



UFSC

Universidade Federal de Santa Catarina

Engenharia da computação
Programa de graduação

Banco de dados de uma Rede Social

Alunos: Gabriel Bittencourt
Matheus Francisco Machado
Luan Rodrigues Silva
Professor orientador: Alexandre Leopoldo

Universidade Federal de Santa
Catarina
Engenharia da computação
Programa de graduação

Relatório

Relatório apresentado para a aprovação parcial na
disciplina de banco de dados no semestre 2016.2.

Alunos: Gabriel Bittencourt
Matheus Francisco Machado
Luan Rodrigues Silva
Professor orientador: Alexandre Leopoldo

Novembro
2016

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Resumo | 1 |
| 2 | Apresentação | 2 |
| 3 | Descrição do Sistema Social Network | 3 |
| 3.1 | Modelagem Conceitual | 3 |
| 3.2 | Modelagem Logica | 4 |
| 4 | Implementação | 4 |
| 4.1 | Script SQL | 4 |
| 4.2 | Linguagem Java e Framework Hibernate | 7 |
| 4.2.1 | Elementos | 7 |
| 4.2.2 | Classes | 7 |
| 4.2.3 | Características das classes | 7 |
| 4.2.4 | Objetos | 8 |
| 4.2.5 | Atributos | 8 |
| 4.2.6 | Métodos | 8 |
| 4.2.7 | Construtores | 8 |
| 4.3 | Hibernate | 8 |
| 5 | Análise dos Resultados | 9 |
| 5.1 | Consultas | 9 |
| 6 | Trabalhos Futuros | 15 |
| | Bibliografia | 16 |

1 Resumo

O desenvolvimento desse trabalho tem como motivação a criação de um banco de dados de uma rede social para alunos de universidade, nela eles poderão trocar conhecimento através de tópicos de perguntas com respostas sobre diversas áreas, para isso foi necessário realizar uma abstração do sistema, de um modo que fosse fácil de entender como cada área se relaciona.

2 Apresentação

O objetivo deste trabalho é a criação de um sistema utilizando banco de dados relacional integrado com uma linguagem de programação responsável pela abstração do problema, que no caso foi Java, com o framework Hibernate para realizar o mapeamento das tabelas no banco de dados, com o intuito de diminuir a complexidade do programa ao longo do desenvolvimento. Um banco de dados é

3 Descrição do Sistema Social Network

O irar efetuar o cadastramento de usuários, quais irão ter acesso a conteúdos relacionados com disciplinas e matérias, no caso as matérias são subseção das disciplina, também tendo acesso a postagem de conteúdos relacionado a uma determinada matéria, sendo esse conteúdo uma aula juntamente com anexos do tipo (PDF e JPG) ou do tipo uma duvida relacionada a tal. Todos os usuários também terão a funcionalidade de seguir disciplinas sendo isso o sistema ira recomendar em sua linha do tempo publicações de uma determinada disciplina. O sistema também fornece troca de mensagem entre usuários, visando melhorar a comunicação de respostas de sua dúvida, uma vez que a dúvida foi postada qualquer usuário pode responder a dúvida. As Matérias são relacionadas a determinada disciplina, exemplo matemática tem como matérias geometria analítica, álgebra. As publicações de são relacionadas a uma determinada matéria, sendo possível pois varias publicação de um determinado conteúdo.

3.1 Modelagem Conceitual

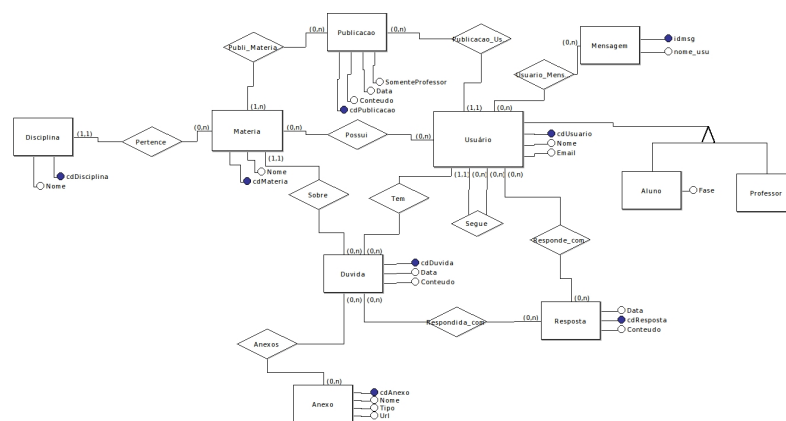


Figura 1: Imagem do modelo conceitual do banco de dados.

3.2 Modelagem Logica

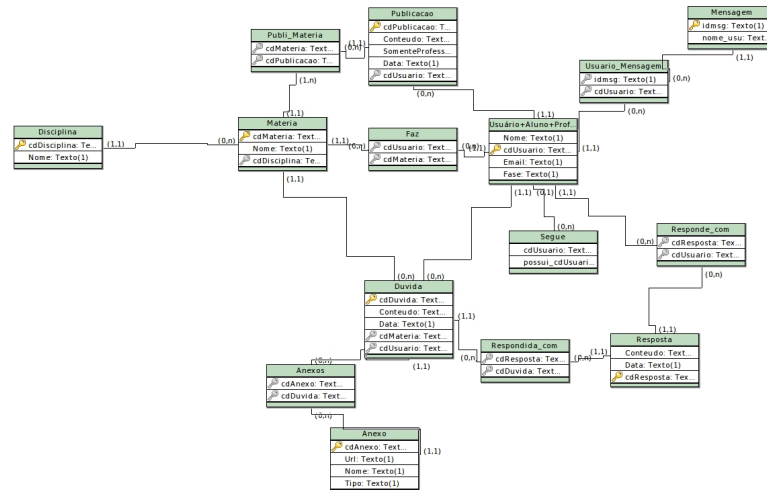


Figura 2: Imagem do modelo conceitual do banco de dados.

4 Implementação

4.1 Script SQL

```

1 CREATE TABLE Disciplina (
2   cdDisciplina int PRIMARY KEY,
3   Nome varchar(60)
4 );
5
6 CREATE TABLE Publicacao (
7   cdPublicacao int PRIMARY KEY,
8   Conteudo varchar(255),
9   Data date,
10  cdUsuario int
11 );
12
13 CREATE TABLE Materia (
14   cdMateria int PRIMARY KEY,
15   Nome varchar(50),
16   cdDisciplina int,

```

```
17 FOREIGN KEY(cdDisciplina) REFERENCES Disciplina (
    cdDisciplina)
18 );
19
20 CREATE TABLE usuario (
21 Nome varchar(50),
22 cdUsuario int PRIMARY KEY,
23 Senha varchar(40),
24 Email varchar(50),
25 Fase varchar(10)
26 );
27
28 CREATE TABLE Duvida (
29 cdDuvida int PRIMARY KEY,
30 Conteudo varchar(300),
31 Data date,
32 cdUsuario int,
33 cdMateria int,
34 FOREIGN KEY(cdUsuario) REFERENCES usuario (cdUsuario
    ),
35 FOREIGN KEY(cdMateria) REFERENCES Materia (cdMateria
    )
36 );
37
38 CREATE TABLE Anexo (
39 cdAnexo int PRIMARY KEY,
40 Url varchar(50),
41 Nome varchar(50),
42 Tipo varchar(10)
43 );
44
45 CREATE TABLE Mensagem (
46 id_msg int PRIMARY KEY,
47 nome_usu varchar(50),
48 mensagem varchar(255)
49 );
50
51 CREATE TABLE Segue (
52 cdUsuario int,
53 possui_cdUsuario int
54 );
```



```
55
56 CREATE TABLE Segue_materia (
57   cdUsuario int,
58   cdMateria int,
59   FOREIGN KEY(cdUsuario) REFERENCES usuario (cdUsuario
60   ),
61   FOREIGN KEY(cdMateria) REFERENCES Materia (cdMateria
62   )
63 );
64
65 CREATE TABLE Publi_Materia (
66   cdMateria int,
67   cdPublicacao int,
68   FOREIGN KEY(cdMateria) REFERENCES Materia (cdMateria
69   ),
70   FOREIGN KEY(cdPublicacao) REFERENCES Publicacao (
71   cdPublicacao)
72 );
73
74 CREATE TABLE Respondeu (
75   data date,
76   hora time,
77   cdUsuario int,
78   cdDuvida int,
79   FOREIGN KEY(cdUsuario) REFERENCES usuario (cdUsuario
80   ),
81   FOREIGN KEY(cdDuvida) REFERENCES Duvida (cdDuvida)
82 );
83
84 CREATE TABLE Anexos (
85   cdAnexo int,
86   cdDuvida int,
87   FOREIGN KEY(cdAnexo) REFERENCES Anexo (cdAnexo),
88   FOREIGN KEY(cdDuvida) REFERENCES Duvida (cdDuvida)
89 );
90
91 CREATE TABLE Usuario_Mensagem (
92   id_msg int,
93   cdUsuario int,
94   FOREIGN KEY(id_msg) REFERENCES Mensagem (id_msg)
```

```
90 FOREIGN KEY(cdUsuario) REFERENCES usuario (cdUsuario
91 );
92
93 ALTER TABLE Publicacao ADD FOREIGN KEY(cdUsuario)
REFERENCES usuario (cdUsuario)
```

4.2 Linguagem Java e Framework Hibernate

Nos desenvolvimentos de sistemas, existem alguns fatores importantes como: o entendimento do código, fácil manutenção, reaproveitamento entre outros. Para isso, a Programação Orientada a Objetos também conhecida como POO, tem a intenção de ajudar nesses fatores, dando tempo e agilidade no desenvolvimento de um sistema para o programador.

A Programação Orientada a Objetos foi criada por Alan Kay, autor da linguagem Smalltalk. Antes mesmo da criação Orientada a Objetos, já existiam algumas aplicações, neste caso da linguagem Simula 67, criada por Ole Johan Dahl e Kristen Nygaard em 1967.

4.2.1 Elementos

A Programação Orientada a Objetos é formada por alguns itens, dentre os quais destacamos: Classes, Objetos, Atributos, Métodos, Construtores, que irão ser mostrados nesse artigo.

4.2.2 Classes

As classes de programação são projetos de um objeto, aonde têm características e comportamentos, ou seja, permite armazenar propriedades e métodos dentro dela. Para construir uma classe é preciso utilizar o pilar da abstração. Uma classe geralmente representa um substantivo, por exemplo: uma pessoa, um lugar, algo que seja “abstrato”.

4.2.3 Características das classes

- Toda classe possui um nome
- Possuem visibilidade, exemplo: public, private, protected
- Possuem membros como: Características e Ações;
- Para criar uma classe basta declarar a visibilidade + digitar a palavra reservada class + NomeDaClasse + abrir e fechar chaves

4.2.4 Objetos

Os objetos são características definidas pelas classes. Neles é permitido instanciar objetos da classe para inicializar os atributos e invocar os métodos

4.2.5 Atributos

Os atributos são as propriedades de um objeto, também são conhecidos como variáveis ou campos. Essas propriedades definem o estado de um objeto, fazendo com que esses valores possam sofrer alterações.

4.2.6 Métodos

Os métodos são ações ou procedimentos, onde podem interagir e se comunicar com outros objetos. A execução dessas ações se dá através de mensagens, tendo como função o envio de uma solicitação ao objeto para que seja efetuada a rotina desejada.

Como boas práticas, é indicado sempre usar o nome dos métodos declarados como verbos, para que quando for efetuada alguma manutenção seja de fácil entendimento.

4.2.7 Construtores

O construtor de um objeto é um método especial, pois inicializa seus atributos toda vez que é instanciado (inicializado).

Toda vez que é digitada a palavra reservada `new`, o objeto solicita para a memória do sistema armazená-lo, onde chama o construtor da classe para inicializar o objeto. A identificação de um construtor em uma classe é sempre o mesmo nome da classe.

4.3 Hibernate

Ferramentas para auxiliar nesta tarefa tornaram-se popular entre os desenvolvedores Java e são conhecidas como ferramentas de mapeamento objeto-relacional (ORM). O Hibernate é uma ferramenta ORM open source e é a líder de mercado, sendo a inspiração para a especificação Java Persistence API (JPA). O Hibernate nasceu sem JPA mas hoje em dia é comum acessar o Hibernate pela especificação JPA. Como toda especificação, ela deve possuir implementações. Entre as implementações mais comuns, podemos citar: Hibernate da JBoss, EclipseLink da Eclipse Foundation e o OpenJPA da Apache. Apesar do Hibernate ter originado a JPA, o EclipseLink é a implementação referencial.

O Hibernate abstrai o seu código SQL, toda a camada JDBC e o SQL será gerado em tempo de execução. Mais que isso, ele vai gerar o SQL que serve para um determinado banco de dados, já que cada banco fala um "dialetto" diferente dessa linguagem. Assim há também a possibilidade de trocar de banco de dados sem ter de alterar código Java, já que isso fica de responsabilidade da ferramenta.

5 Análise dos Resultados

Neste tópico iremos listar uma serie de análise de resultado, visando manter a consistência do banco de dados, sera realizada consultas utilizando o conceito de álgebra relacional. Assim iremos extrair informações do banco de dados.

5.1 Consultas

- Primeira consulta tem o objetivo de listar os usuários cadastrados no banco de dados.

```
1  /**
2  * Lista usuarios cadastrados
3  */
4  @SuppressWarnings("unchecked")
5  public void print() {
6
7      System.out.println("\n-- Relacao de
          usuario cadastrados --");
8      List<Usuario> listUsuario = (List<
          Usuario>)modelCtr.list(Usuario.class)
9      ;
10     for (Usuario usuario : listUsuario) {
11         System.out.println("Id: "+usuario.
12             getCdusuario()+" Nome: "+usuario.
13                 getNome());
14     }
15 }
```

Codigo Sql para realizar a mesma pesquisa:

```
1  Select cdUsuario , Nome
```

```

2      from usuario
3      order by cdUsuario

```

Resultado da consulta: Alguns usuários cadastrados em nosso sistema

| Output pane | |
|-------------|---|
| Data Output | Explain Messages History |
| | |
| | cdusuario nome Integer character varying(50) |
| 1 | 0 Rúben Louzada |
| 2 | 1 Ricardina Lóio |
| 3 | 2 Clotilde Passarinho |
| 4 | 3 Camilo Mafra |
| 5 | 4 Romano Marrero |
| 6 | 5 Belmiro Hidalgo |
| 7 | 6 Alexandre Rosmaninho |
| 8 | 7 Osvaldo Bivar |
| 9 | 8 Luciana Chávez |
| 10 | 9 Veridiana Larangeira |
| 11 | 10 Umbelina Semedo |
| 12 | 11 Rúben Louzada |
| 13 | 12 Araci Neiva |
| 14 | 13 Tércio Açoies |
| 15 | 14 Hedvigies Damasceno |
| 16 | 15 Olga Mascareñas |
| 17 | 16 Eduarda Pajares |
| 18 | 17 Adérito Moura |
| 19 | 18 Guido Páez |
| 20 | 19 Edite Alencastre |
| 21 | 20 Ilídio Piñero |
| 22 | 21 Bibiana Alcántara |
| 23 | 22 Mercedes Lucena |
| 24 | 23 Alberta Aragão |
| 25 | 24 Camilo Henriques |
| 26 | 25 Ronaldo Tabajara |
| 27 | 26 Ginéculo Benevides |
| 28 | 27 Fulvio Pequeno |
| 29 | 28 Zulmira Coimbra |
| 30 | 29 Clotilde Colaço |

Figura 3: Imagem com resultado de usuários cadastrados .

- Objetivo é listar todas as Disciplinas cadastradas.

```

1  /**
2  * Lista disciplinas
3  */
4  @SuppressWarnings("unchecked")
5      public void print() {
6
7          System.out.println("\n-- Numero de
              Disciplinas --");

```

```

8      List<Disciplina> listDisciplina = (List<
        Disciplina>)modelCtr.list(Disciplina.
            class);
9      for (Disciplina disc : listDisciplina) {
10         System.out.println("Id: "+disc.
            getCddisciplina()+ " Nome: "+
            disc.getNome());
11     }
12 }

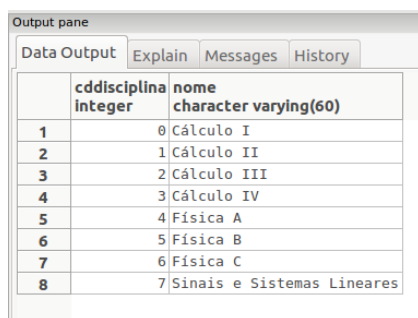
```

Código Sql para realizar a mesma pesquisa:

```

1  Select  cddisciplina, Nome
2         from Disciplina
3         order by cddisciplina

```



| | cddisciplina Integer | nome character varying(60) |
|---|-------------------------|-------------------------------|
| 1 | 0 | Cálculo I |
| 2 | 1 | Cálculo II |
| 3 | 2 | Cálculo III |
| 4 | 3 | Cálculo IV |
| 5 | 4 | Física A |
| 6 | 5 | Física B |
| 7 | 6 | Física C |
| 8 | 7 | Sinais e Sistemas Lineares |

Figura 4: Imagem com resultado de disciplinas cadastradas.

- Objetivo é listar todos os usuários e o número de dúvidas postadas por cada usuário.

```

1  /**
2  * Lista disciplinas
3  */
4  @SuppressWarnings("unchecked")
5      public void print() {
6          int cdUsuario = 0;
7          System.out.println("\n-- Relacao
            de usuario por Id:\n");
8          Usuario pac = (Usuario)modelCtr.
            getByCode(Usuario.class,
            cdUsuario);

```

```

9         while(pac != null) {
10             int duvidas = 0, codigoDuvida =
                0;
11             Duvida dv = (Duvida) modelCtr.
                getByCode(Duvida.class,
                codigoDuvida);
12             while(dv != null) {
13                 if(dv.getUsuario().
                    getCdusuario() == pac.
                    getCdusuario()) {
14                     duvidas++;
15                 }
16                 codigoDuvida++;
17                 dv = (Duvida) modelCtr.
                    getByCode(Duvida.class,
                    codigoDuvida);
18             }
19             System.out.println("Id: "+pac.
                getCdusuario()+" Nome: "+pac.
                getNome()+"      Numero de
                duvidas : "+duvidas);
20             cdUsuario++;
21             pac = (Usuario)modelCtr.
                getByCode(Usuario.class,
                cdUsuario);
22         }
23     }

```

Resultado da consulta: Alguns usuários cadastrados em nosso sistema

-- Relação de Usuários por Id:

```

Id: 0 Nome: Armanda Mexia      Numero de duvidas : 1
Id: 1 Nome: Aluísio Becerra    Numero de duvidas : 1
Id: 2 Nome: Adelaide Rodovalho Numero de duvidas : 1
Id: 3 Nome: Capitolina Mendoza Numero de duvidas : 0
Id: 4 Nome: Apuã Damásio      Numero de duvidas : 0
Id: 5 Nome: Emanuel Alcántara  Numero de duvidas : 0
Id: 6 Nome: Anabela Loio      Numero de duvidas : 0
Id: 7 Nome: Adalberto Acuña    Numero de duvidas : 0
Id: 8 Nome: Arminda Cervera    Numero de duvidas : 0
Id: 9 Nome: Heloísa Castelo Branco Numero de duvidas : 0
Id: 10 Nome: Lénia Menezes     Numero de duvidas : 0
Id: 11 Nome: Marcelo Corrêa    Numero de duvidas : 0
Id: 12 Nome: Geraldo Cabrera   Numero de duvidas : 0
Id: 13 Nome: Jónatas Travassos Numero de duvidas : 0
Id: 14 Nome: Firmino Lousada   Numero de duvidas : 0
Id: 15 Nome: Arcidres Velasco  Numero de duvidas : 0
Id: 16 Nome: Leopoldina Luz    Numero de duvidas : 0
Id: 17 Nome: Tairine Guilheiro Numero de duvidas : 0
Id: 18 Nome: Sarita Guarabira  Numero de duvidas : 0
Id: 19 Nome: Arminda Faro      Numero de duvidas : 0

```

Figura 5: Imagem com resultado dos usuários com a quantidade de dúvidas relacionadas.

Nessa consulta, fizemos um relacionamento de três tabelas, Disciplina, Materia e Duvida.

```

1 SELECT * FROM Disciplina
2 INNER JOIN Materia ON Materia.cddisciplina =
   Disciplina.cddisciplina
3 INNER JOIN Duvida ON Duvida.cdmateria = Materia.
   cdmateria;

```

| | cddisciplina integer | nome character varying(60) | cdmateria integer | nome character varying(50) | cddisciplina integer | cdduvida integer | conteudo character varying(500) | data date | cdusuario integer | cdmateria integer |
|---|-------------------------|-------------------------------|----------------------|-------------------------------|-------------------------|---------------------|---|--------------|----------------------|----------------------|
| 1 | 0 | Cálculo I | 0 | Integral | 0 | 0 | Como faz essa integral? integral(x^2 dx) avaliado de 0 ate pi | 2016-02-29 | 0 | 0 |
| 2 | 1 | Cálculo II | 1 | Derivadas | 1 | 1 | Derivando funcao seno, d/dx[sen x] | 2016-07-20 | 1 | 1 |
| 3 | 2 | Cálculo III | 2 | Convolução | 2 | 2 | Qual é a definição de convolução? Procurei em vários livros e não | 2016-10-02 | 2 | 2 |

Figura 6: Imagem com resultado dos usuários com a quantidade de dúvidas relacionadas.

Realizamos um update na tabela, colocando a fase do aluno como 9, onde o id era par.

```

1 UPDATE usuario
2 SET Fase = '9'

```


3 **WHERE** cdUsuario % 2 = 0;

| Nome | cdUsuario | Senha | Email | Fase |
|--------------------|-----------|------------|------------------------------------|------|
| Udo Fragoso | 0 | 20snh@165 | udofragoso20@gmail.com | 4 |
| Simão Sucupira | 1 | 323snh@376 | simaosucupira75@hotmail.com | 9 |
| Emiliano Vilhena | 2 | 66snh@334 | emilianovilhena85@hotmail.com.br | 9 |
| Hermenegildo Brum | 3 | 266snh@39 | hermenegildobrum89@hotmail.com | 8 |
| Quintino Higuera | 4 | 102snh@375 | quintinoiguera62@hotmail.com.br | 5 |
| Eloi Marmou | 5 | 379snh@139 | eloimarmou6@bol.com.br | 8 |
| Vânia Caminha | 6 | 250snh@195 | vaniacaminha20@hotmail.com | 5 |
| Tainá Moraes | 7 | 210snh@278 | tainamoraes24@hotmail.com | 2 |
| Léia Carreira | 8 | 190snh@225 | leiacarreira47@gmail.com | 5 |
| Emília Pessanha | 9 | 23snh@253 | emiliapessanha52@bol.com.br | 5 |
| Paraguaçu Franca | 10 | 385snh@228 | paraguacufranca56@hotmail.com.br | 4 |
| Celeste Ribeiro | 11 | 287snh@80 | celestebreiro19@hotmail.com | 9 |
| Jorgina Neves | 12 | 150snh@63 | jorginaneves42@hotmail.com | 7 |
| Isadora Godinho | 13 | 98snh@103 | isadoragodinho16@gmail.com | 3 |
| Isabel Gorjão | 14 | 25snh@331 | isabelgorjao75@hotmail.com.br | 1 |
| Sofia Ulloa | 15 | 140snh@96 | sofiaulloa89@hotmail.com | 1 |
| Guadalupe Carvalho | 16 | 275snh@190 | guadalupecarvalho39@hotmail.com.br | 4 |
| Hermesinda Torres | 17 | 319snh@215 | hermesindatorres44@hotmail.com | 4 |

Figura 7: Imagem da tabela usuário antes do update.

| Nome | cdUsuario | Senha | Email | Fase |
|--------------------|-----------|------------|------------------------------------|------|
| Udo Fragoso | 0 | 20snh@165 | udofragoso20@gmail.com | 9 |
| Simão Sucupira | 1 | 323snh@376 | simaosucupira75@hotmail.com | 9 |
| Emiliano Vilhena | 2 | 66snh@334 | emilianovilhena85@hotmail.com.br | 9 |
| Hermenegildo Brum | 3 | 266snh@39 | hermenegildobrum89@hotmail.com | 8 |
| Quintino Higuera | 4 | 102snh@375 | quintinoiguera62@hotmail.com.br | 9 |
| Eloi Marmou | 5 | 379snh@139 | eloimarmou6@bol.com.br | 8 |
| Vânia Caminha | 6 | 250snh@195 | vaniacaminha20@hotmail.com | 9 |
| Tainá Moraes | 7 | 210snh@278 | tainamoraes24@hotmail.com | 2 |
| Léia Carreira | 8 | 190snh@225 | leiacarreira47@gmail.com | 9 |
| Emília Pessanha | 9 | 23snh@253 | emiliapessanha52@bol.com.br | 5 |
| Paraguaçu Franca | 10 | 385snh@228 | paraguacufranca56@hotmail.com.br | 9 |
| Celeste Ribeiro | 11 | 287snh@80 | celestebreiro19@hotmail.com | 9 |
| Jorgina Neves | 12 | 150snh@63 | jorginaneves42@hotmail.com | 9 |
| Isadora Godinho | 13 | 98snh@103 | isadoragodinho16@gmail.com | 3 |
| Isabel Gorjão | 14 | 25snh@331 | isabelgorjao75@hotmail.com.br | 9 |
| Sofia Ulloa | 15 | 140snh@96 | sofiaulloa89@hotmail.com | 1 |
| Guadalupe Carvalho | 16 | 275snh@190 | guadalupecarvalho39@hotmail.com.br | 9 |
| Hermesinda Torres | 17 | 319snh@215 | hermesindatorres44@hotmail.com | 4 |

Figura 8: Imagem da tabela usuário após o update.

6 Trabalhos Futuros

Esse projeto de banco de dados de uma rede social para universitários é apenas o começo de um grande projeto. Há uma grande quantidade de coisas para serem implementadas, já que o que foi feito foi basicamente um camada intermediária entre o banco de dados e a aplicação e a implementação do banco de dados. Futuramente é pretendido continuar esse projeto implementando a interface do usuário final, utilizando alguma linguagem web.

Bibliografia

Heuser C. A. Projeto de banco de dados, 6^a edição Porto Alegre, Brasil, UFRGS. 2009.