

Trabalho Final PPD WEB Service - Pessoas

Filipe Soares Fernandes, RA:0022921
Marco Aurélio Monteiro Lima, RA:0022933
Pâmela Evelyn Carvalho, RA:0015669

Definindo o formulário WEB

O formulário Web criado para este trabalho possui como função consumir os dados de um WEB service, no qual será possível consultar e inserir dados de um banco de dados. O formulário aqui chamado de cadastro apresenta 5 campos em relação aos dados providos de uma pessoa. Estes campo são: o nome, a senha, o email, o login e o cpf.

Figura 1: Tela do formulário WEB



A imagem mostra uma interface web para cadastro. No topo, o título "Cadastro" está em uma barra cinza. Abaixo, há cinco campos de entrada de texto, cada um com um rótulo cinza: "Informe seu login", "Informe sua senha", "Informe seu email", "Informe seu perfil" e "Informe seu cpf". No final da lista de campos, há um botão azul com o texto "enviar" em branco.

Fonte: Autores (2019)

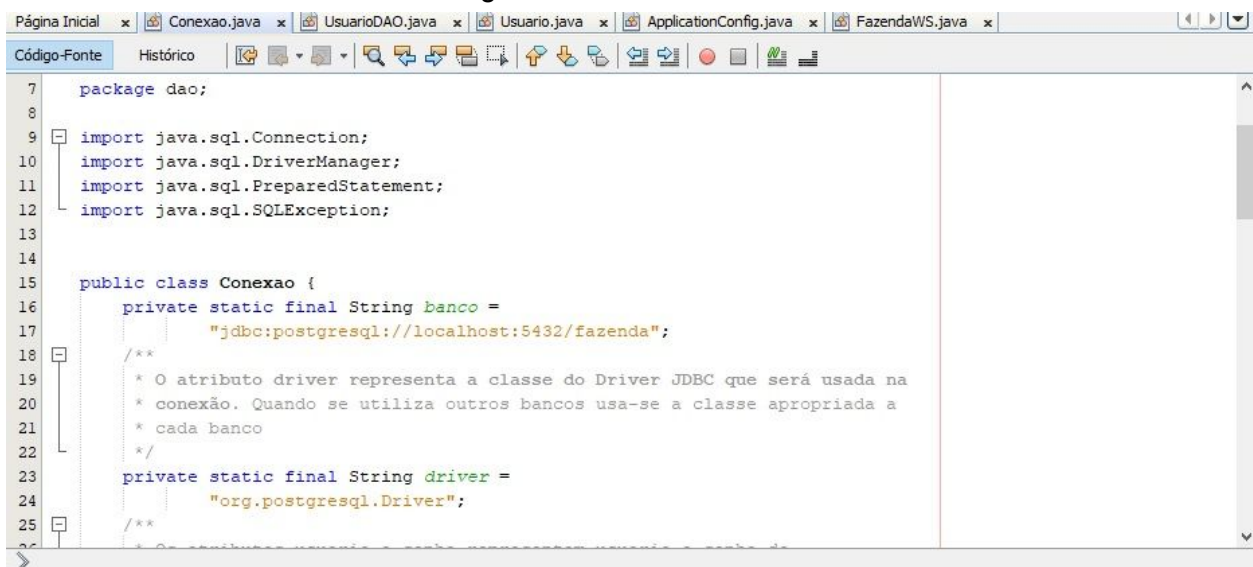
Definindo o WEBSERVICE

A linguagem utilizada para desenvolvimento dessa aplicação foi o JAVA, com utilização do NETBEANS como IDE.

Uma aplicação WEBSERVICE denominada fazenda foi criada neste trabalho, no qual foram definidos 3 pacotes. O pacote dao, possui duas classes, uma classe chamada Conexao e outra

UsuarioDAO. A classe Conexao é responsável por realizar a conexão entre a aplicação webservice com o banco de dados.

Figura 2: Classe Conexão

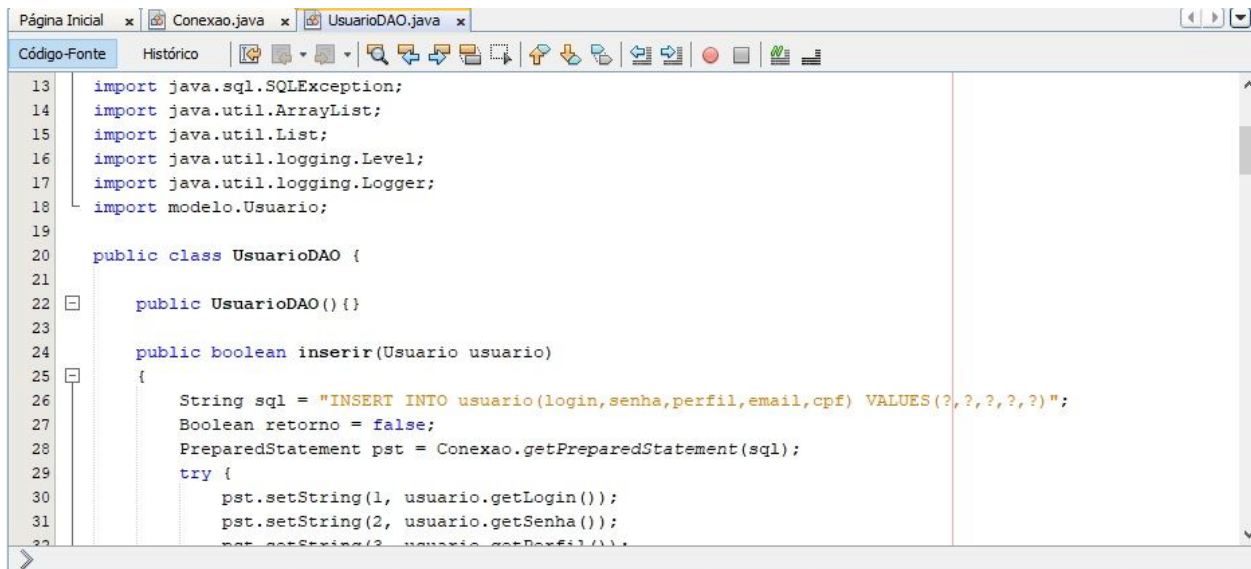


```
7 package dao;
8
9 import java.sql.Connection;
10 import java.sql.DriverManager;
11 import java.sql.PreparedStatement;
12 import java.sql.SQLException;
13
14
15 public class Conexao {
16     private static final String banco =
17         "jdbc:postgresql://localhost:5432/fazenda";
18
19     /**
20      * O atributo driver representa a classe do Driver JDBC que será usada na
21      * conexão. Quando se utiliza outros bancos usa-se a classe apropriada a
22      * cada banco
23      */
24     private static final String driver =
25         "org.postgresql.Driver";
26 }
```

Fonte: Autores (2019)

Já a classe UsuarioDao possui os métodos inserir, list e o get.

Figura 3: Classe UsuarioDAO

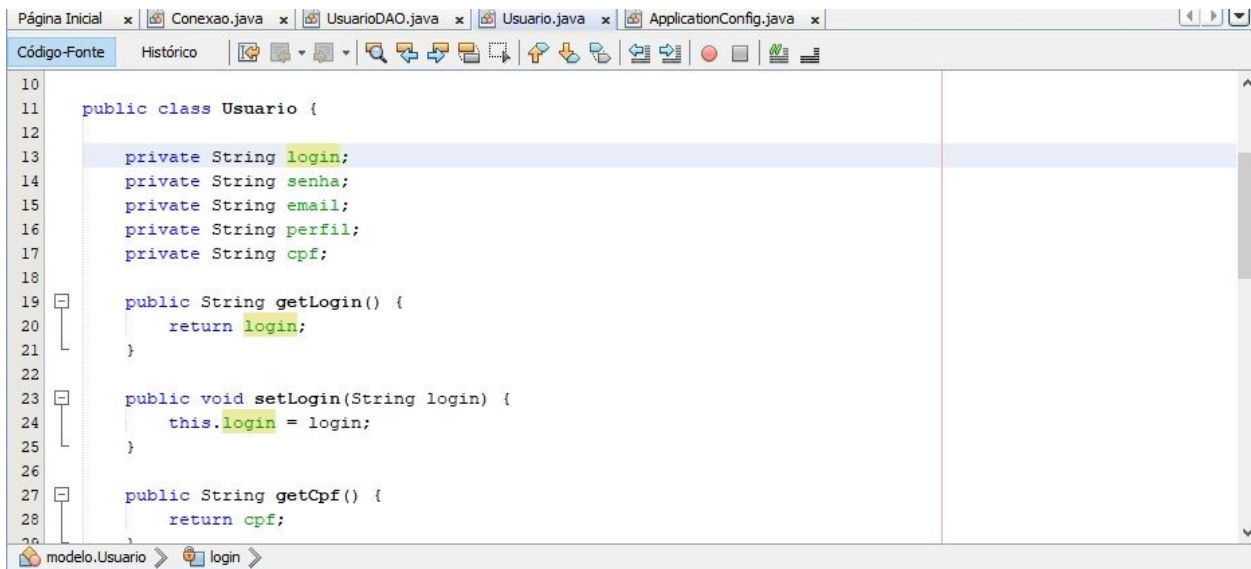


```
13 import java.sql.SQLException;
14 import java.util.ArrayList;
15 import java.util.List;
16 import java.util.logging.Level;
17 import java.util.logging.Logger;
18 import modelo.Usuario;
19
20 public class UsuarioDAO {
21
22     public UsuarioDAO() {}
23
24     public boolean inserir(Usuario usuario)
25     {
26         String sql = "INSERT INTO usuario(login,senha,perfil,email,cpf) VALUES(?, ?, ?, ?, ?)";
27         Boolean retorno = false;
28         PreparedStatement pst = Conexao.getPreparedStatement(sql);
29         try {
30             pst.setString(1, usuario.getLogin());
31             pst.setString(2, usuario.getSenha());
32             pst.setString(3, usuario.getPerfil());
```

Fonte: Autores (2019)

No pacote modelo está presente a classe Usuario, na qual estão presentes os atributos necessários para criação do formulário, além dos gets e sets, dos atributos.

Figura 4: Classe Usuario



```
10
11 public class Usuario {
12
13     private String login;
14     private String senha;
15     private String email;
16     private String perfil;
17     private String cpf;
18
19     public String getLogin() {
20         return login;
21     }
22
23     public void setLogin(String login) {
24         this.login = login;
25     }
26
27     public String getCpf() {
28         return cpf;
29     }
```

Fonte: Autores (2019)

```
public class Usuario {

    private String login;
    private String senha;
```

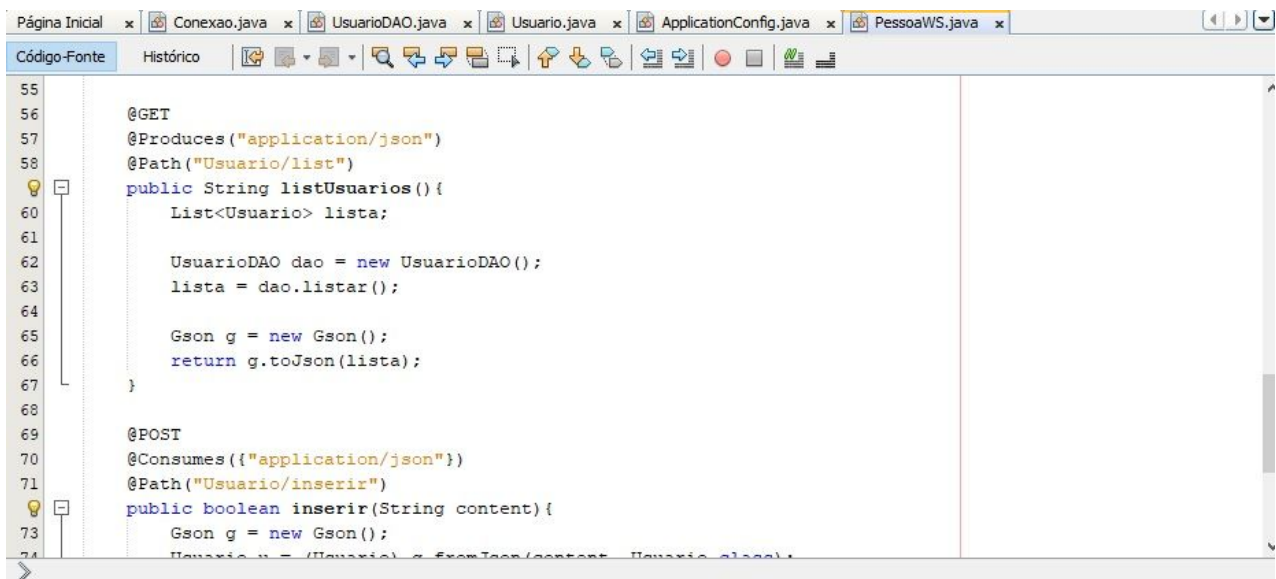
```

        private String email;
        private String perfil;
        private String cpf;
    }

```

O pacote WS é composto por duas classes: a classe ApplicationConfig e classe PessoaWS. A classe PessoaWS é onde estão os métodos get e post. Estes dois métodos foram definidos como necessário pelo professor.

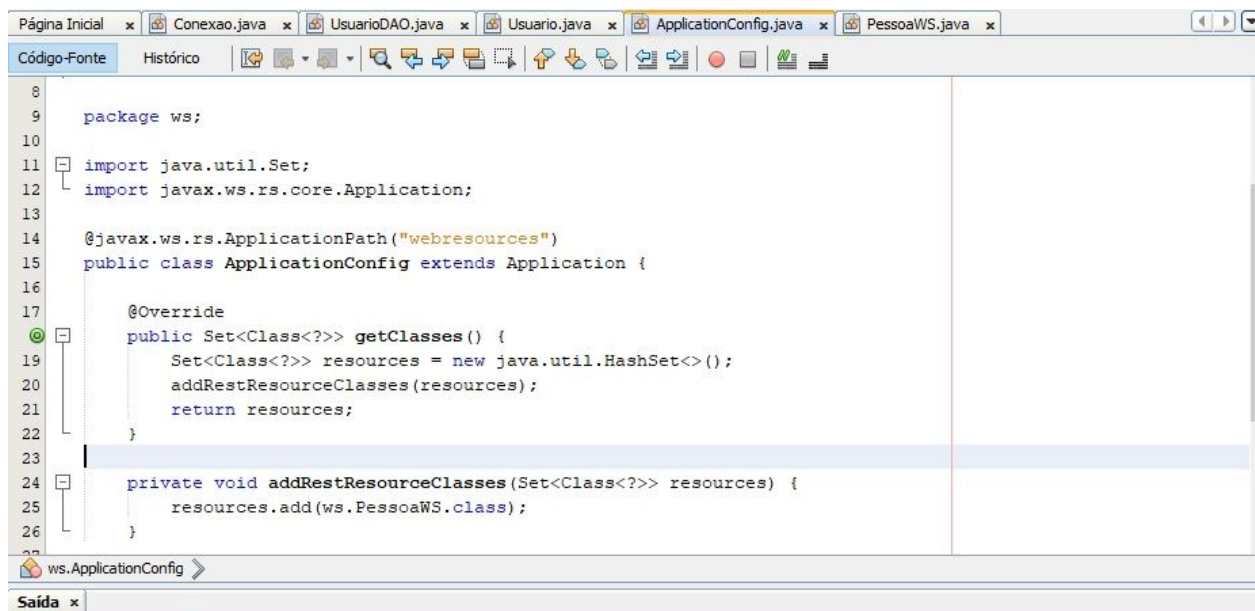
Figura 5: Classe Pessoa



Fonte: Autores (2019)

Já a classe ApplicationConfig é responsável pela especificação REST.

Figura 6: Classe ApplicationConfig;



```

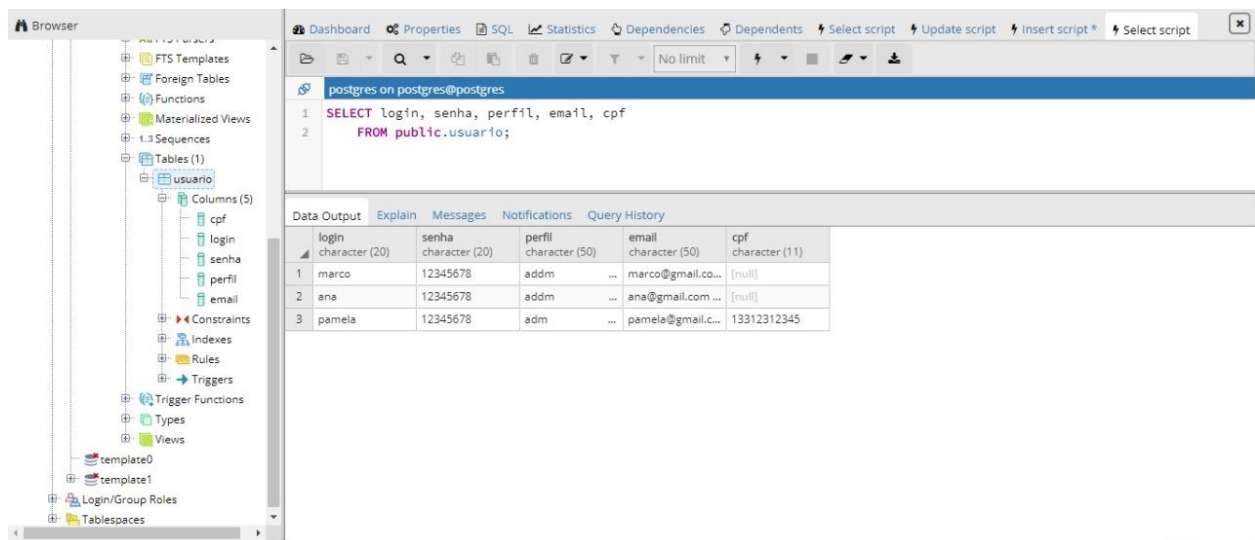
8
9 package ws;
10
11 import java.util.Set;
12 import javax.ws.rs.core.Application;
13
14 @javax.ws.rs.ApplicationPath("webresources")
15 public class ApplicationConfig extends Application {
16
17     @Override
18     public Set<Class<?>> getClasses() {
19         Set<Class<?>> resources = new java.util.HashSet<>();
20         addRestResourceClasses(resources);
21         return resources;
22     }
23
24     private void addRestResourceClasses(Set<Class<?>> resources) {
25         resources.add(ws.PessoaWS.class);
26     }
27
28 ws.ApplicationConfig >

```

Fonte: Autores (2019)

A figura 7 representa o banco de dados criado no PgAdmin3, no qual os dados inseridos são condizentes com os atributos observados na classe usuario.

Figura 7: Banco de Dados



	login	senha	perfil	email	cpf
1	marco	12345678	addm	marco@gmail.co...	[null]
2	ana	12345678	addm	ana@gmail.com ...	[null]
3	pamela	12345678	adm	pamela@gmail.c...	13312312345

Fonte: Autores (2019)