

Relatorio sobre o desenvolvimento de uma aplicação WEB

Leonardo B. Gonçalves¹, Marcelo Miletto¹, Tiago F. Pinheiro¹

¹Curso de Ciência da computação - Universidade Federal do Pampa (UNIPAMPA)
CTA Alegrete - Av. Tiarajú, 810 - Ibirapuitã - Alegrete - RS - Brasil

leo-brondani@hotmail.com, marcelocm97@hotmail.com, tiagodrehmer@hotmail.com

Resumo. *Hoje em dia os websites são comuns na vida de todos. Segundo a União Internacional das Telecomunicações, órgão vinculado à Organização das Nações Unidas (ONU), número de internautas no mundo já é de 3,2 bilhões no mundo. Mas para todas esses sites, existe um mecanismo para que tudo funcione como devia. Neste artigo vamos falar um pouco sobre esses mecanismo. Vamos falar em particular do WebSite C Tutoriais, criado pelos alunos do curso de ciência da computação para a disciplina Programação para Web.*

1. Introducao

A partir de nossos conhecimentos desenvolvemos um web site básico com tutoriais na linguagem de programação "C". O conteúdo do site é para programadores iniciantes, os tutoriais são simples e diretos, e ao final de cada tutorial, criamos um pequeno teste sobre o conteúdo aprendido. Em todo o processo de desenvolvimento, usamos elementos Html, Php, JavaScript e um bando de dados sql. Nosso site além de possibilitar ao usuario navegar pelos tutoriais, permite o cadastro de conta. Tendo uma conta, entramos em uma nova característica do site, que é o forum. No forum qualquer um acessar as perguntas e respostas. Um usuario com conta cadastrada e logado no site, além de ter acesso ao forum, ele pode participar, ou seja, criar questões e responder questões de outros usuarios.

2. Desenvolvimento

Para uma melhor organização do nosso trabalho, seguimos o padrão arquitetural Model-View-Controller (MVC).

2.1. Model

A model é a responsável pela manipulação de dados, ou seja, a leitura, escrita e validação dos dados. Usamos a model em nosso projeto nas partes de conta do usuário e no forum. Na conta do usuário, a model fica responsável por salvar dados no banco, e também ler dados do usuário no banco e fazer a validação para o login. Após a validação a model retorna um aviso para a controller, confirmando ou não se está tudo correto.

2.2. Controller

A controller é responsável por receber dados do usuario e mandar para model validar ou para a view mostra na tela. É a controller que define quem vai fazer o que em dado momento. Quando um usuário deseja criar uma pergunta no forum, a controller irá pegar as requisições do usuário e mandar para a model escrever no banco, e depois mandar para view mostrar a questão no forum.

2.3. View

A view é simplesmente a camada no padrão mvc que faz a interação com o usuário, é ela que exibe os dados para o usuário. Quando um usuário realizar o login ou criar uma questão, é a view que pega esses dados e manda para a controller.

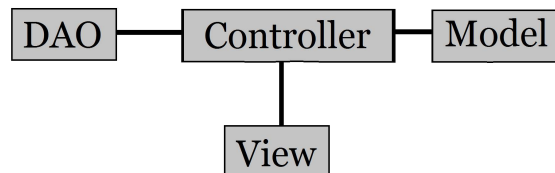


Figure 1. MVC

3. Modelo de classes

O modelo de classes do nosso projeto é composto por três classes: Validações, Sessão e Fórum.

3.1. Validações

Na Validações funciona bem como o nome já diz, serve para validar alguns casos principalmente do usuário. Essa classe possui como atributo um objeto da classe usuário, e trabalha em cima dele na parte de cadastro e edição. Quando o usuário for criar uma conta no site, é necessário que ele use um e-mail. Nesta classe verificamos se o e-mail informado pelo usuário já existe ou não no banco com o método "isEmailUnico", na primeira linha desse método criamos uma string com o seguinte comando sql "SELECT * FROM forum.usuario WHERE email = 'email informado pelo usuario' ", e mandamos para um método da classe "usuarioDAO" que retorna "true" ou "false" (verdadeiro ou falso) se o usuário já existir no banco. Outro método de validação dessa classe é o login, que consiste da mesma ideia do outro método já mencionado, o usuário entra com os dados de login (username e senha), e o método "isLoginValido", cria duas "queries" (comando sql), uma para verificar o username e outra para verificar a senha. Outros métodos dessa classe, servem para verificar se o usuário editou com sucesso alguns dados (e-mail, username ou senha).

3.2. Sessão

Na Sessão é a parte onde o usuário está conectado a partir de uma conta. Nesta parte criamos uma classe Sessão. Essa classe foi criada para manter dados do usuário. Após o login, os dados do usuário do banco, são passados para esta classe, possibilitando assim o uso do nome do usuário para alguns recursos do site, e também para edição, evitando buscar no banco os dados do usuário sempre que necessário. Assim como o login, temos o método logout, que serve basicamente para "limpar" a sessão, permitindo que outro usuário faça o login. Outro método desta classe é o refresh, que serve basicamente para atualizar os dados da sessão, quando o usuário edita o perfil.

3.3. Forum

Na parte do Forum, é onde possibilita o usuario a mandar questões para o site e salvá-las no banco, ou também possibilita o usuário a responder questões de outros usuários. Para criar uma questão ou responder, já que é preciso que o usuário esteja logado no site, criamos dois metodos para verificar isso: "buttonNovaQuestao" e "buttonNovaResposta". Os outros metodos servem para manipular as questões, como o metodo "selectQuestoes", que retorna um array contendo todas as questões no banco, que usamos para mostrar na pagina principal do forum, onde o pode navegar pelas questões. Também criamos os metodos para criar uma nova questão e também criar uma nova resposta. Esses metodos trabalham com metodos da classe QuestaoDAO como auxiliares.

4. Modelo de dados

O modelo de dados do projeto consiste em três tabelas, usuario, questao e resposta.

A tabela usuario é usada para armazenar dados do usuário, como nome e dados de login. Ela tem relação com as outras duas outras tabelas(questao e resposta). A tabela de questao e usada para armazenar perguntas do forum, criadas por usuário, nesta tabela além de armazenar dados das questões, também armazena o ID do usuário, afim de quando a view for mostrar a pergunta, também mostrar o nome do usuário que a criou. A tabela resposta armazena as respostas de determinada questão, nela é armazenado o ID da questão e também o ID do usuario que criou a resposta.

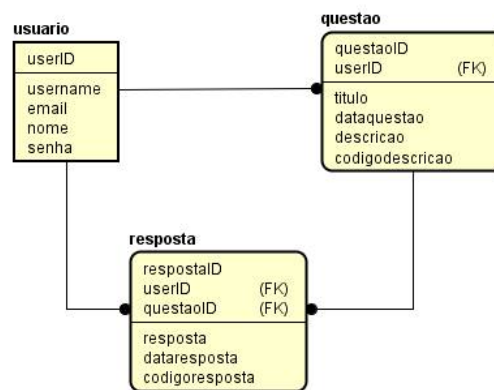


Figure 2. Modelo ER

5. Conclusao

5.1. Lições aprendidas

Aprendemos com esse projeto um pouco mais sobre organização, usando o padrao MVC, que é fundamental para um projeto grande. Aprendemos também a trabalhar com banco de dados, e a importância de modelar um banco bem estruturado. O paradigma da orientação a objetos foi uma lição importante que aprendemos, pois além de ajudar bastante com a organização, também possibilita a reutilização de código, evitando despercio de linhas e tempo.

5.2. Principais desafios

Durante o desenvolvimento da aplicação web, encontramos dificuldades principalmente na parte da utilização do padrão mvc.

6. References

Links:

<http://pt.stackoverflow.com/questions/55486/o-que->

<http://www.w3schools.com/sql/default.asp>

<http://www.w3schools.com/php/default.asp>