campoAviso"]; \$sql = "UPDATE `avisos` SET `titulo` = " . \$title . " , `conteudo` = " . \$aviso . " , `ativo` = '0' WHERE id = " . \$id . " "; \$result_id = mysql_query(\$sql) or die(mysql_error()); echo "<h2> Aviso de Título: " . \$title . " foi atualizado com êxito!</h2>"; </pre>	<pre> \$id = \$_REQUEST["id_ori"]; \$title = \$_REQUEST["titulo"]; \$aviso = \$_REQUEST["campoAviso"]; \$sql = "UPDATE \"SYSTEM\".AVISOS AVISOS SET TITULO = " . \$title . " , CONTEUDO = " . \$aviso . " , ATIVO = 0 WHERE AVISOS.ID = " . \$id; \$stmt = db2_exec(\$conexao, \$sql, array('cursor' => DB2_SCROLLABLE)); echo "<h2> Aviso de Título: " . \$title . " foi atualizado com êxito!</h2>"; </pre>
<pre> \$nome = \$_REQUEST["nome"]; \$login = \$_REQUEST["login"]; \$senha = \$_REQUEST["senha"]; </pre>	<pre> \$nome = \$_REQUEST["nome"]; \$login = \$_REQUEST["login"]; \$senha = \$_REQUEST["senha"]; </pre>


```

die(mysql_error());
$total = mysql_num_rows($result_id);

if($total)
{
    // Abre tabela HTML
    echo "<center>";
    echo "<table border=1 cellpadding=3
cellspacing=0>\n";
    echo
"<tr><th>Id</th><th>Titulo</th><th>Estado</th><th>Dat
a</th><th>Autor</th></tr>\n";

    // Efetua o loop no banco de dados
    while($dados = mysql_fetch_array($result_id))
    {
        echo "<tr><td> . $dados[\"aid\"] .
</td><td><a href=\"admin.php?opc=4&aviso=
\" . $dados[\"aid\"] . \"><font color=\\\"#0000FF\\\"><center>\" .
stripslashes($dados[\"titulo\"] . \"</center></font></a>
</td>";
        echo "<td> <center>";
        if ($dados[\"ativo\"] == 0){
            echo "<font color=\\\"green\\\">Ativo
</font>";
        } else {
            echo "<font color=\\\"#FF0000\\\">Inativo
</font>";
        }
        echo "</center></td>";
        echo "<td><center> . date(\"d/m/Y às H:i:s\",
$dados[\"data\"] . \"</center> </td> <td><center> \" .
$dados[\"nome\"] . \"</center> </td>";
        echo "</tr>\n";
    }

    // Fecha tabela
    echo "</table>\n";
    echo "</center>";
    echo "<br><br>";
    echo "<font color=\\\"green\\\">Ativo </font>
Significa que o aviso aparecerá na página de
aviso!<br>";
    echo "<font color=\\\"#FF0000\\\">Inativo </font>
significa que o aviso não aparecerá na página de avisos.
Para restaurá-lo basta clicar no título dele e proceder o
salvamento!";
    echo "<h3>Convém ressaltar que a edição de
avisos só é permitido ao autor!</h3>";
    echo "<br><br>";

}
else
{
    echo "<B>Nenhum aviso cadastrado!</B>\n";
}

```

```

WHERE AVISOS.AUTOR_ID =
USUARIOS.ID ORDER BY AVISOS.\"DATA\"
DESC";

$sachou = FALSE;

$stmt = db2_exec($conexao, $SQL, array('cursor' =>
DB2_SCROLLABLE));
// Efetua o loop no banco de dados
// Abre tabela HTML
echo "<center>";
echo "<table border=1 cellpadding=3
cellspacing=0>\n";
echo
"<tr><th>Id</th><th>Titulo</th><th>Estado</th><th>
Data</th><th>Autor</th></tr>\n";
while ($dados = db2_fetch_array($stmt))
{
    echo "<tr><td> . $dados[0] .
</td><td><a href=\"admin.php?opc=4&aviso=
\" . $dados[0] . \"><font color=\\\"#0000FF\\\"><center>\" .
stripslashes($dados[1]) . \"</center></font></a>
</td>";
    echo "<td> <center>";
    if ($dados[4] == 0){
        echo "<font color=\\\"green\\\">Ativo
</font>";
    } else {
        echo "<font
color=\\\"#FF0000\\\">Inativo </font>";
    }
    echo "</center></td>";
    echo "<td><center> . date(\"d/m/Y às
H:i:s\", $dados[3]) . \"</center> </td> <td><center> \" .
$dados[5] . \"</center> </td>";
    echo "</tr>\n";
    $sachou = TRUE;
}

// Fecha tabela
echo "</table>\n";
echo "</center>";
echo "<br><br>";
echo "<font color=\\\"green\\\">Ativo </font>
Significa que o aviso aparecerá na página de
aviso!<br>";
    echo "<font color=\\\"#FF0000\\\">Inativo
</font> significa que o aviso não aparecerá na
página de avisos. Para restaurá-lo basta clicar no
título dele e proceder o salvamento!";
    echo "<h3>Convém ressaltar que a edição de
avisos só é permitido ao autor!</h3>";
    echo "<br><br>";

if(!$sachou)
{
    echo "<B>Nenhum aviso cadastrado!</B>\n";
}

```

Nestas modificações verificamos alguns pontos importantes:

- A solução utilizada para “contar” o número de linhas que retornaram na pesquisa, em substituição a função do MySQL (*mysql_num_rows*) foi a de contar manualmente, porém outras soluções são possíveis, como o utilizando COUNT na SQL, etc.
- A substituição dos índices dos vetores de “nomes” para índices, que a princípio pode causar uma confusão, porém é facilmente aprendido.

Por exemplo, se possuímos uma tabela usuarios (MySQL) com os seguintes campos: id, nome e senha. Após uma pesquisa geral do estilo SELECT id, nome FROM usuários, para mostrarmos os números de id e nome dos usuários, bastaria fazer:

```
$SQL = "SELECT id, nome FROM usuario ";
$result_id = mysql_query($SQL) or die(mysql_error());
// Efetua o loop no banco de dados
while($dados = mysql_fetch_array($result_id))
{
    echo "Id: " . $dados["id"] . "Nome: " . $dados["nome"];
}
```

Com o DB2, necessitamos “lembrar” que o acesso é feito via vetor numerado. Além de que todo vetor começa na posição 0.

No exemplo:

```
$SQL = "SELECT id, nome FROM \SYSTEM\usuario ";
$stmt = db2_exec($conexao, $SQL, array('cursor' => DB2_SCROLLABLE));
// Efetua o loop no banco de dados
while ($dados = db2_fetch_array($stmt))
{
    echo "Id: " . $dados[0] . "Nome: " . $dados[1];
}
```

Ou seja, os índices do vetor de resposta está na ordem da pesquisa efetuada, como id foi a primeira informação a ser “pesquisada” ela fica na posição 0 do vetor, o nome na posição 1 e assim sucessivamente.

- Convém citar outra função equivalente, *mysql_close()* tem como equivalente a *db2_close()*.

Conclusão:

Conclui-se assim que é plenamente viável a utilização e a migração de pequenas aplicações que estavam rodando na plataforma LAMP para a plataforma LADP.

Esta migração não é complicada, bastando apenas alguns ajustes fáceis de serem feitos, aliados a uma adequação a nova sintaxe do banco de dados.

Isto sem falar nas ferramentas gráficas que auxiliam e muito a utilização do Banco de dados, com uma citação especial ao Centro de Controle, completíssimo e de alto nível! Foi através dele que criei as tabelas utilizadas.

Estas ferramentas dispensam um conhecimento prévio da plataforma, já que possui aplicativos que auxiliam nas atividades mais comuns, seguindo a filosofia do “avançar e concluir”.

Fica aí a sugestão Teste e veja os benefícios!

Apêndice:

Relação de tabelas:

MySQL	DB2
<pre>CREATE TABLE `avisos` (`id` int(11) NOT NULL auto_increment, `titulo` varchar(255) NOT NULL default "", `conteudo` text NOT NULL, `autor_id` int(11) NOT NULL default '0', `data` int(11) NOT NULL default '0', `ativo` smallint(6) default '0', UNIQUE KEY `id` (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1 PACK_KEYS=0 AUTO_INCREMENT=5 ;</pre>	<pre>CREATE TABLE "SYSTEM"."AVISOS" ("ID" INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH +0 INCREMENT BY +1 MINVALUE +0 MAXVALUE +2147483647 NO CYCLE NO CACHE NO ORDER) , "TITULO" VARCHAR(250) NOT NULL , "CONTEUDO" LONG VARCHAR NOT NULL , "AUTOR_ID" INTEGER NOT NULL , "DATA" INTEGER NOT NULL WITH DEFAULT 0 , "ATIVO" SMALLINT WITH DEFAULT 0) IN "USERSPACE1" ; ALTER TABLE "SYSTEM"."AVISOS" ADD CONSTRAINT "CC1149219846453" UNIQUE ("ID"); ALTER TABLE "SYSTEM"."AVISOS" ALTER COLUMN "ID" RESTART WITH 6;</pre>
<pre>CREATE TABLE `usuario` (`id` int(11) NOT NULL auto_increment, `nome` varchar(50) NOT NULL default "", `login` varchar(20) NOT NULL default "", `senha` varchar(40) NOT NULL default "", `ativo` int(11) default '0', PRIMARY KEY (`login`), UNIQUE KEY `id` (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1 PACK_KEYS=0 COMMENT='Tabela de usuarios' AUTO_INCREMENT=4 ;</pre>	<pre>CREATE TABLE "SYSTEM"."USUARIOS" ("ID" INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH +0 INCREMENT BY +1 MINVALUE +0 MAXVALUE +2147483647 NO CYCLE NO CACHE NO ORDER) , "NOME" VARCHAR(50) NOT NULL , "LOGIN" VARCHAR(20) NOT NULL , "SENHA" VARCHAR(40) NOT NULL , "ATIVO" INTEGER NOT NULL WITH DEFAULT 0) IN "USERSPACE1" ; COMMENT ON TABLE "SYSTEM"."USUARIOS" IS 'Tabela</pre>

	de Usuários'; ALTER TABLE "SYSTEM"."USUARIOS" ADD CONSTRAINT "CC1149216012453" PRIMARY KEY ("LOGIN"); ALTER TABLE "SYSTEM"."USUARIOS" ADD CONSTRAINT "CC1149216019859" UNIQUE ("ID"); ALTER TABLE "SYSTEM"."USUARIOS" ALTER COLUMN "ID" RESTART WITH 6;
--	--

Algoritmos do Sistema:

Avisos.php

Fazer uma busca no Banco de dados na tabela avisos pelos registros com coluna "ativo" = 0, ordenado por ordem decrescente de data;
Enquanto houver avisos neste estado, construir uma tabela e colocar os dados do aviso retornados na pesquisa.
FIM

Login.php

Fornecer um formulário de nome de usuário e senha para o utilizador.
Ao submeter, enviar o resultado a login_vai.php
Fim

Login_vai.php

Recebe os dados via POST do formulário da página login.php
Trata estes dados e caso tenha recebido usuário e senha, faz uma pesquisa no Banco de dados na tabela usuários pelo login recebido.
Caso o login e a senha (comparada com seu MD5) estejam cadastrados nas tabelas, salva os dados do usuário na Sessão e redireciona para a página admin.php
Caso contrário, redireciona para a página login.php e sinaliza erro.
Fim

Admin.php

Verifica se o usuário tem permissões para visitar, caso contrário, redireciona para a página de login.
Depois, caso a página tenha sido chamada sem parâmetro (GET) no endereço, construir a página geral de admin, com as seguintes informações:
Busque no banco de dados avisos o número de avisos que possuem no campo id_autor o id do usuário atual, disponibilizando esta informação.

Busque na tabela avisos os seguintes dados: identificação do aviso, título do aviso, data do aviso e identificação do autor do aviso. Com posse desta última informação, recupere o nome completo do autor na tabela usuários e monte uma tabela com as seguintes informações:

Identificação do aviso, Título do aviso, Data de criação e Nome do autor.

Sendo que ao se clicar no título do aviso deve-se chamar a página de administração com o argumento com valor 1.

Apresente os menus de atividade seguintes:

Edição do usuário corrente;

Adição de novo usuário;

Adição de novo Aviso;

Estes menus devem chamar a página de administração com os seguintes valores, 2, 3, e 4 respectivamente.

Fim

Caso tenha vindo como argumento o valor 1, fazer uma pesquisa na tabela de avisos com a identificação do aviso em questão.

Construir um formulário com as informações retornadas do bando de dados para edição

Dar como opção o salvamento e o apagamento do aviso.

Caso opte pelo salvamento, prossegue-se chamando a página de administração com o argumento 5.

Caso opte pelo apagamento, prossegue-se chamando a página de administração com o argumento 6

Fim.

Caso tenha vindo como argumento o valor 2, fazer uma pesquisa na tabela de usuários pela identificação do usuário corrente, fornecendo um formulário para edição das informações buscadas no banco de dados, como nome, login e senha.

Dar como opção o salvamento.

Caso optar pelo salvamento, prossegue-se chamando a página de administração com o argumento 7.

Fim

Caso tenha vindo o argumento 3, forneça um formulário para inclusão de novo usuário, com os seguintes campos: Nome completo, login e senha. Ofereça também a opção de Adicionar o usuário em questão.

Caso opte por esta opção, após o preenchimento dos dados, prossegue-se chamando a página de administração com o argumento 8.

Fim

Caso tenha vindo o argumento 4, forneça um formulário para inclusão de novo aviso, com os seguintes campos: Título e aviso. Ofereça também a opção de Salvar o aviso em questão.

Caso opte por esta opção, após o preenchimento dos dados, prossegue-se chamando a página de administração com o argumento 9.

Fim

Caso tenha vindo o argumento 5, faça uma atualização, colocando os novos dados recebidos do formulário no lugar do aviso antigo que foi editado, baseado

no seu número de identificação. Sempre nesta operação setar ele como ativo novamente.

Fim

Caso tenha vindo o argumento 6, faça uma atualização na tabela de avisos setando o aviso identificado como inativo.

Fim

Caso tenha vindo o argumento 7, faça uma atualização na tabela de usuários, colocando os valores recebidos do formulário na linha identificada pelo número de identificação do usuário que está sendo atualizado.

Fim

Caso tenha vindo o argumento 8, faça uma inserção na tabela de usuários, colocando os dados recebidos do formulário, validando o novo usuário. A senha deve ser a assinatura MD5 da cadeia de caracteres passada.

Fim

Caso tenha vindo o argumento 9, faça uma inserção na tabela de avisos, colocando os dados recebidos do formulário, validando assim o novo aviso. A data deve ser a hora atual do sistema.

Fim