UNIVERSIDADE FEDERAL DO ABC

CONECTIVIDADE E INCLUSÃO DIGITAL PARA SÃO PAULO

FERRAMENTAS COMPUTACIONAIS

RELATÓRIO R6 – FERRAMENTAS COMPUTACIONAIS PARA AVALIAÇÃO E MONITORAMENTO DA POLÍTICA DE CONECTIVIDADE E INCLUSÃO DIGITAL

Prof. Sérgio Amadeu da Silveira, UFABC (coordenador)

Santo André, Dezembro de 2015.

UFABC – Universidade Federal do ABC
Av. dos Estados, 5001. Bairro Bangu. Santo André - SP - Brasil. CEP 09210-580

CONECTIVIDADE E INCLUSÃO DIGITAL PARA SÃO PAULO

Equipe de Projeto

Coordenador

Prof. Dr. Sérgio Amadeu da Silveira – CECS/UFABC

Pesquisadores Principais

Prof. Dr. Claudio Luis de Camargo Penteado – CECS/UFABC
Prof. Dr. Carlos Alberto Kamienski – CMCC/UFABC

Colaboradores

Waleska Barbosa da Silva
Juliano Ratusznei
Nilton Queiroz Pinheiro
Raul Iago Ataide de Souza Melo
Paulo Roberto Elias de Souza
Renata Faleiros Camargo Moreno

Sumário

1. Introdução

Este relatório visa apresentar as ferramentas desenvolvidas para dar suporte à analise dos resultados do programa WiFi Livre SP[1]. As ferramentas aqui descritas possibilitam trimestralmente a disponibilidade de dados íntegros e confiáveis para avaliação e monitoramento da política de conectividade e inclusão digital. A partir dos dados gerados automaticamente por estas ferramentas é possível obter informações que podem ser utilizadas para aprimorar os aspectos técnicos da abertura de sinal WiFi e fomentar a implementação e avaliação de uma política pública de acesso gratuíto a Internet por meio de uma rede sem fio.

Este documento contém a descrição de seis ferramentas desenvolvidas no decorrer do programa, que são: a) Coleta de Dados das Empresas e Carga no Banco de Dados, uma ferramenta responsável por obter os dados fornecidos pelas empresas prestadoras de serviço de conectividade à Internet e tornar esses dados legíveis para análise; b) Coleta de Dados do SIMET e Carga no Banco de Dados, uma ferramenta responsável por obter os dados provenientes do SIMET[2] e tornar esses dados legíveis para análise; c) Esquema de Banco de Dados, um modelo explicativo do banco de dados relacional que armazena todos os dados que serão analisados; d) Limpeza dos Dados, script responsável por realizar correções e adaptações nos dados armazenados no banco de dados, a fim de padronizar unidades de medidas, remover duplicações, corrigir dados perdidos, etc.; e) Geração de Tabelas, um programa responsável por gerar os dados contidos em tabelas de resultados disponíveis nos relatórios técnicos; f) Geração de Gráficos, um programa responsável por gerar os dados utilizados nos gráficos disponíveis nos relatórios técnicos.

A seguir cada ferramenta será apresentada detalhadamente, assim como suas configurações e adequações específicas. Todos os programas listados estão disponíveis no GitHub[3] e também em CD anexo ao relatório.

[1] http://wifilivre.org/
[2] http://simet.nic.br
[3] https://github.com/WifiLivre

2. Coleta de Dados das Empresas e Carga no Banco de Dados

Programa desenvolvido na linguagem JAVA, versão 1.7, com o objetivo de realizar a leitura dos dados fornecidos pelas empresas ZIVA[4] e WCS[5] em arquivos no formato XML e disponibilizar estes dados de acordo com o modelo do banco de dados apresentado no tópico 5 do presente documento.

A cada 10 minutos, o programa consulta um arquivo de texto dinâmico, atualizado de acordo com a página HTML fornecida pela empresa, que possui o endereço dos arquivos XML da companhia WCS, realiza o download dos arquivos XML, realiza a leitura dos dados contidos nesses arquivos e insere estes dados no banco de dados *WiFi-Livre-SP*, nas tabelas *dados_praca*, *praça_aps*, *total_ziva*, *usuarios_total_ziva* (Figura 1). A cada download, os arquivos XML são armazenados, lidos e descartados, para que não aja duplicidade de última versão dos arquivos originais.

3. Coleta de Dados do SIMET e Carga no Banco de Dados

Em todas as praças em operação foi instalado um equipamento para medir a qualidade do serviço oferecido aos usuários, chamado de SIMET (Sistema de Medição de Tráfego Internet)[6]. O SIMET executa testes de desempenho em redes com acesso à Internet e quando instalado em um Access Point ele realiza testes automaticamente que ficam disponíveis para consulta. Os dados provenientes do SIMET para as praças podem ser obtidos a partir do próprio website do programa WiFi Livre SP[7].

O presente programa é utilizado para capturar os dados contendo relatório diário das praças, disponibilizado pelo website do programa, e grava estes dades de acordo com o modelo do banco de dados apresentado no tópico 5 do presente documento.

Diariamente o programa faz o download dos dados do SIMET. Estes dados são fornecidos por meio de dois arquivos: o arquivo de log das praças em que ocorreu coleta de dados do SimetBox e o arquivo em formato .CSV, fornecido pela prefeitura, que contém as informações sobre a situação e localização das praças atendidas pelo projeto WiFi Livre SP. Nesse arquivo é possível encontrar o identificador do relatório do simetbox para cada praça. A partir destes arquivos o programa obtem os dados a serem analisados e os insere em suas respectivas tabelas no banco de dados. O programa foi desenvolvido em JAVA, versão 1.7.

[4] http://dhcp.americanet.com.br/mrtg/getstatus.php?format=xml
[5] http://187.62.212.1/prodam/xml
[6] http://simet.nic.br
[7] http://wifilivre.sp.gov.br

4. Esquema de Banco de Dados

O banco de dados utilizado para armazenar os dados de acompanhamento e análise do programa WiFi Livre SP foi modelado de modo que forneça integridade dos dados, ao mesmo tempo que rapidez nas consultas e geração de relatórios. A Figura 1 contem o modelo relacional do banco de dados.

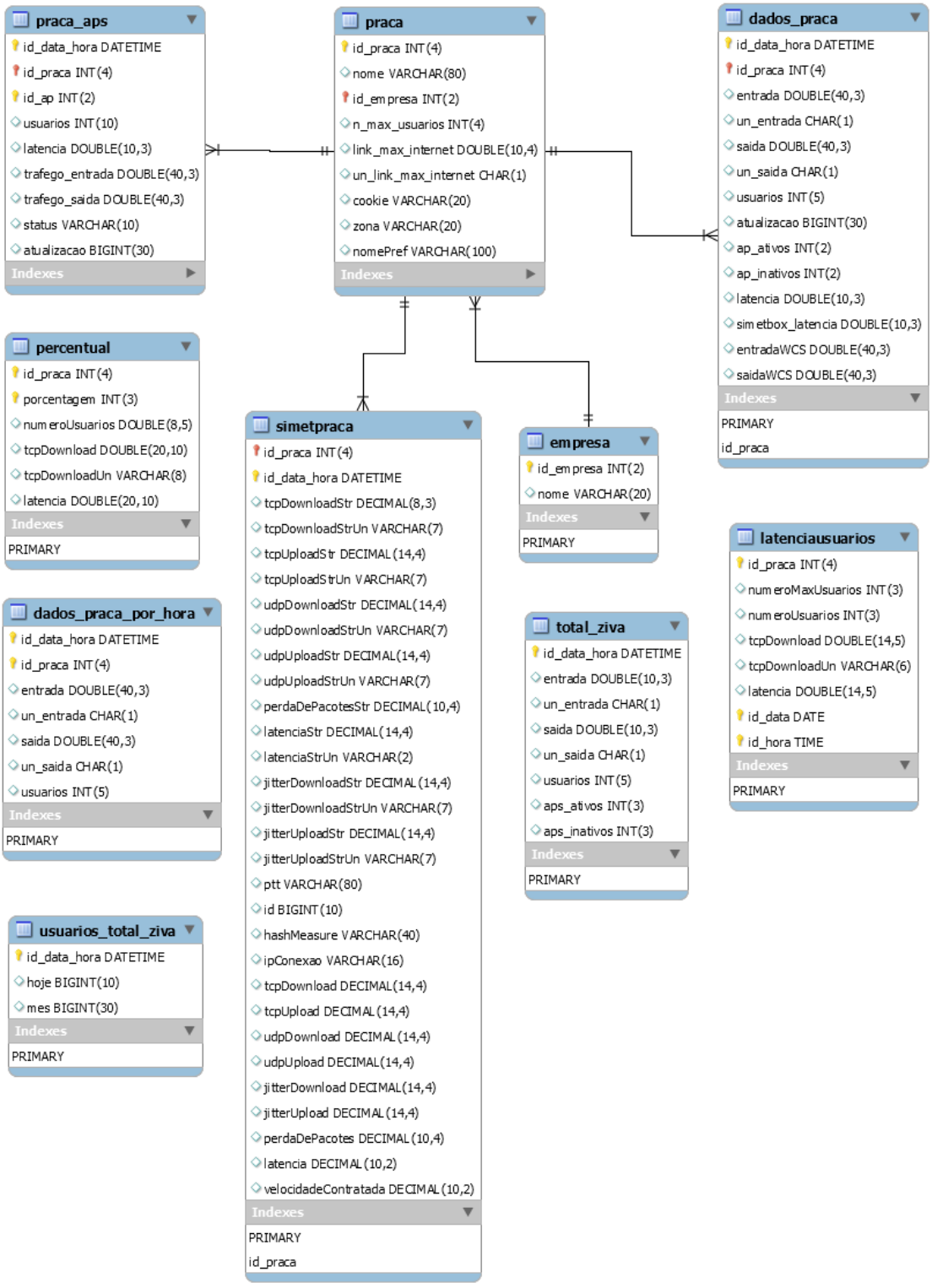

Figura 1 – Modelo Relacional do Banco de Dados

5. Limpeza dos Dados

Os dados obtidos pelas empresas e pelo SIMET necessitam de diversos ajustes e padronizações para que estejam adequados à análise, que são realizados por diversos programas, os quais usam as tabelas apresentadas na Figura 1. Os seguintes programas realizam estes ajustes:

- **AtualizandoSimetKbpsParaMbits** – Programa responsável pela atualização de todos os registros da tabela simetpraca que estão em Kbits para Mbits. O programa realiza o calculo da conversão.
- **colunaEntradaSaidaWCS** – Programa que acessa a tabela *dados_praca* e faz uma copia das colunas *entrada* e *saída,* apenas das praças que pertencem a empresa WCS. As copias são armazenada, na mesma tabela, nas colunas *entradaWCS* e *saidaWCS*. A seguir, o programa calcula quantos bits passaram pelo ponto de acesso nas praças da empresa WCS e compara esta informação com o fornecido pela empresa Ziva.
- **Transforma10minParaHora** – Programa responsável por unir os dados provenientes do SIMET com os dados fornecidos pelas empresas. Devido à diferença de granularidade dos dados foi adotada a granularidade de hora, o que gerou a necessidade de acumular os dados capturados de 10 em 10 minutos em 1 hora.
- **MineraDados** – Programa responsável por gerar os dados da tabela *latenciausuarios* contendo os campos distintos de data e hora. A tabela *latenciausuarios* é utilizada para facilitar a geração de relatórios por horas do dia.
- **MineraDados2** – Programa responsável por gerar os dados da tabela *percentual*. A tabela *percentual* é utilizada para gerar relatórios de análise do percentual de uso da carga da praça.

6. Geração de Tabelas

As tabelas apresentadas nos relatórios técnicos de análise de desempenho das praças contem dados gerados automaticamente pelos seguintes programas:

- **TabelaDesempenho** – Programa responsável por gerar as tabelas de desempenho. Busca e Automatiza os cálculos e fornece via console as informações separadas por ponto e vírgula. Estas informações são manipuladas no editor de planilhas, onde são obtidas as tabelas finais utilizadas no relatório.
- **AcessoUsuários** – Programa responsável por gerar as informações de acesso de usuários por praça e várias estatisticas referentes a quantidade de acessos. O programa obtem os dados de acesso e calcula média, desvio padrão, máximo e mínimo de acessos por praça. Todos os dados são gerados e fornecidos via console e separados por virgula. Estas informações são manipuladas no editor de planilhas, onde são obtidas as tabelas finais utilizadas no relatório.
- **Disponibilidade** – Programa responsável por gerar os percentuais de disponibilidade por praça. O programa consulta no banco de dados o quanto

cada praça esteve disponível de acordo com os dados das empresas e de acordo com os dados do SIMET, calcula o percentual respectivo de cada fonte e fornece o resultado via console e separado por vígula. Estas informações são manipuladas o editor de planilhas, onde são obtidas as tabelas finais utilizadas no relatório.

7. Geração de Gráficos

Os gráficos apresentados nos relatórios técnicos de análise de desempenho das praças contém dados gerados automaticamente por um programa desenvolvido na linguagem JAVA que consulta as informações das empresas e do SIMET no banco de dados e obtem os dados agrupados no formato adequado para a geração dos gráficos. Uma vez obtidos, os dados são manipulados no editor de planilhas para geração do gráfico. A seguir o programa de obtenção dos dados será descrito, assim como o processo de manipulação para geração dos gráficos.

O programa de obtenção dos dados contem 19 classes, que são descritas a seguir:

- A classe *Principal* é a primeira a ser executada, nela é necessário informar login, senha e local onde todos os banco de dados estão armazenados. Durante a execução é mostrado ao usuário todas as bases de dados encontradas no local fornecido, desta forma pode-se selecionar uma delas para o relatório e o local para salvar todos os dados gerados serem armazenados. Os dados informados para esta classe são essenciais para gerar os dados nas classes posteriores.
- A classe *AcessaBanco* é utilizada para realizar consultas ao banco de dados.
- A classe *ListaPraca* recebe dados como login, senha, local do banco de dados e diretório em que os dados gerados serão armazenados. Ao receber estes dados é feita uma consulta ao banco de dados para mostrar ao usuário quais praças poderão ser incluídas no relatório, ao selecionar as praças desejadas são chamadas todas as classes que geram os dados para o relatório, uma classe para cada gráfico.
- As classes que geram os gráficos recebem por parâmetro dados como login, senha, local completo do banco de dados, diretório em que se deve armazenar os dados gerados e lista com as praças que devem ser incluídas no relatório. A diferença entre as classes que geram os relatórios são os dados gerados, pois cada classe gera as informações para geração do gráfico respectivo[8].
    - fig4 - gera os dados de taxa de entrada total vs. número de usuários(serie temporal).
    - fig5 - gera os dados de taxa de entrada total vs. número de usuários(por hora).
    - fig6 - gera os dados de taxa de entrada total vs. taxa de utilização.
    - fig7 - gera os dados de histograma e distribuição acumulada da taxa de entrada total.

[8] As numerações de figuras nessa seção se referem aos gráficos do Relatório R2, que apresenta trimestralmente os resultados do monitoramento do desempenho das praças digitais.

- fig8 - gera os dados de taxa de entrada por usuário vs. número de usuários (série temporal).
- fig9 - gera os dados de taxa de entrada por usuário vs. numero de usuários (por hora).
- fig10 - gera os dados de taxa de entrada por usuário vs. taxa de utilização.
- fig11 - gera os dados de histograma e distribuição acumulada de taxa de entrada por usuário
- fig12 - gera os dados de latência vs. número de usuários(serie temporal) .
- fig13 - gera os dados de latência vs. numero de usuários (por hora).
- fig14 - gera os dados de latência vs. taxa de utilização.
- fig15 - gera os dados de histograma e distribuição acumulada da latência.
- fig16 - gera os dados de perda de pacotes (série temporal).
- fig17 - gera os dados de perda de pacotes vs. número de usuários(por hora)
- fig18 - gera os dados perda de pacotes vs. taxa de utilização
- fig19 - gera os dados de histograma e distribuição acumulada da perda de pacotes.

A partir dos dados que são gerados pelo programa descrito, os gráficos são obtidos a partir de um template de planilha para cada gráfico. Os templates geram até 15 gráficos em um mesmo arquivo (mais gráficos no mesmo arquivo pode gerar problemas de sobrecarga do editor de planilhas).

Nos gráficos por horas do dia (referentes às figuras 5, 9, 13 e 17) cada gráfico contem 24 linhas de dados, cada uma referente a uma hora do dia. A atualização do gráfico consiste em substituir os dados no template.

Nos gráficos por horas do dia da semana (referentes às figuras 4, 6, 8, 10, 12, 14, 16 e 18) é importante verificar se os dados selecionados de uma determinada praça não estão incorretos. A forma utilizada tem sido clicar sobre as linhas do gráfico e verificar a faixa dos dados, caso haja algum dado que não está sendo incluso ou que seja de outra praça fazer a correção manualmente da faixa.

Nos histogramas (referentes às figuras 7, 11, 15 e 19), apenas os valores da linha com o id da praça e a primeira linha precisam ser atualizados.

ANEXO 1 - Código do Sistema de Geração de Gráficos

a) Classe AcessaBanco

```
import java.sql.*;

public class AcessaBanco {

	private String sql;
	private Connection conexao;
	private String usuario;		//nome usuario banco
	private String senha;				//senha do usuario no
banco
	private String banco;	//Local e Servidor do banco

	public AcessaBanco(){
		this.usuario = "root";
		this.senha = "";
		this.banco = "//localhost/novonovembro";
		//Construtor de Carregar o drive MySql de Acesso ao Banco
		this.carregarDriver();
	}

	public AcessaBanco(String usuarioBanco,String senhaBanco,String
caminhoBanco){
		this.usuario = usuarioBanco;
		this.senha = senhaBanco;
		this.banco = caminhoBanco;
		//Construtor de Carregar o drive MySql de Acesso ao Banco
		this.carregarDriver();
	}

	public void setSql(String _sql){
		this.sql = _sql;
	}

	public String getSql(){
		return(sql);
	}
	public void carregarDriver(){
		try{
			Class.forName("com.mysql.jdbc.Driver").newInstance();
			//JOptionPane.showMessageDialog(null, "Driver carregado
com sucesso!");
			System.out.println("Driver carregado com sucesso!");
		}catch(Exception e){
			//JOptionPane.showMessageDialog(null,"Driver não
carregado, verifique importação do driver\n"+e);
			System.out.println("Driver não carregado, verifique
importação do driver\n"+e);
		}
	}

	public void abrirConexao(){
		try{
			conexao = DriverManager.getConnection("jdbc:mysql:"+
banco,usuario,senha);
		}
```

```
		catch(Exception e)
		{
			e.printStackTrace();
			System.out.println("Conexão não inicializada :(");
		}
	}

	public void fecharConexao(){
		try {
			conexao.close();
		} catch (SQLException e) {
			e.printStackTrace();
			System.out.println("Não foi possivel fechar conexão ...
continua aberta :(");
		}
	}

	public ResultSet consulta(){
		try {
			PreparedStatement psmt = conexao.prepareStatement(sql);
			ResultSet consulta = psmt.executeQuery();
			return consulta;
		} catch (SQLException e) {
			e.printStackTrace();
			System.out.println("Não realiza a consulta :(");
			return null;
		}
	}

	public void executa(){
		try {
			PreparedStatement psmt = conexao.prepareStatement(sql);
			psmt.execute();
		} catch (SQLException e) {
			e.printStackTrace();
			System.out.println("Erro de execução Sql");
		}
	}

}
```

b) Classe Principal

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

import javax.swing.*;


public class Principal {
	static String login="",senha="",local="";
	public static void main(String[] args) {
```

```
            login=JOptionPane.showInputDialog("Digite o login da base de
dados para se conectar");
            senha=JOptionPane.showInputDialog("Digite a senha da base de
dados para se conectar");
            local=JOptionPane.showInputDialog("Digite o caminho raiz de
todos os banco de dados exemplo : //localhost");
            String title = "WifiLivre SP";
          JFrame frame = new JFrame(title);
          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
          Container contentpane = frame.getContentPane();

          //lista combobox
          JComboBox<String> comboBox1 = new JComboBox();
          contentpane.add(comboBox1, BorderLayout.NORTH);

          //botao

          JButton btnConfirma ;
            btnConfirma = new JButton("Confirmar banco de dados
selecionado");
            contentpane.add(btnConfirma,BorderLayout.SOUTH);
          frame.add( btnConfirma );

          //evento do botao
          btnConfirma.addActionListener( new ActionListener() {
              @Override
              public void actionPerformed( ActionEvent e ) {
                   if(e.getSource() == btnConfirma){
                                String selecionado
=comboBox1.getSelectedItem().toString();
                                if(JOptionPane.showConfirmDialog(null,
"Deseja utilizar o banco "+ selecionado +
"?","Pergunta",JOptionPane.YES_NO_OPTION)==JOptionPane.YES_OPTION){
                                     local= local+"/"+selecionado ;
                                     System.out.println(local);
                                     JFileChooser fc = new JFileChooser();

                           // restringe a amostra a diretorios apenas

fc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);

                           int res = fc.showOpenDialog(null);

                           if(res == JFileChooser.APPROVE_OPTION){
                               String diretorio =
fc.getSelectedFile().getAbsolutePath();
                               //JOptionPane.showMessageDialog(null, "Voce
escolheu o diretório: " + diretorio);
                               ListaPraca Lista = new ListaPraca();
                               Lista.main(login,senha,local,diretorio);
                               frame.dispose();
                           }
                           else{
                               JOptionPane.showMessageDialog(null, "Voce nao
selecionou nenhum diretorio.");
                           }
                                }
                   }
              }
```

```
            } );

          //-----------------------------------
            AcessaBanco Conecta = new AcessaBanco(login,senha,local);
            Conecta.setSql( "show databases;");
            Conecta.abrirConexao();
            ResultSet rs3 = Conecta.consulta();
            try {
                  while (rs3.next()) {
                      comboBox1.addItem(rs3.getString(1));
                  }
            } catch (SQLException e) {
                  e.printStackTrace();
            }

          frame.setSize(300, 200);
          frame.setVisible(true);


      }

}
```

c) Classe ListaPraca

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

public class ListaPraca {
        public  void main(String login, String senha,String local,String diretorio)  {
                    List<Integer> Pracas = new ArrayList<Integer>();
                DefaultTableModel model = new DefaultTableModel(null, new String
[] {"Código", "Nome Praça","Utilizar"}) {
                                        public Class getColumnClass(int c) {
                                          switch (c) {
                                            case 2: return Boolean.class;
                                            default: return String.class;
                                          }
```

```
                                } };
                JTable table = new JTable(model);
                JScrollPane scrollPane = new JScrollPane(table);
                JFrame frame = new JFrame("Selecionar praças");
                table.setSize(500,400);
                //frame.add(table,BorderLayout.NORTH);
                frame.add(scrollPane,BorderLayout.NORTH);

                AcessaBanco Conecta = new AcessaBanco();
                    Conecta.setSql( "SELECT id_praca,nomePref FROM `praca`;");
                    Conecta.abrirConexao();
                    ResultSet rs3 = Conecta.consulta();
                    try {
                            while(rs3.next()){


                                                model.addRow(new Object []
{rs3.getString(1),rs3.getString(2),false});









                            }
                    } catch (SQLException e1) {
                            e1.printStackTrace();
                    }



                JButton btnConfirma = new JButton("Confirmar praças");
                btnConfirma.setSize(new Dimension(50, 50));
                btnConfirma.setLocation(500, 350);
                frame.add(btnConfirma,BorderLayout.SOUTH);

                btnConfirma.addActionListener( new ActionListener() {
                   @Override
                   public void actionPerformed( ActionEvent e ) {
                      if(e.getSource() == btnConfirma){
                            if(JOptionPane.showConfirmDialog(null, "Deseja utilizar
as praças
selecionadas?","Pergunta",JOptionPane.YES_NO_OPTION)==JOptionPane.YES_OPT
ION){
                                    int x=table.getRowCount();
                                    for(int k=0;k<x;k++){
                                            if((boolean)table.getValueAt(k, 2)==true){

        System.out.println(table.getModel().getValueAt(k, 0)+"
"+table.getModel().getValueAt(k, 1));
```

```
                                                    Pracas.add(Integer.parseInt((String)
table.getModel().getValueAt(k, 0)));

                                                }

                                            }
                                            if(Pracas.size()>0){
                                            fig4 class4 = new fig4();
                                class4.main(login,senha,local,diretorio,Pracas);
                                fig5 class5 = new fig5();
                                class5.main(login,senha,local,diretorio,Pracas);
                                fig6 class6 = new fig6();
                                class6.main(login,senha,local,diretorio,Pracas);
                                fig7 class7 = new fig7();
                                class7.main(login,senha,local,diretorio,Pracas);
                                fig8 class8 = new fig8();
                                class8.main(login,senha,local,diretorio,Pracas);
                                fig9 class9 = new fig9();
                                class9.main(login,senha,local,diretorio,Pracas);
                                fig10 class10 = new fig10();
                                class10.main(login,senha,local,diretorio,Pracas);
                                fig11 class11 = new fig11();
                                class11.main(login,senha,local,diretorio,Pracas);
                                fig12 class12 = new fig12();
                                class12.main(login,senha,local,diretorio,Pracas);
                                fig13 class13 = new fig13();
                                class13.main(login,senha,local,diretorio,Pracas);
                                fig14 class14 = new fig14();
                                class14.main(login,senha,local,diretorio,Pracas);
                                fig15 class15 = new fig15();
                                class15.main(login,senha,local,diretorio,Pracas);
                                fig16 class16 = new fig16();
                                class16.main(login,senha,local,diretorio,Pracas);
                                fig17 class17 = new fig17();
                                class17.main(login,senha,local,diretorio,Pracas);
                                fig18 class18 = new fig18();
                                class18.main(login,senha,local,diretorio,Pracas);
                                fig19 class19 = new fig19();
                                class19.main(login,senha,local,diretorio,Pracas);
                                            frame.dispose();
                                            }else{
                                                    JOptionPane.showMessageDialog(frame,
"Selecione alguma praça para executar análise!");
                                                    Pracas.clear();
                                            }
                                    }
                            }

                        }
                    } );
```

```
                frame.pack(); frame.validate();
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setSize(500,500);
                frame.setVisible(true);
              }



}
```

d) Classe fig4

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig4 {
	public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
		int numeros = pracas.size();
		String pesquisa="id_praca="+pracas.get(0);
		for(int k=1;k<numeros;k++){
			pesquisa=pesquisa+" or id_praca="+pracas.get(k);

		}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql( "select day(id_data_hora) as
dia,month(id_data_hora) as mes,year(id_data_hora)  as
ano,date_format(id_data_hora,'%Y-%m-%d') as DataCompleta,
id_praca,sum(usuarios) , format(sum(entrada)/1024,2,'de_DE') as entrada from
dados_praca_por_hora "+
				"where "+ pesquisa+
				" group by id_praca asc , dia asc , mes asc, ano
asc "+
				"order by id_praca asc , ano asc , mes asc, dia
asc;");
		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig4.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("dia ; mes ; ano ; data completa ; praca ;
soma usuarios ; entrada total em Mbits \n");
```

```
                    while(rs3.next()){

                          try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+"\n");
                          } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                          }




                    }
                    StrW.close();
                    System.out.println("FIM 4");
              } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              }


      }

}
```

e) Classe fig5

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig5 {
      public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
            int numeros = pracas.size();
            String pesquisa="id_praca="+pracas.get(0);
            for(int k=1;k<numeros;k++){
                  pesquisa=pesquisa+" or id_praca="+pracas.get(k);

            }
            AcessaBanco Conecta = new AcessaBanco(login,senha,local);
            Conecta.setSql( "select DATE_FORMAT(id_data_hora,'%H') as hora ,
id_praca,CEILING(avg(usuarios)) as mediaUsuarios ,
format((avg(entrada)/1024),2,'de_DE') as mediaEntrada "+
                        "from dados_praca_por_hora where "+
               "("+pesquisa+") "+
               "group by id_praca,hora;" );
```

```
		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig5.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("hora ; pracas; médiaUsuarios ;
MédiaEntradaMbits  \n");
			while(rs3.next()){

				try {

	StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+"\n");
				} catch (IOException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}




			}
			StrW.close();
			System.out.println("FIM 5");
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}


	}

}
```

f) Classe fig6

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig6 {
	public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
```

```
			int numeros = pracas.size();
			String pesquisa="dados_praca_por_hora.id_praca="+pracas.get(0);
			for(int k=1;k<numeros;k++){
				pesquisa=pesquisa+" or
dados_praca_por_hora.id_praca="+pracas.get(k);

			}
			AcessaBanco Conecta = new AcessaBanco(login,senha,local);
			Conecta.setSql( "select day(id_data_hora) as
dia,month(id_data_hora) as mes,year(id_data_hora)  as
ano,date_format(id_data_hora,'%Y-%m-%d') as
DataCompleta,dados_praca_por_hora.id_praca,format(sum(entrada)/1024,2,'de_DE'
) as entradaTotal,avg(usuarios) as usu
,format(avg(usuarios)/praca.n_max_usuarios,2,'de_DE') as PorcentagemUso from
dados_praca_por_hora   "+
					"left Join "+
				"praca on praca.id_praca= dados_praca_por_hora.id_praca "+
				"where  "+

		"DATE_FORMAT(id_data_hora,'%H')>=8 and
DATE_FORMAT(id_data_hora,'%H')<=18 and "+
						"("+pesquisa+" ) "+
	"group by id_praca asc , dia asc , mes asc, ano asc "+
	"order by id_praca asc , ano asc , mes asc, dia asc;");
			Conecta.abrirConexao();
			ResultSet rs3 = Conecta.consulta();
			try {
				BufferedWriter StrW = null;
				try {
					StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig6.csv"));
				} catch (IOException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				}
				StrW.write("dia ; mes ; ano ; data e hora ; praca ;
entradaTotal Mbits ; MédiaUsuarios  ; porcentagem de uso \n");
				while(rs3.next()){

					try {

		StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+";"+rs3.getString(8)+"\n");
					} catch (IOException e) {
						// TODO Auto-generated catch block
						e.printStackTrace();
					}




				}
				StrW.close();
				System.out.println("FIM 6");
			} catch (SQLException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			} catch (IOException e) {
```

```
                    // TODO Auto-generated catch block
                    e.printStackTrace();
             }


      }

}
```

g) Classe fig7

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig7 {
public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
       int numeros = pracas.size();
       String pesquisa="id_praca="+pracas.get(0);
       for(int k=1;k<numeros;k++){
              pesquisa=pesquisa+" or id_praca="+pracas.get(k);

       }
              AcessaBanco Conecta = new AcessaBanco(login,senha,local);
              Conecta.setSql( "select id_praca,count(case when entrada/1024 <=
0.1 then 1 else null end) as '0.1', "+
                                  "count(case when entrada/1024 > 0.1 and
entrada/1024 <= 0.3 then 1 else null end) as '0.3', "+
                     "count(case when entrada/1024 > 0.3 and entrada/1024 <=
0.6 then 1 else null end)as '0.6', "+
                     "count(case when entrada/1024 > 0.6 and entrada/1024 <= 1
then 1 else null end)as '1', "+
                     "count(case when entrada/1024 >1 and entrada/1024 <= 2
then 1 else null end)as '2', "+
                     "count(case when entrada/1024 >2 and entrada/1024 <= 3
then 1 else null end)as '3', "+
                     "count(case when entrada/1024 >3 and entrada/1024 <= 4
then 1 else null end)as '4', "+
                     "count(case when entrada/1024 >4 and entrada/1024 <= 5
then 1 else null end)as '5', "+
                     "count(case when entrada/1024 >5 and entrada/1024 <= 7.5
then 1 else null end)as '7.5', "+
                     "count(case when entrada/1024 >7.5 and entrada/1024 <= 10
then 1 else null end)as '10', "+
                     "count(case when entrada/1024 >10 and entrada/1024 <= 15
then 1 else null end)as '15', "+
                     "count(case when entrada/1024 >15 and entrada/1024 <= 30
then 1 else null end)as '30', "+
                     "count(case when entrada/1024 >30 and entrada/1024 <= 45
then 1 else null end)as '45', "+
                     "count(case when entrada/1024 >45 then 1 else null end) as
```

```
'mais' "+
                    "from dados_praca_por_hora "+
                    "where "+
                           pesquisa+
                           " group by id_praca;");
             Conecta.abrirConexao();
             ResultSet rs3 = Conecta.consulta();
             try {
                    BufferedWriter StrW = null;
                    try {
                           StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig7.csv"));
                    } catch (IOException e1) {
                           // TODO Auto-generated catch block
                           e1.printStackTrace();
                    }
                    StrW.write("praca ; 0.1 ; 0.3 ; 0.6 ; 1 ; 2 ; 3 ; 4 ; 5 ;
7.5 ; 10 ; 15 ; 30 ; 45 ; mais \n");
                    while(rs3.next()){

                           try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+";"+rs3.getString(8)+";"+rs3.getString(9)+";"+rs3.getString(10)+";"+rs
3.getString(11)+";"+rs3.getString(12)+";"+rs3.getString(13)+";"+rs3.getString
(14)+";"+rs3.getString(15)+"\n");
                           } catch (IOException e) {
                                  // TODO Auto-generated catch block
                                  e.printStackTrace();
                           }




                    }
                    StrW.close();
                    System.out.println("FIM 7");
             } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
             } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
             }


      }
}
```

h) Classe fig8

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig8 {
	public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
		int numeros = pracas.size();
		String pesquisa="id_praca="+pracas.get(0);
		for(int k=1;k<numeros;k++){
			pesquisa=pesquisa+" or id_praca="+pracas.get(k);

		}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql( "select day(id_data_hora) as
dia,month(id_data_hora) as mes,year(id_data_hora)  as
ano,date_format(id_data_hora,'%Y-%m-%d') as DataCompleta,
id_praca,sum(usuarios)
,format(((sum(entrada)/1024)/sum(usuarios)),2,'de_DE')as entradaMbits from
dados_praca_por_hora "+
				"where "+
			pesquisa+
			" group by id_praca asc , dia asc , mes asc, ano asc "+
				"order by id_praca asc , ano asc , mes asc, dia
asc;");
		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig8.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("dia ; mes ; ano ; data completa ; praca ;
soma usuarios ; entrada total em Mbits  por usuario \n");
			while(rs3.next()){

				try {

	StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+"\n");
				} catch (IOException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}




			}
			StrW.close();
			System.out.println("FIM 8");
		} catch (SQLException e) {
			// TODO Auto-generated catch block
```

```
                    e.printStackTrace();
              } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              }


       }

}
```

i) Classe fig9

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig9 {
       public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
              int numeros = pracas.size();
              String pesquisa="id_praca="+pracas.get(0);
              for(int k=1;k<numeros;k++){
                    pesquisa=pesquisa+" or id_praca="+pracas.get(k);

              }
              AcessaBanco Conecta = new AcessaBanco(login,senha,local);
              Conecta.setSql( "select DATE_FORMAT(id_data_hora,'%H') as hora ,
id_praca,CEILING(avg(usuarios)) as mediaUsuarios, format(avg(entrada),3 ,
'de_DE') as MédiaEntrada ,
format(((avg(entrada)/avg(usuarios))/1024),3,'de_DE') as mediaEntradaMbits "+
                           "from dados_praca_por_hora where "+
                  "("+pesquisa+ ") "+
                           "group by id_praca,hora;" );
              Conecta.abrirConexao();
              ResultSet rs3 = Conecta.consulta();
              try {
                    BufferedWriter StrW = null;
                    try {
                           StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig9.csv"));
                    } catch (IOException e1) {
                           // TODO Auto-generated catch block
                           e1.printStackTrace();
                    }
                    StrW.write("hora ; praca ; mediaUsuarios ; médiaEntrada
;EntradaPorUsuarioMbits de uso \n");
                    while(rs3.next()){

                           try {

       StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
```

```
";"+rs3.getString(4)+";"+rs3.getString(5)+"\n");
                        } catch (IOException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }




                  }
                  StrW.close();
                  System.out.println("FIM 9");
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }

}
```

j) Classe fig10

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig10 {
      public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
            int numeros = pracas.size();
            String pesquisa="dados_praca_por_hora.id_praca="+pracas.get(0);
            for(int k=1;k<numeros;k++){
                  pesquisa=pesquisa+" or
dados_praca_por_hora.id_praca="+pracas.get(k);

            }
            AcessaBanco Conecta = new AcessaBanco(login,senha,local);
            Conecta.setSql( "select
id_praca,format(avg(entradaPorUsuMbits),3,'de_DE'),
format(PorcentagemUso,4,'de_DE') from "+
                        "(select day(id_data_hora) as
dia,month(id_data_hora) as mes,year(id_data_hora)  as ano,id_data_hora as
DataCompleta,dados_praca_por_hora.id_praca,(sum(entrada)/avg(usuarios))/1024
as entradaPorUsuMbits,avg(usuarios) as usu
,avg(usuarios)/praca.n_max_usuarios as PorcentagemUso from
dados_praca_por_hora "+
                                    "left Join "+
```

```
                                                        "praca on praca.id_praca=
dados_praca_por_hora.id_praca "+
                                              "where  "+

            "DATE_FORMAT(id_data_hora,'%H')>=8 and
DATE_FORMAT(id_data_hora,'%H')<=18 and "+
                                                                  "("+pesquisa
+" ) "+
                                              "group by id_praca asc , dia asc ,
mes asc, ano asc ,id_data_hora "+
                                              "order by id_praca asc , ano asc ,
mes asc, dia asc ,id_data_hora asc) as pesquisa "+
                                              "group by id_praca asc,
PorcentagemUso asc "+
                                              "order by id_praca asc,
PorcentagemUso asc;");
            Conecta.abrirConexao();
            ResultSet rs3 = Conecta.consulta();
            try {
                  BufferedWriter StrW = null;
                  try {
                        StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig10.csv"));
                  } catch (IOException e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                  }
                  StrW.write("praca ; entrada por usuario Mbits;
porcentagem de uso \n");
                  while(rs3.next()){

                        try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
"\n");
                        } catch (IOException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }




                  }
                  StrW.close();
                  System.out.println("FIM 10");
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }

}
```

k) Classe fig11

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig11 {


public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
	int numeros = pracas.size();
	String pesquisa="dados_praca_por_hora.id_praca="+pracas.get(0);
	for(int k=1;k<numeros;k++){
		pesquisa=pesquisa+" or
dados_praca_por_hora.id_praca="+pracas.get(k);

	}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql( "select dados_praca_por_hora.id_praca,count(case
when (entrada/praca.n_max_usuarios)/1024 <= 0.1 then 1 else null end) as
'0,1', "+
				"count(case when
(entrada/praca.n_max_usuarios)/1024 > 0.1 and
(entrada/praca.n_max_usuarios)/1024 <= 0.2 then 1 else null end) as '0,2', "+
			"count(case when (entrada/praca.n_max_usuarios)/1024 > 0.2
and (entrada/praca.n_max_usuarios)/1024 <= 0.3 then 1 else null end)as '0,3',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 > 0.3
and (entrada/praca.n_max_usuarios)/1024 <= 0.4 then 1 else null end)as '0,4',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >0.4
and (entrada/praca.n_max_usuarios)/1024 <= 0.5 then 1 else null end)as '0,5',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >0.5
and (entrada/praca.n_max_usuarios)/1024 <= 0.6 then 1 else null end)as '0,6',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >0.6
and (entrada/praca.n_max_usuarios)/1024 <= 0.7 then 1 else null end)as '0,7',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >0.7
and (entrada/praca.n_max_usuarios)/1024 <= 0.8 then 1 else null end)as '0,8',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >0.8
and (entrada/praca.n_max_usuarios)/1024 <= 0.9 then 1 else null end)as '0,9',
"+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >0.9
and (entrada/praca.n_max_usuarios)/1024 <= 1 then 1 else null end)as '1', "+
			"count(case when (entrada/praca.n_max_usuarios)/1024 >1
then 1 else null end) as 'mais' "+
			"from dados_praca_por_hora "+
			"left Join "+
					"praca on praca.id_praca=
dados_praca_por_hora.id_praca "+
```

```
                    "where "+
                                                        pesquisa+
                                  " group by id_praca;");
              Conecta.abrirConexao();
              ResultSet rs3 = Conecta.consulta();
              try {
                    BufferedWriter StrW = null;
                    try {
                          StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig11.csv"));
                    } catch (IOException e1) {
                          // TODO Auto-generated catch block
                          e1.printStackTrace();
                    }
                    StrW.write("praca ; 0,1 ; 0,2 ; 0,3 ; 0,4 ; 0,5; 0,6 ;
0,7 ; 0,8 ; 0,9 ; 1; mais \n");
                    while(rs3.next()){

                          try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+";"+rs3.getString(8)+";"+rs3.getString(9)+";"+rs3.getString(10)+";"+rs
3.getString(11)+";"+rs3.getString(12)+"\n");
                          } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                          }



                    }
                    StrW.close();
                    System.out.println("FIM 11");
              } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              }


      }

}
```

l) Classe fig12

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
```

```
public class fig12 {

	public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
		int numeros = pracas.size();
		String pesquisa="id_praca="+pracas.get(0);
		for(int k=1;k<numeros;k++){
			pesquisa=pesquisa+" or id_praca="+pracas.get(k);

		}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql("select id_data , id_hora , id_praca ,
numeroUsuarios , replace(latencia,'.',',') valor_virgulado "+
                                                "from latenciausuarios
where "+
                                pesquisa+
                                " ORDER BY id_praca ASC , id_data ASC,id_hora
ASC;");
		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig12.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("Data ; hora; praca ; usuarios ; latencia
\n");
			while(rs3.next()){

				try {

	StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+"\n");
				} catch (IOException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}




			}
			StrW.close();
			System.out.println("FIM 12");
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}


	}
```

```
}
```

m) Classe fig13

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig13 {


	public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
		int numeros = pracas.size();
		String pesquisa="id_praca="+pracas.get(0);
		for(int k=1;k<numeros;k++){
			pesquisa=pesquisa+" or id_praca="+pracas.get(k);

		}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql("select DATE_FORMAT(id_hora,'%H') as hora ,
id_praca , format(avg(numeroUsuarios),2,'de_DE') as
mediaUsuarios,format(avg(latencia),2,'de_DE')  as mediaLatencia "+
				"from latenciausuarios where "+
			pesquisa+
			" group by id_praca,hora;");
		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig13.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("hora ; praca; mediaUsuarios ; mediaLatencia
\n");
			while(rs3.next()){

				try {

	StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+"\n");
				} catch (IOException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}
```

```
                }
                StrW.close();
                System.out.println("FIM 13");
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }


      }

}
```

n) Classe fig14

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig14 {
public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas){
      int numeros = pracas.size();
      String pesquisa="id_praca="+pracas.get(0);
      for(int k=1;k<numeros;k++){
            pesquisa=pesquisa+" or id_praca="+pracas.get(k);

      }
            AcessaBanco Conecta = new AcessaBanco(login,senha,local);
            Conecta.setSql( "select id_praca,format(mediaLatencia,2
,'de_DE') , format( PorcentagemUso,4,'de_DE') from "+
                        "(select  id_praca , numeroMaxUsuarios as
numeroMax,numeroUsuarios as usuarios,avg(latencia)  as mediaLatencia ,
(numeroUsuarios/numeroMaxUsuarios) as PorcentagemUso  "+
                                    "from latenciausuarios where  "+
                                          pesquisa +
                                " group by id_praca , numeroUsuarios
"+
                                "order by id_praca asc ,
numeroUsuarios asc) as pesquisas "+

                                "group by id_praca , PorcentagemUso
"+
                                "order by id_praca asc ,
PorcentagemUso asc;" );
            Conecta.abrirConexao();
```

```
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig14.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("praca; mediaLatencia  ; porcentagem de uso
\n");
			while(rs3.next()){

				try {

	StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
"\n");
				} catch (IOException e) {
					// TODO Auto-generated catch block
					e.printStackTrace();
				}




			}
			StrW.close();
			System.out.println("FIM 14");
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}


	}

}
```

o) Classe fig15

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig15 {

public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
	int numeros = pracas.size();
```

```
	String pesquisa="id_praca="+pracas.get(0);
	for(int k=1;k<numeros;k++){
		pesquisa=pesquisa+" or id_praca="+pracas.get(k);

	}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql(
				"select id_praca,count(case when latencia <= 1 then
1 else null end) as '1', "+
				   "count(case when latencia > 1 and latencia <= 2
then 1 else null end) as '2', "+
			"count(case when latencia > 2 and latencia <= 4 then 1 else
null end)as '4', "+
			"count(case when latencia > 4 and latencia <= 8 then 1 else
null end)as '8', "+
			"count(case when latencia >8 and latencia <= 16 then 1 else
null end)as '16', "+
			"count(case when latencia >16 and latencia <= 32 then 1 else
null end)as '32', "+
			"count(case when latencia >32 and latencia <= 64 then 1 else
null end)as '64', "+
			"count(case when latencia >64 and latencia <= 128 then 1 else
null end)as '128', "+
			"count(case when latencia >128 and latencia <= 256 then 1
else null end)as '256', "+
			"count(case when latencia >256 and latencia <= 512 then 1
else null end)as '512', "+
			"count(case when latencia >512 and latencia <= 1024 then 1
else null end)as '1024', "+
			"count(case when latencia >1024 and latencia <= 2048 then 1
else null end)as '2048', "+
			"count(case when latencia >2048 and latencia <= 4096 then 1
else null end)as '4096', "+
			"count(case when latencia >4096 then 1 else null end) as
'8192' "+
			"from latenciausuarios "+
			"where "+
					pesquisa +
				" group by id_praca;");


		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig15.csv"));
			} catch (IOException e1) {
				// TODO Auto-generated catch block
				e1.printStackTrace();
			}
			StrW.write("praca ; 1 ; 2 ; 4 ; 8 ; 16; 32 ; 64 ; 128 ;
256 ; 512; 1024;2048;4096;8192 \n");
			while(rs3.next()){

				try {

	StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
```

```
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+";"+rs3.getString(8)+";"+rs3.getString(9)+";"+rs3.getString(10)+";"+rs
3.getString(11)+";"+rs3.getString(12)+";"+rs3.getString(13)+";"+rs3.getString
(14)+";"+rs3.getString(15)+"\n");
                        } catch (IOException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }




                  }
                  StrW.close();
                  System.out.println("FIM 15");
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }
}
```

p) Classe fig16

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;




public class fig16 {

public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas){
      int numeros = pracas.size();
      String pesquisa="simet.id_praca="+pracas.get(0);
      String pesquisa2="dados.id_praca="+pracas.get(0);
      for(int k=1;k<numeros;k++){
            pesquisa=pesquisa+" or simet.id_praca="+pracas.get(k);
            pesquisa2=pesquisa2+" or dados.id_praca="+pracas.get(k);
      }
            AcessaBanco Conecta = new AcessaBanco(login,senha,local);
            Conecta.setSql( "select dados.id_praca,
date_format(id_data_hora,'%Y-%m-%d %T'), format(case when(perda/100)>1 then 1
else (perda/100) end,5 ,'de_DE') as perda from dados_praca_por_hora as dados
"+
                              "left join ( "+
```

```
                                        "select day(id_data_hora) as
dia,month(id_data_hora) as mes,year(id_data_hora)  as ano,hour(id_data_hora)
as hora,id_praca ,avg(perdaDePacotes) as perda from simetpraca as simet "+

            "where  "+

pesquisa+
                                                    " group by simet.id_praca
,hora , dia  , mes , ano  "+
                                                "order by simet.id_praca asc,
ano asc , mes asc, dia asc ,hora asc) as pesquisa "+
                                        "on day(dados.id_data_hora) = dia
and   month(dados.id_data_hora) = mes and year(dados.id_data_hora) = ano and
hour(dados.id_data_hora) = hora and dados.id_praca = pesquisa.id_praca "+
                                        "where  "+

pesquisa2+
                                        " order by "+
                                                "dados.id_praca asc ,
id_data_hora asc;" );
            Conecta.abrirConexao();
            ResultSet rs3 = Conecta.consulta();
            try {
                  BufferedWriter StrW = null;
                  try {
                        StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig16.csv"));
                  } catch (IOException e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                  }
                  StrW.write("praca; data e hora  ; pocentagem perda \n");
                  while(rs3.next()){

                        try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
"\n");
                        } catch (IOException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }




                  }
                  StrW.close();
                  System.out.println("FIM 16");
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }
```

```
}
```

q) Classe fig17

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig17 {


public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
	int numeros = pracas.size();
	String pesquisa="simet.id_praca="+pracas.get(0);
	String pesquisa2="praca.id_praca="+pracas.get(0);
	for(int k=1;k<numeros;k++){
		pesquisa=pesquisa+" or simet.id_praca="+pracas.get(k);
		pesquisa2=pesquisa2+" or praca.id_praca="+pracas.get(k);
	}
		AcessaBanco Conecta = new AcessaBanco(login,senha,local);
		Conecta.setSql( "select praca.id_praca ,
hour(praca.id_data_hora) as hora, ceil(avg(praca.usuarios)) as usuarios,perda
from dados_praca_por_hora as praca "+
				"left join "+
				"(select hour(simet.id_data_hora) as
hora,simet.id_data_hora,simet.id_praca ,format(case
when(avg(PerdaDePacotes)/100)>1 then 1 else (avg(PerdaDePacotes)/100) end,4
,'de_DE')  as perda from simetpraca as simet "+
                                                        "where  "+
                                                pesquisa +
                                    " group by simet.id_praca  ,hora  "+
                                "order by simet.id_praca asc,hora asc) as
pesquisa "+
                                "on hour(praca.id_data_hora) =
hour(pesquisa.id_data_hora) and praca.id_praca= pesquisa.id_praca "+
                                "where  "+
                                                pesquisa2 +

                            " group by "+
                                    "praca.id_praca  , hora "+
                            "order by "+
                                    "praca.id_praca asc , hora asc;");


		Conecta.abrirConexao();
		ResultSet rs3 = Conecta.consulta();
		try {
			BufferedWriter StrW = null;
			try {
				StrW = new BufferedWriter(new
```

```
FileWriter(diretorio+"\\fig17.csv"));
                    } catch (IOException e1) {
                          // TODO Auto-generated catch block
                          e1.printStackTrace();
                    }
                    StrW.write(" praca; hora; ;media usuarios  ; pocentagem
perda \n");
                    while(rs3.next()){

                          try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+"\n");
                          } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                          }




                    }
                    StrW.close();
                    System.out.println("FIM 17");
              } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
              }


      }


}
```

r) Classe fig18

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig18 {

public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas) {
      int numeros = pracas.size();
      String pesquisa="simet.id_praca="+pracas.get(0);
      String pesquisa2="dados.id_praca="+pracas.get(0);
      for(int k=1;k<numeros;k++){
```

```
			pesquisa=pesquisa+" or simet.id_praca="+pracas.get(k);
			pesquisa2=pesquisa2+" or dados.id_praca="+pracas.get(k);
		}
			AcessaBanco Conecta = new AcessaBanco(login,senha,local);
			Conecta.setSql( "select dados.id_praca, format(case
when(avg(perda)/100)>1 then 1 else (avg(perda)/100) end,5,'de_DE') as
perda,format(usuarios/n_max_usuarios,4,'de_DE') as percentual from
dados_praca_por_hora as dados "+
					"left join ( "+
							"select day(id_data_hora) as
dia,month(id_data_hora) as mes,year(id_data_hora)  as ano,hour(id_data_hora)
as hora,id_praca ,avg(perdaDePacotes) as perda from simetpraca as simet "+

		"where  "+
								pesquisa+
							" group by simet.id_praca  ,hora
, dia  , mes , ano  "+
						"order by simet.id_praca asc, ano asc
, mes asc, dia asc ,hora asc) as pesquisa "+
							"on day(dados.id_data_hora) =
dia  and  month(dados.id_data_hora) = mes and year(dados.id_data_hora) = ano
and hour(dados.id_data_hora) = hora and dados.id_praca = pesquisa.id_praca "+
					"left Join "+
							"praca on praca.id_praca=
dados.id_praca "+
					"where "+
								pesquisa2+
					" group by id_praca, percentual "+
					"order by "+
							"dados.id_praca asc , percentual
asc , perda asc;");


			Conecta.abrirConexao();
			ResultSet rs3 = Conecta.consulta();
			try {
				BufferedWriter StrW = null;
				try {
					StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig18.csv"));
				} catch (IOException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				}
				StrW.write("praca; pocentagem perda ; porcentagem de uso
\n");
				while(rs3.next()){

					try {

		StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
"\n");
					} catch (IOException e) {
						// TODO Auto-generated catch block
						e.printStackTrace();
					}
```

```
                }
                StrW.close();
                System.out.println("FIM 18");
            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }


      }


}
```

s) Classe fig19

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;


public class fig19 {


public  void main(String login, String senha,String local,String
diretorio,List<Integer> pracas){
      int numeros = pracas.size();
      String pesquisa="id_praca="+pracas.get(0);
      for(int k=1;k<numeros;k++){
            pesquisa=pesquisa+" or id_praca="+pracas.get(k);

      }
            AcessaBanco Conecta = new AcessaBanco(login,senha,local);
            Conecta.setSql(
                        "select id_praca,count(case when perdaDePacotes/100
= 0 then 1 else null end) as '0', "+
                        "count(case when perdaDePacotes/100 > 0 and
perdaDePacotes/100 <= 0.01 then 1 else null end) as '1', "+
                "count(case when perdaDePacotes/100 > 0.01 and
perdaDePacotes/100 <= 0.02 then 1 else null end)as '2', "+
                "count(case when perdaDePacotes/100 > 0.02 and
perdaDePacotes/100 <= 0.04 then 1 else null end)as '4', "+
                "count(case when perdaDePacotes/100 > 0.04 and
perdaDePacotes/100 <= 0.08 then 1 else null end)as '8', "+
                "count(case when perdaDePacotes/100 > 0.08 and
perdaDePacotes/100 <= 0.16 then 1 else null end)as '16', "+
                "count(case when perdaDePacotes/100 > 0.16 and
perdaDePacotes/100 <= 0.32 then 1 else null end)as '32', "+
                "count(case when perdaDePacotes/100 > 0.32 and
```

```
perdaDePacotes/100 <= 0.64 then 1 else null end)as '64', "+
                "count(case when perdaDePacotes/100 >0.64 then 1 else null
end) as 'mais' "+
                "from simetpraca "+
                "where "+
                                                pesquisa+
                                  " group by id_praca;");
            Conecta.abrirConexao();
            ResultSet rs3 = Conecta.consulta();
            try {
                  BufferedWriter StrW = null;
                  try {
                        StrW = new BufferedWriter(new
FileWriter(diretorio+"\\fig19.csv"));
                  } catch (IOException e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                  }
                  StrW.write("praca ; 0 ; 1 ; 2 ; 4 ; 8; 16 ; 32 ; 64 ;
mais \n");
                  while(rs3.next()){

                        try {

      StrW.write(rs3.getString(1)+";"+rs3.getString(2)+";"+rs3.getString(3)+
";"+rs3.getString(4)+";"+rs3.getString(5)+";"+rs3.getString(6)+";"+rs3.getStr
ing(7)+";"+rs3.getString(8)+";"+rs3.getString(9)+";"+rs3.getString(10)+"\n");
                        } catch (IOException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }




                  }
                  StrW.close();
                  System.out.println("FIM 19");
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }


      }
}
```