

# Taekwondo nos jogos olímpicos: modelo de banco de dados

Thomaz Macedo, Eduardo Furtado  
Departamento de Ciência da Computação  
Universidade de Brasília  
Distrito Federal, Brasil  
71009-000

tho.macedo@gmail.com, eduardoxfurtado@gmail.com

## Resumo

*O Brasil é a próxima sede das Olimpíadas, um enorme evento do qual pessoas mundo inteiro participam, direta ou indiretamente. Um evento desse porte gera quantidade imensa de dados. Além de dados sobre as competições, informações úteis aos espectadores e turistas podem ser integradas ao contexto, gerando informações mais completas sobre a estruturação da cidade sede do evento. Como projeto, apresentamos, com o tema do esporte olímpico Taekwondo, um banco de dados pequeno para armazenar e gerenciar esses dados. O sistema foi integrado a uma CRUD em PHP sobre servidores Apache, duas importantíssimas tecnologias relacionadas a bancos de dados web. O banco de dados escolhido foi o MySQL e ferramentas como o SmartDraw e o MySQL Workbench foram utilizadas para o desenvolver o DER e o MER. O banco de dados foi instanciado em uma máquina servidor e uma VPN foi criada para permitir o acesso remoto.*

## I. INTRODUÇÃO

Taekwondo é uma arte marcial de origem coreana praticada há mais de 50 anos e seu nome significa "o caminho dos pés e das mãos". As primeiras competições de nível mundial começaram a surgir em meados da década de 1970 mas apenas em 1988, nos jogos olímpicos de Seul, o Taekwondo foi introduzido no contexto de uma olimpíada. Participou nesse ano e em 1992 como esporte de demonstração e em 2000, em Sidney, foi oficialmente integrado como esporte olímpico. Presente desde então em todas as edições, o Taekwondo é um dos esportes que mais gera medalhas: 8 no total. São quatro modalidades de peso para cada gênero de competidores. Um evento do porte de uma olimpíada requer um alto investimento em infraestrutura por parte da cidade sede. Muitos parques esportivos devem ser reformados e adaptados, instalações devem ser construídas para receber os atletas e, além disso, toda a questão de mobilidade deve ser replanejada para atender ao grande número de turistas que vêm acompanhar o evento. Em muitos casos é também necessário sinalizar toda a cidade, preparar

equipes de orientação ao turista e distribuir cartilhas e outros materiais que possam auxiliar os não-locais em situações de emergência ou mesmo informar sobre restaurantes e hotéis mais próximos. Com o intuito de prover uma forma de disponibilizar e organizar os dados sobre as competições de Taekwondo nas olimpíadas e, ao mesmo tempo, agregar as informações úteis aos turistas, propomos nesse trabalho um banco de dados relacional que modela o contexto desse tipo de evento.

## II. METODOLOGIA

### I. DER - Diagrama Entidade Relacionamento

O passo inicial do projeto foi a elaboração de um DER onde as informações a respeito da competição possam ser vinculadas a informações úteis aos espectadores do evento e aos turistas. O local das competições é o elemento que permitiu associar todas essas informações. O desenho do projeto levou em consideração o endereço geral como composto por cidade, bairro,

endereço e descrição. O campo endereço no caso corresponde a o nome da rua, avenida ou quadra (considerando cidades como Brasília) e a descrição é que deve conter os detalhes como edifício, número, etc. Cada endereço está associado a um bairro que por sua vez se relaciona com a cidade. É sabido que os jogos olímpicos têm sede em apenas uma cidade, mas considerando que o projeto possa ser utilizado para armazenar dados de mais de uma edição do evento, a

existência de uma entidade cidade é justificada.

Pelo DER apresentado na Figura 1, percebe-se que pelo endereço da arena esportiva é possível vincular os hospitais e restaurantes mais próximos, assim como as linhas de ônibus que levam às proximidades do evento e essas informações podem ser então utilizadas pela organização da competição para orientar os turistas.

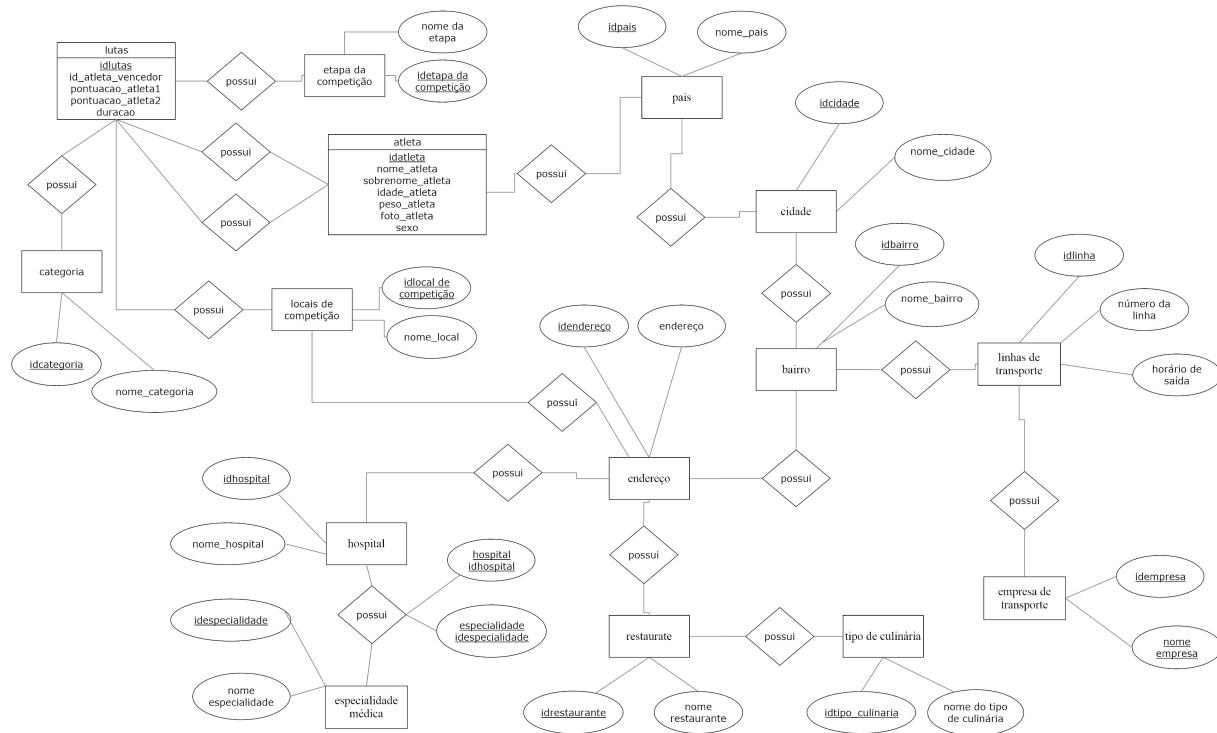


Figura 1: Diagrama Entidade Relacionamento proposto

## II. MF - Modelo Físico

O próximo passo é a escolha de um SGBD, para a implementação física do banco de dados. O grau de complexidade do projeto permite sua implementação em sistemas bem simples, como o SQLite, por exemplo. Porém, ao escolher um SGBD buscamos atender aos seguintes requisitos: open-source, gratuito, bem documentado e de ampla utilização em escala de produção. Justificadamente, foi escolhido o MySQL 5 por apresentar todas as características mencionadas anteriormente. O banco de dados escolhido foi obtido através do pacote Xampp 1.8.2, que traz Apache 2.4.7, MySQL 5.5.34 e PHP 5.4.22. Xampp tem sido utilizado em escala de produção em muitos sistemas web.

Após a instalação e configuração básica do pacote, foi necessária a instalação do MySQL Workbench 6.0 CE, o console de administração do MySQL. Trata-se de uma ferramenta bastante completa que traz todas as funcionalidades requeridas por um DBA e também possui funcionalidades de projetar um banco de dados a partir de modelos físicos. O modelo físico do banco foi gerado utilizando essa ferramenta e pode ser observado na Figura 2. É importante notar que o modelo físico traz, além dos relacionamentos descritos no DER, a referência explícita de chaves primárias e estrangeiras nas tabelas, pois o MF será utilizado para gerar o script DDL de criação do banco e a referenciação correta entre as entidades é de extrema importância para uma implementação bem sucedida.

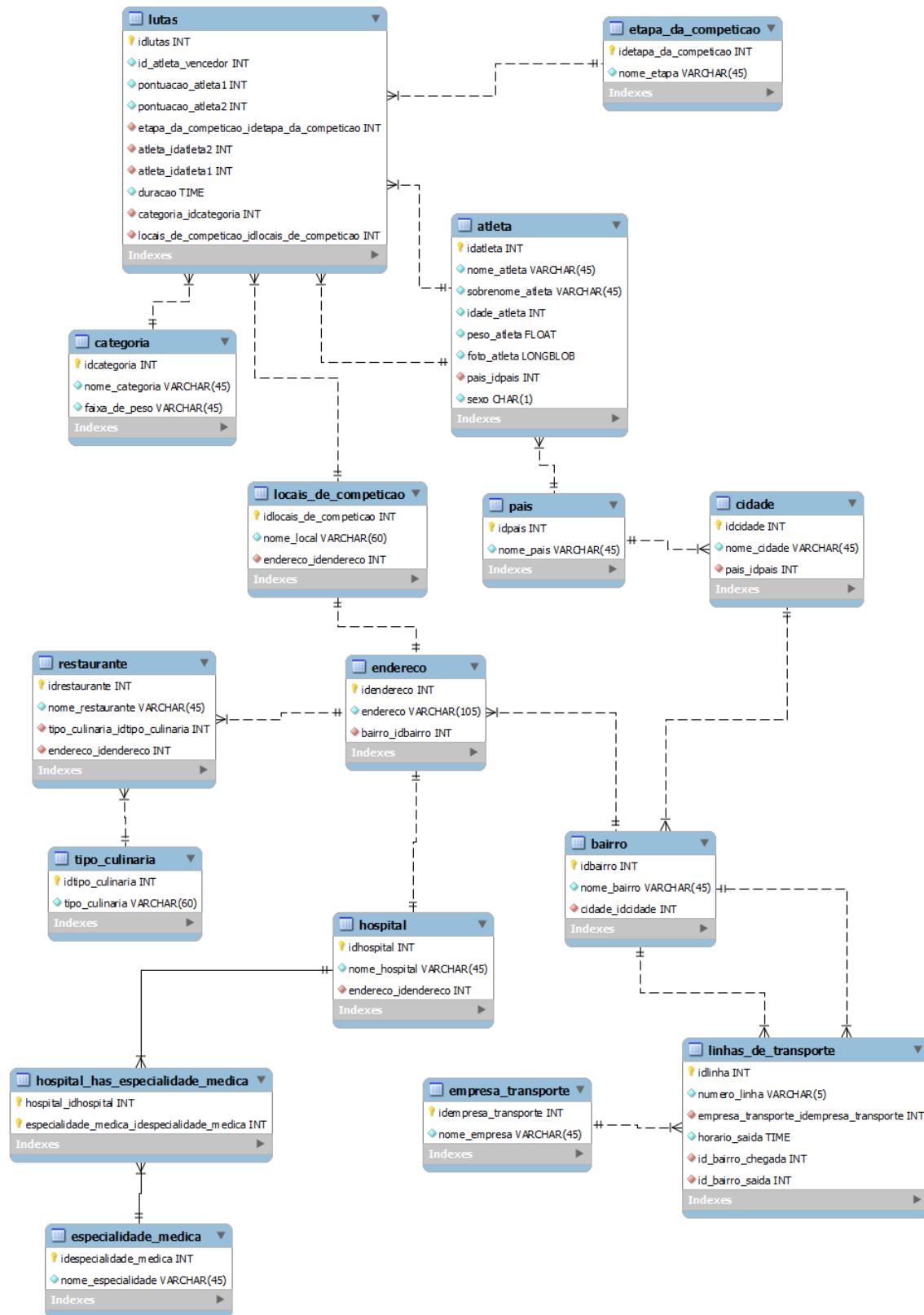


Figura 2: Modelo físico do banco de dados

### III. Normalização

Antes de executar o script de criação do banco a partir do modelo físico definido anteriormente, validou-se o modelo em relação à normalização das tabelas. A normalização de dados em um BD consiste numa série de procedimentos aplicados às tabelas que permitirão um armazenamento mais consistente e acesso mais eficiente aos dados. A normalização reduz a redundância em bancos de dados e reduz as chances de inconsistência. O trabalho original de Codd definiu 3 formas normais para bancos de dados relacionais (o conceito dessas formas normais não será abordado nesse artigo). É desejado que um banco de dados apresente tabelas na terceira forma normal, onde além de já estar na 2FN, todo atributo não chave da tabela não possuir dependência funcional transitiva em relação à chave primária. Diz-se que  $X \rightarrow Y$  é uma dependência funcional transitiva na relação R se existir um conjunto de atributos Z em R, que não sejam chave candidata de R, de tal forma que as dependências  $X \rightarrow Z$  e  $Z \rightarrow Y$  existem [1]. A tabela *lutas* foi usada para verificar a normalização do modelo proposto.

*lutas* (idlutas, id\_atleta\_vencedor, pontuacao\_atleta1, pontuacao\_atleta2, etapa\_da\_competicao

\_id\_etapa\_da\_competicao, id\_atleta2, id\_atleta1, duracao, categoria\_id\_categoria, locais\_de\_competicao\_idlocais\_de\_competicao)

A dependência funcional transitiva é dada pelos atributos *id\_atleta\_vencedor*, *etapa\_da\_competicao*, *id\_etapa\_da\_competicao*, *id\_atleta2*, *id\_atleta1*, *categoria\_id\_categoria*, *locais\_de\_competicao*, *idlocais\_de\_competicao*, que são referenciados pelas seguintes tabelas:

*atleta* (idatleta, nome\_atleta, sobrenome\_atleta, idade\_atleta, idade\_atleta, peso\_atleta, foto\_atleta, pais\_idpais, sexo)

*pais\_idpais* referencia *pais(idpais)*

*categoria* (idcategoria, nome\_categoria, faixa\_de\_peso)

*locais\_de\_competicao* (idlocais\_de\_competicao, nome\_local, endereco\_idendereco)

*endereco\_idendereco* referencia *endereco(idendereco)*

Para todas as tabelas às quais atributos de *lutas* faz referência, todos os atributos são monovalorados, dependetes funcionais totais ou são referentes a outra tabela - onde a mesma estrutura se repete - o que faz com que todo o modelo, por fim, esteja na terceira forma normal.

### IV. Álgebra Relacional

Algumas informações importantes podem ser obtidas do modelo normalizado. Dentre as consultas mais comuns estão as que permitem conhecer os vencedores e da competição. Não há uma tabela que armazene diretamente esses dados, e portanto, informações desse tipo podem ser obtidas via consultas ao banco. Para exemplificar, o campeão da categoria *Peso Mosca Masculino* a seguinte consulta deve ser realizada:

$\Pi_{\text{nome\_atleta}} (\sigma_{\text{id\_atleta\_vencedor} = \text{idatleta}} \wedge (\sigma_{\text{etapa\_da\_competicao\_idetapa\_da\_competicao} = \text{idetapada\_da\_competicao}} \wedge \text{idetapa\_da\_competicao} = \text{'Final'}}) \wedge (\sigma_{\text{categoria\_idcategoria} = \text{idcategoria}} \wedge \text{nome\_categoria} = \text{'Peso Mosca Masculino'}}) (\text{lutas X categoria})) (\text{lutas X atleta}))$

Do mesmo modo, podemos usar a mesma consulta para verificar a terceira colocada da categoria *Peso Pesado Feminino*:

$\Pi_{\text{nome\_atleta}} (\sigma_{\text{id\_atleta\_vencedor} = \text{idatleta}} \wedge (\sigma_{\text{etapa\_da\_competicao\_idetapa\_da\_competicao} = \text{idetapada\_da\_competicao}} \wedge \text{idetapa\_da\_competicao} = \text{'Semifinal'}}) \wedge (\sigma_{\text{categoria\_idcategoria} = \text{idcategoria}} \wedge \text{nome\_categoria} = \text{'Peso Pesado Feminino'}}) (\text{lutas X categoria}))$

*ria* )) (*lutas X atleta*))

A informação dos hospitais mais próximos ao local de competição pode ser obtida pela seguinte consulta:

$$\Pi_{\text{nome\_hospital}} (\sigma_{\text{bairro} = \text{id\_bairro}} (\sigma_{\text{endereco\_idendereco} = \text{idendereco}} (\text{hospital X endereco}) \wedge \text{endereco\_id\_endereco} = \text{id\_endereco} (\text{locais\_de\_competicao X endereco})) (\text{endereco X bairro}))$$

### V. CRUD

CRUD é o termo utilizado em computação para descrever as quatro operações básicas relacionadas à persistência no contexto de bancos de dados: Create, Read, Update e Delete. Na maior parte dos casos, CRUDs são implementadas através de uma aplicação que capaz de se comunicar e prover uma interface gráfica que permite ao usuário realizar suas operações sem conhecer o esquema do banco e sem escrever queries SQL. O PHP foi escolhido como linguagem para desenvolvimento da CRUD por já constar no pacote Xampp e por ser voltada para a web. É muito utilizada para construir aplicações sobre MySQL e com documentação extensa na web. Muitos códigos de exemplo podem ser facilmente encontrados e adaptados para casos de uso mais específicos. A interface

desenvolvida foi baseada numa antiga aplicação de um dos integrantes do projeto.

Como alternativa para o CRUD desenvolvido em PHP, há um software mais robusto, completo e amplamente utilizado no mercado, o *phpMyAdmin*, que é

um sistema completo para gerenciar bancos de dados, mas que também pode ser utilizado como uma CRUD, mediante a criação de um usuário com permissões restritas a apenas estas operações básicas no banco de dados, que pode acessar o sistema do *phpMyAdmin* para tal.

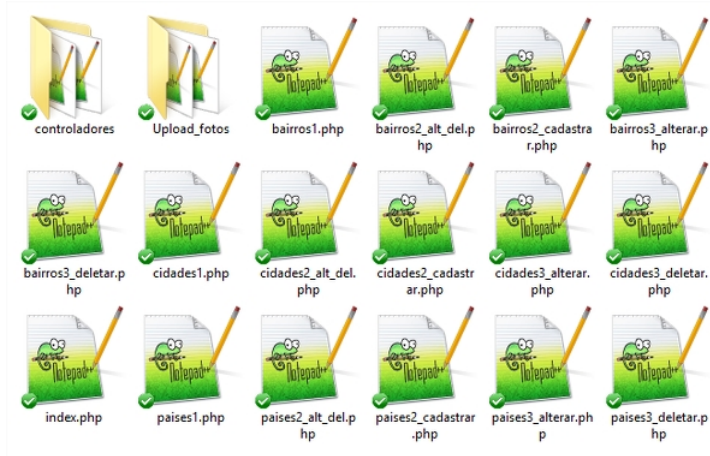


Figura 3: Estrutura do CRUD desenvolvido em PHP



Figura 4: CRUD - Tela inicial da aplicação

192.168.25.25/crud/bairros1.php

Nome:

**Create**

cidade:

- Belem, Brasil
- Belo Horizonte, Brasil
- Brasilia, Brasil
- Cuiaba, Brasil
- Fortaleza, Brasil
- Goiania, Brasil
- Manaus, Brasil
- Natal, Brasil
- Porto Alegre, Brasil
- Rio de Janeiro, Brasil

**read**

Cadastrar

bairros já cadastradas:

- Asa Norte, Brasilia, Brasil
- Asa Sul, Brasilia, Brasil
- Ceilandia, Brasilia, Brasil
- Gama, Brasilia, Brasil
- Guara, Brasilia, Brasil
- Lago Norte, Brasilia, Brasil
- Lago Sul, Brasilia, Brasil
- Noroeste, Brasilia, Brasil
- Parkway, Brasilia, Brasil
- Sobradinho, Brasilia, Brasil

**read**

Alterar/Deletar **update / delete**

Figura 5: CRUD - exemplo

192.168.25.25/crud/bairros2\_alt\_del.php

ID: 3

Bairro:

cidade:

- Belem, Brasil
- Belo Horizonte, Brasil
- Brasilia, Brasil
- Cuiaba, Brasil
- Fortaleza, Brasil
- Goiania, Brasil
- Manaus, Brasil
- Natal, Brasil
- Porto Alegre, Brasil
- Rio de Janeiro, Brasil

**read**

Modificar **update**

Deletar **delete**

Figura 6: CRUD - Outro exemplo

Nome:

Sobrenome:

Idade:

Peso(kg):

Foto:  No file chosen

Sexo (m/f):

Nacionalidade:

- Angola
- Argentina
- Brasil
- Camaroes
- Canada
- China
- Coreia do Sul
- Croacia
- Cuba
- Espanha

Figura 7: CRUD - Exemplo do cadastro de atletas com fotos

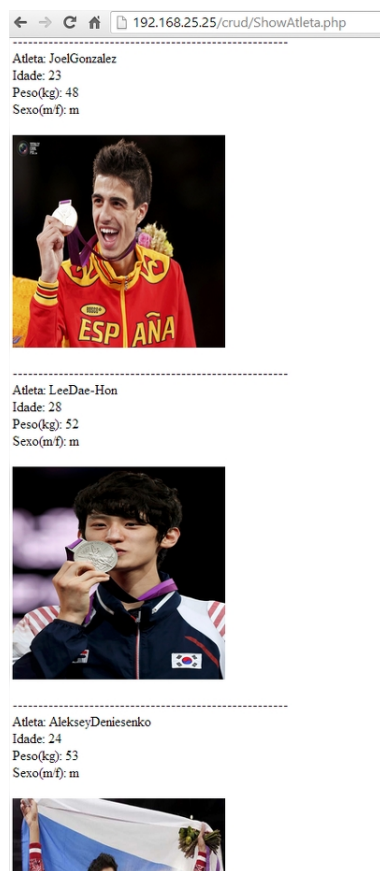


Figura 8: CRUD - Módulo de exibição das entradas cadastradas na tabela atletas

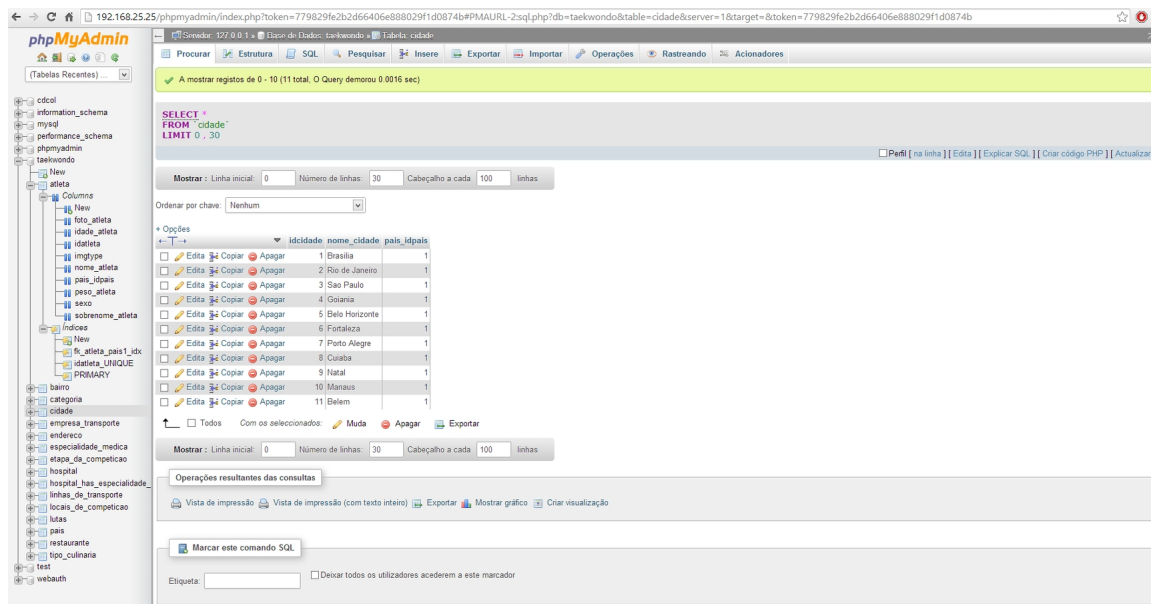


Figura 9: Ferramenta phpmyadmin

Muitos dos dados do banco foram povoados utilizando a própria aplicação CRUD desenvolvida, outros foram adicionados com queries escritas pelos integrantes e ainda outros foram adicionados utilizando a ferramenta phpmyadmin.

Os resultados são satisfatórios, e consistem de um banco funcional e populado.

### III. RESULTADOS

O banco de dados criado para o projeto é capaz de gerenciar dados referentes à competição de taekwondo, dispondo de dados sobre as lutas como pontuação de cada atleta, data de realização e local de realização e relacionar essas informações com outras úteis para turistas e espectadores através da chave identificadora *idbairro* da tabela *bairro*. A CRUD, embora não customizada, atende aos requisitos mínimos para uma aplicação que recebe essa nomenclatura e pode ser adaptada para atender às demais tabelas do modelo, não só as 3 testadas nesse projeto.

O modelo proposto é capaz de atender a várias edições do evento, já que possui uma tabela *cidade* e essa pode ser referenciada pelos locais de competição (indiretamente). O projeto não só abordou aspectos relacionados a banco de dados relacionais como também aborda o paradigma de bancos de dados web, visto que todas as ferramentas do pacote Xampp fo-

ram utilizadas e a VPN arquitetada para o projeto confere as propriedades de Web DB ao sistema, através das máquinas conectadas a essa rede. O projeto foi bem sucedido e atende a todas as requisitos da especificação.

### REFERÊNCIAS

- [1] <http://support.microsoft.com/kb/283878/pt-br> Descrição dos conceitos básicos de normalização de banco de dados  
Acessado: 10/12/2013
- [2] <http://www.youtube.com/watch?v=JhWVi-pY43Q> Tutorial on Storing and Retriving images along with text from mysql database  
Acessado: 08/12/2013
- [3] <http://www.wikinewsfacts.com/php-scripts-codes-examples/php-crud-example-1> PHP. MYSQL. CRUD example: Create Retrieve Update Delete  
Acessado: 09/12/2013
- [4] Elmasri, R.; Navathe  
*Fundamentals of Database Systems*, Sixth Edition, Pearson, 2010, pag 161-179