

UNIVERSIDADE LUSÓFONA DE HUMANIDADE E TECNOLOGIA

**Título:**

Plataforma Tempo Real de Gestão de Tráfego

**Local de Trabalho:**

InnoWave Technologies

Rua Castilho, 5<sup>a</sup> e B, sala 19

1250-066 Lisboa

**Aluno:**

Flávio Alfredo Dafá

**Orientador Universidade Lusófona:**

Professor Rui Ribeiro

**Orientador Empresa:**

Eng.<sup>º</sup> André Garrido

**Objectivos:**

- Recepção em tempo real de dados de tráfego (Sensores de contagem, painéis de mensagens, sistema SOS).
- Disponibilidade de informação de forma gráfica
- Verificação das diferentes ocorrências de trânsito.

Estágio curricular do 3º ano da Licenciatura em Engenharia Informática.

## Índice

Agradecimentos .....	4
Abstract .....	5
Introdução .....	6
2. Objectivos do projecto .....	7
2.1. Contexto de trabalho .....	7
3. InnoWave Silego.....	8
3.1 Aspectos inovadores: .....	8
3.2 Domínio de aplicação:.....	9
3.2.1 Sistemas de Transporte Inteligente: .....	9
3.2.2 Telecomunicações .....	9
4.Arquitectura .....	10
4.1 Visão Global.....	10
4.2 A tecnologia XML .....	10
4.3 A tecnologia JAVA .....	11
4.3.1 A classe XML_READER.....	12
4.3.2 A classe Ind_Filt_Op.....	12
4.3.3 A classe Lembrete .....	12
4.4 O Oracle .....	12
4.4.1 Acidentes por Estradas e por Horas .....	13
4.4.2 Acidentes por Estradas e por Mês .....	13
4.4.3 Acidentes por minutos, hora, dia, dias de semana e mês .....	13
4.4.4 Acidentes por dias do mês.....	13
4.4.5 Acidentes por dias de semana .....	14
5. Desenho de interacção do sistema .....	14
5.1 Visão Geral .....	14
5.2 Trocas de Informações .....	14
5.3. Representação Gráfica .....	15
6. Conclusão .....	17

7. Bibliografia .....	18
8. Anexo.....	19

## Índice de figuras

1. Figura 1 – Visão Geral .....	10
2. Figura 2 – Definição do documento XML.....	11
3. Figura 3 – Trocas de Informações.....	14
4. Figura 4 – Representação Gráfica das <i>baselines</i> .....	16

## Agradecimentos

Em primeiro lugar, quero dirigir os meus agradecimentos ao meu orientador, o Professor Rui Ribeiro, pela sua disponibilidade e pelo acompanhamento exercido durante os últimos meses, e ao meu orientador da InnoWave Technologies, o Eng. André Garrido pela oportunidade de ter efectuado o meu estágio curricular numa empresa em forte expansão como é o caso da INNOWAVE.

Quero também deixar uma palavra de apreço aos engenheiros Pedro Pinto, Tiago Mendes Gonçalves por prestarem toda a orientação e ensinamentos necessários para o meu bom desempenho durante esta experiência.

Agradeço, em geral, a todos os funcionários que tão bem me receberam e Acompanharam ao longo do estágio, tornando a minha integração neste departamento fácil e rápida.

A minha família e pessoas mais próximas, um abraço pelo apoio que me foi dado.

## Abstract

This project has the objective to implement a platform for receiving real-time traffic data from multiple systems in the existing road infrastructure, including sensors for counting, message boards, SOS system, messaging utility road, among other. This information should be available to operators on a graphical and intuitive.

With this project we show how the procedures are generated for the creation of baselines. Showing the technologies involved their settings and the interaction between them. These technologies are XML, JAVA and ORACLE.

The XML is responsible for the settings of the indicators, the filters and the options. The indicators give the values that help in the sensor definition, the filters help in choosing values to show and options are responsible for the choice of procedures to execute.

The Java technology is responsible for reading and storing the XML document. Connects to the databases and schedule the procedures calls.

Oracle is the database where the query is executed through the procedures. The results returned are used in the creation of baselines.

The Silego shows how the plotted values received from a query result indicator and filter provided by XML to java.

## Introdução

Foi muito gratificante para mim poder realizar o meu estágio curricular na InnoWave Technologies, onde tive a possibilidade de poder assimilar e aplicar grande parte dos conhecimentos adquiridos na faculdade.

Este estágio permitiu-me alargar os meus horizontes e testar as minhas limitações e dificuldades em aplicar a teoria adquirida aos casos reais que iam surgindo. Graças aos ensinamentos e experiências que me foram transmitidas pelos vários profissionais que comigo contactaram, foi-me possível aperceber de muitos pormenores importantes quer na análise de projectos, quer, simplesmente, na sua elaboração.

Devo dizer que, para mim, este estágio foi mais do que um simples estágio curricular. Fiquei muito satisfeito e entusiasmado pelo facto de ter realizado projectos que serão implementados na realidade.

O objectivo primordial deste estágio foi o de adquirir experiência profissional na área de TI, nas suas diversas vertentes. Esse objectivo, foi atingido através do exercício de actividades semelhantes às desempenhadas pelos profissionais que gentilmente disponibilizaram a sua ajuda e saber na orientação do meu estágio. Por outro lado, é inquestionável que o contacto frequente com os responsáveis de estágio no seu local de trabalho, me trouxe a percepção daquilo que é, de facto, o dia-a-dia de trabalho de um consultor.

## 2. Objectivos do projecto

Este projecto tem como o objectivo a construção de uma plataforma de recepção em tempo real de dados de tráfego de diversos sistemas existentes nas infra-estruturas rodoviárias, nomeadamente sensores de contagem, painéis de mensagem, sistemas SOS, mensagens de concessionárias de estradas entre outras. Esta informação deverá estar disponível de forma gráfica e intuitiva em vários ecrãs do Centro de Controlo da EP para que os operadores possam mais facilmente, para além da informação das câmaras que dispõem, verificar as diferentes ocorrências de trânsito.

### 2.1. Contexto de trabalho

O contexto de trabalho é um processo fundamental para a consolidação da formação e a interligação com mercado de trabalho.

Esse objectivo foi claramente atingido, visto que participei na implementação da plataforma de recepção em tempo real de dados de tráfego de diversos sistemas existentes nas infra-estruturas rodoviárias. A minha participação neste projecto foi a implementação de uma aplicação que auxilia a construção de *baseline*. A aplicação desenvolvida vai buscar dados à base de dados e fornece-os a aplicação responsável pela construção do gráfico.

E essa interacção fez-me ter a percepção daquilo que é, de facto, o dia-a-dia de trabalho de um consultor.

### 3. Innowave Silego

Solução de *Business Intelligence* (BI) em tempo real.

O SILEGO representa uma mudança fundamental na maneira como as empresas podem ver, monitorizar e actuar sobre eventos de negócio ou eventos técnicos, como são por exemplo os processos, mensagens, alterações de dados ou transacções.

O SILEGO tem como objectivo reduzir a latência na tomada de decisões operacionais. Este objectivo é atingido em três passos principais:

- Recolha de dados de qualquer tipo de fonte (base de dados, RSS, email, ficheiros de logs, sensores...);
- Correlação de dados e cálculo de indicadores em tempo real;
- Accionamento inteligente de acções automáticas que permitem actuar sobre os indicadores calculados.

#### 3.1 Aspectos inovadores:

Os aspectos inovadores são:

- Vasta abrangência de colectores de dados;
- Motor de correlação com processamento ultra rápido, e altamente configurável, de acordo com gramática específica;
- Gestão dinâmica de alarmes e alertas;
- Configuração pelo utilizador final de *workflows* para tratamento de alarmes e alertas;
- Utilização de inteligência artificial em tempo real para agregação e correlação de dados;
- *Dashboards* de apresentação multi-plataforma;

- Plataforma agnóstica ao tipo de negócio, ou tipo de indicador que processa.

### **3.2 Domínio de aplicação:**

O domínio de aplicação divide-se em sistemas de transporte inteligente e telecomunicações.

#### **3.2.1 Sistemas de Transporte Inteligente:**

- Monitorização em tempo real de qualidade de serviço nas estradas portuguesas
- Avaliação em tempo real do nível de perigosidade das estradas portuguesas
- Cálculo em tempo real do tempo necessário para completar trajectos
- Integração com outros sistemas de informação (aplicações mobile)
- Integração com Painéis de Mensagens Variável para recolher e colocar informação com valor para o utilizador final
- Fornecedor de informação para partilha com o utilizador final via aplicações de televisão interactiva

#### **3.2.2 Telecomunicações**

- Monitorização em tempo real de qualidade de serviço da execução de processos
- Cálculo em tempo real de indicadores de negócio
- Definição de alertas que são disparados por *workflows* de resposta sempre que um determinado *threshold* é ultrapassado para os indicadores de negócio calculados.

O SILEGO reduz a latência na tomada de acções, permitindo decisões de negócio mais rápidas e eficazes. Permite às empresas tornarem-se mais ágeis, respondendo de maneira mais rápida às necessidades dos clientes, fornecedores e parceiros.

## 4. Arquitectura

### 4.1 Visão Global

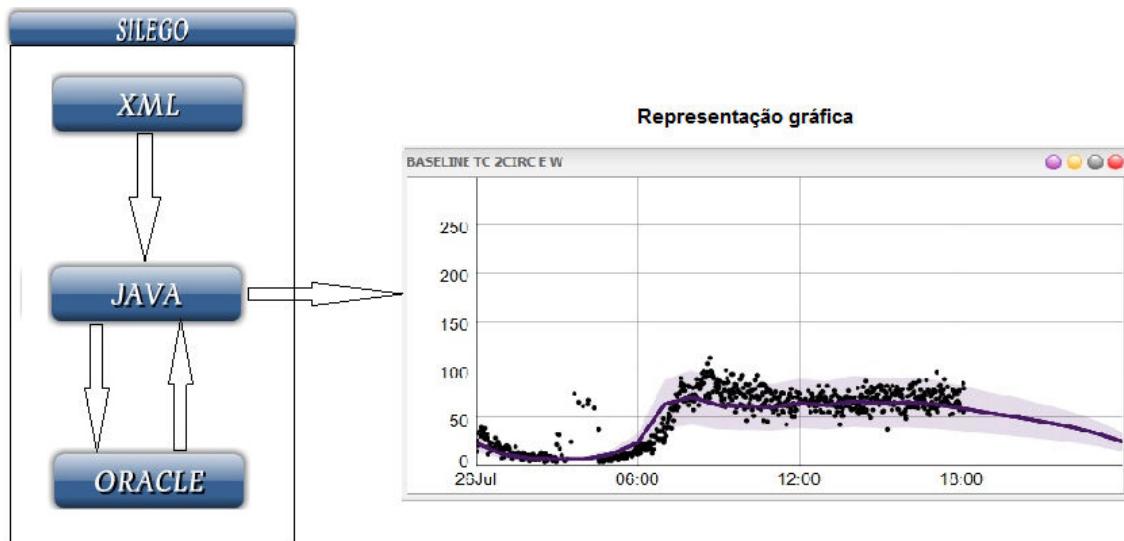


Figura 1 – Silego e Baselines

Na arquitectura geral do trabalho foram usadas três tecnologias: Java, XML e Oracle.

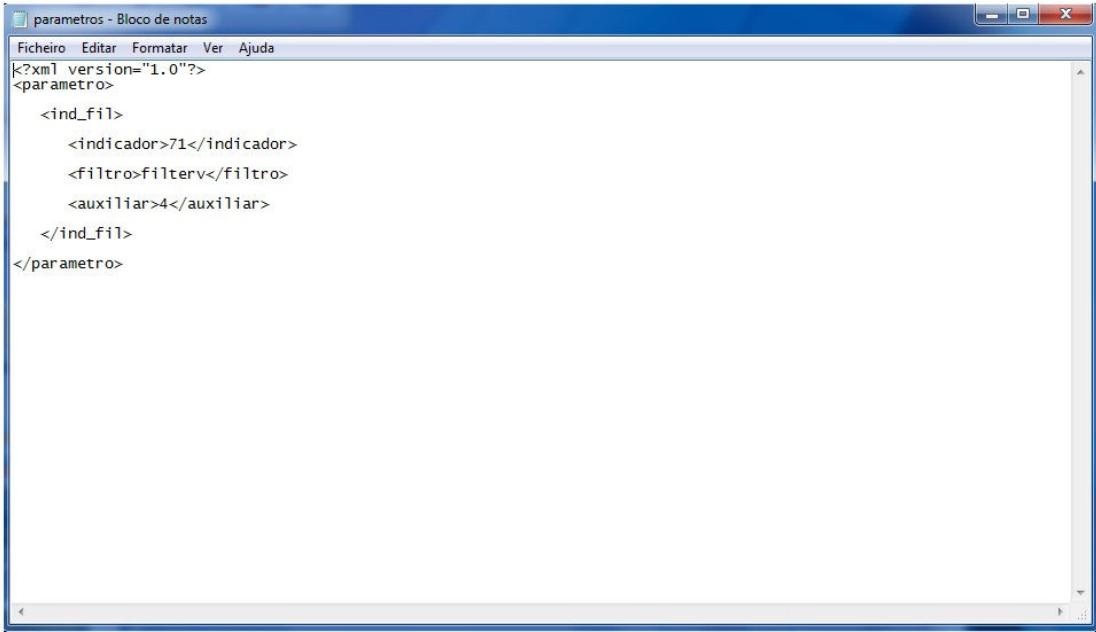
### 4.2 A tecnologia XML

*XML(Extensible Markup Language)* é uma tecnologia capaz de descrever diversos tipos de dados.

O XML é usado neste projecto para fornecer dados que são usados na chamada de procedimentos. A tecnologia fornece três valores, o indicador, o filtro e a opção. O indicador é o responsável pela definição e/ou escolha do sensor ao qual se pretende executar uma consulta *query*.

O filtro é usado para complementar a pesquisa, isto é, para definir o elemento a integrar a pesquisa.

A opção é responsável pela escolha do script a executar. Dentro da base de dados do Silego são definidos cinco *script's* que são seleccionados através deste parâmetro. Em baixo está a figura 1 que ilustra a definição do ficheiro *XML*.



```
<?xml version="1.0"?>
<parametro>
    <ind_fil>
        <indicador>71</indicador>
        <filtro>filterv</filtro>
        <auxiliar>4</auxiliar>
    </ind_fil>
</parametro>
```

Figura 2 – Definição do ficheiro XML

#### 4.3 A tecnologia JAVA

Essa tecnologia define as classes responsáveis pela leitura do documento *XML*, agendamento da execução dos procedimentos e a execução dos mesmos.

Na primeira fase é feita a leitura do documento *xml*, são retirados os três valores nele contigo, o indicador, o filtro e a opção. Esses valores são retirados e são guardados num objecto do tipo lista.

Na segunda fase o objecto do tipo lista é percorrido através de um ciclo e são retirados os dados nele guardado. É feita uma conexão com a base de dados e passados os dados para a execução do procedimento responsável pela execução dos outros.

Na terceira e última fase é feito o agendamento das tarefas a executar através da definição do tempo (a hora, o minuto e o segundo).

#### 4.3.1 A classe XML\_READER

A classe XML\_READER é a classe responsável pela leitura do ficheiro *xml*, pela criação de objecto do tipo lista (onde são guardados esses valores) e a conexão com a base de dados.

Esta começa por fazer uma leitura ao ficheiro *xml*, os valores lidos são colocados num objecto de *ArrayList* para depois retorná-los. O objecto é percorrido e os valores nele guardados são retirados, esses valores são depois utilizados para chamar o procedimento *procedure general*. Mas, antes é feita uma conexão com a base de dados de onde são executados os procedimentos.

#### 4.3.2 A classe Ind\_Filt\_Op

Esta classe é usada como auxiliar para a classe Xml\_Reader. São definidos os métodos GET e SET que são responsáveis pela definição e retorno dos valores.

#### 4.3.3 A classe Lembrete

A classe Lembrete é responsável pelo agendamento das tarefas, isto é, define a hora, o minuto e o segundo para a execução das tarefas. Quando é atingido o tempo definido, é disparado um *timer* que executa os métodos das classes xml\_Reader e Ind\_Filt\_Op.

### 4.4 O Oracle

É nessa tecnologia que são criados os *script's* que executam as pesquisas (*query*). Neste projecto são criados cinco *script's* que fornecem dados para construção dos gráficos que estarão disponíveis no centro de controlo da EP.

#### 4.4.1 Acidentes por Estradas e por Horas

Este *script* fornece os dados dos acidentes por estradas e por horas, ou seja, recebendo o indicador e o filtro, faz um inserir numa tabela destino do resultado da selecção retornada. Neste caso o *script* retorna por cada estrada e por cada hora o total dos acidentes.

#### 4.4.2 Acidentes por Estradas e por Mês

O *script* responsável pelo fornecimento dos números de acidentes que acontecem por estradas e por mês. Esse *script* recebe um indicador e um filtro, e coloca numa tabela o resultado da selecção retornada. O script retorna por cada estrada e por cada mês as ocorrências registadas.

#### 4.4.3 Acidentes por minutos, hora, dia, dias de semana e mês

Este *script* coloca numa tabela o número total de acidentes registados por minutos, hora, dia, dias de semana e mês. Este *script* recebe o indicador e o filtro e faz um *insert* do resultado retornado pela selecção.

#### 4.4.4 Acidentes por dias do mês

O *script* responsável pelo fornecimento dos números de acidentes por dias do mês. Este *script* guarda a soma total dos acidentes ocorridos por cada dia durante um mês, isso é, recebe um indicador e um filtro, e faz um *select* e coloca numa tabela o resultado retornado através de um *insert*.

#### 4.4.5 Acidentes por dias de semana

Este *script* fornece a soma dos acidentes por dias da semana. Este *script* faz um *insert* numa tabela do resultado de um *select* através do indicador e filtro fornecidos.

### 5. Desenho de interacção do sistema

#### 5.1 Visão Geral

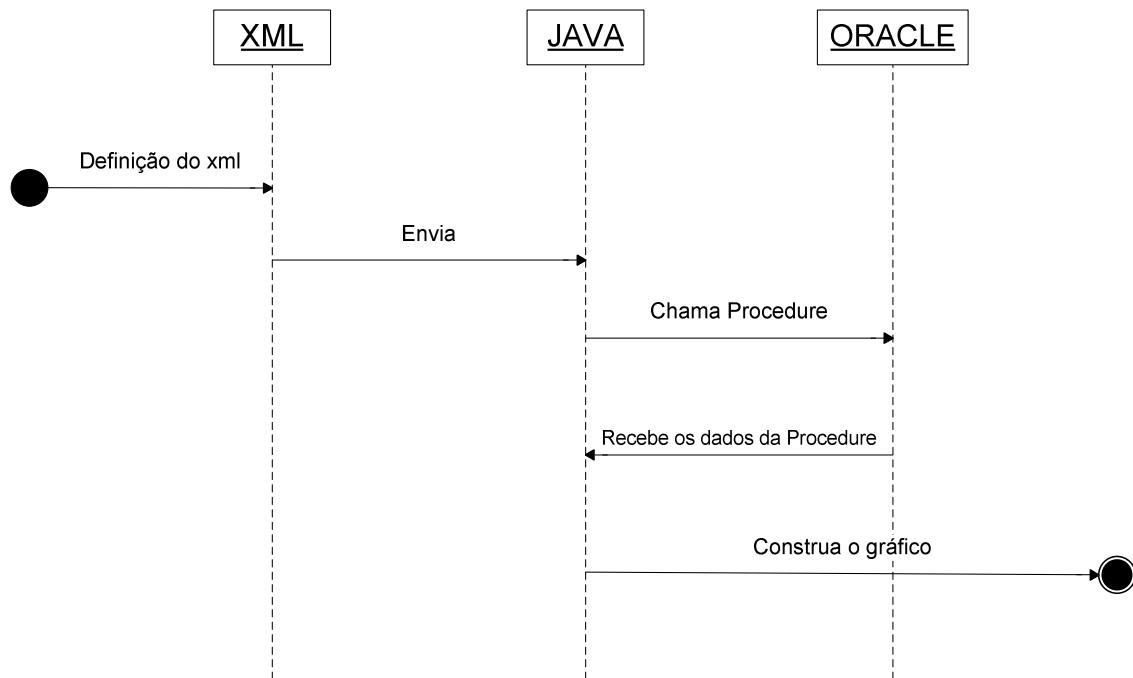


Figura 3 – Trocas de Informações

#### 5.2 Trocas de Informações

Na figura 3 está representado o esquema de trocas de informações entre vários módulos silego.

O processo da interacção entre os módulos começa com a definição dos valores (indicador, filtro e opção) através do xml. Estes valores são passados para o java, que os usa para chamar procedimentos.

O java conecta-se à base de dados do SILEGO, e fornece à procedure (Oracle) os valores recebidos do xml. Esta por sua vez, faz os cálculos necessários e retorna o resultado para o java.

O java usa o resultado recebido da procedure (Oracle) e gera o gráfico, que é neste caso o resultado final da interacção.

### 5.3. Representação Gráfica

A representação gráfica consiste na utilização de *baselines* gerados pelos procedimentos e os resultados que vão ser gerados em tempo real.

Os procedimentos vão buscar os dados à base de dados do Silego. Estes dados são usados para calcular os valores para construção dos gráficos.

A figura 2 mostra os gráficos gerados com os dados calculados pelos procedimentos (gráficos em linhas), e os gráficos que vão ser gerados em tempo real (gráfico em dispersão). O gráfico em dispersão, para além de nos dar o total dos carros a circularem nas estradas, também permite-nos ter a ideia da hora, do dia, e do mês com o maior e/ou o menor afluências.

Os resultados das representações gráficas nunca são iguais, porque dependem das horas, dos dias, dos dias da semana e dos meses.

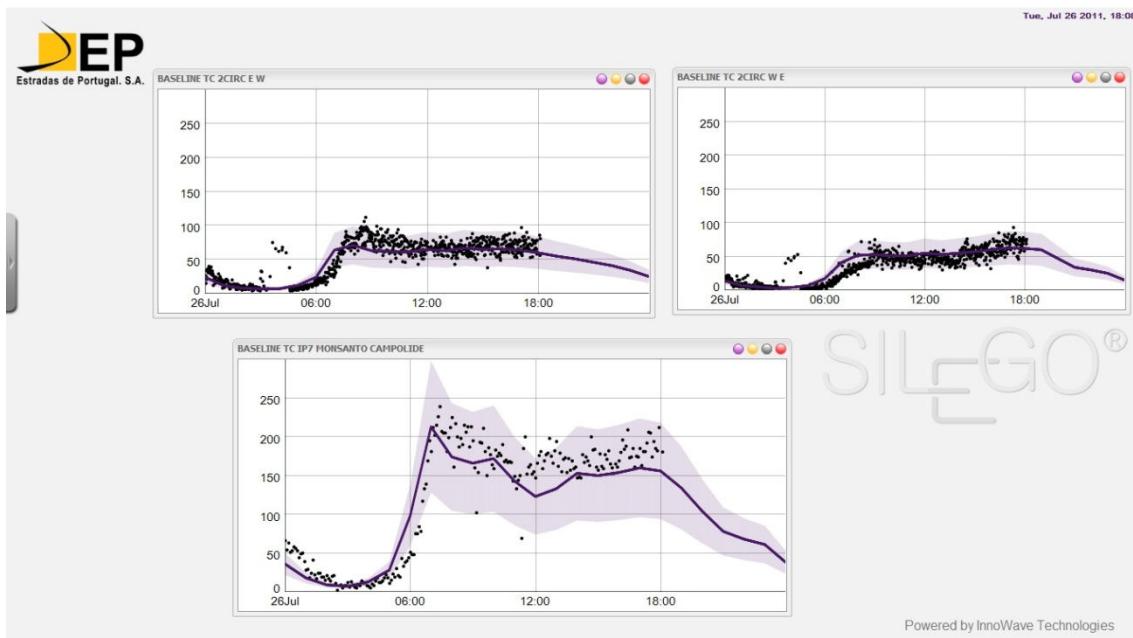


Figura 4 – Representação gráfica das baselines

## 6. Conclusão

O balanço que posso efectuar da minha passagem pela InnoWave Technologies é bastante positivo. Este período permitiu-me consolidar e aprofundar grande parte dos conhecimentos adquiridos em muitas cadeiras leccionadas na faculdade. A integração num ambiente de trabalho foi crucial para a aplicação prática desses conhecimentos teóricos, através da realização de actividades muito interessantes e com utilidade real.

Em suma, considero-me bastante satisfeito com o trabalho desenvolvido e com o conhecimento adquirido ao longo do período de estágio, estando também extremamente consciente que o mesmo resultou de um esforço genuíno da minha parte e da entidade acolhedora, que proporcionou uma oportunidade profissional de grande valor e mérito.

Posso considerar que este estágio foi produtivo pois, com a orientação dos profissionais que comigo trabalharam, pude aprender quais as abordagens que se devem efectuar quando da análise e realização de um projecto.

## 7. Bibliografia

- Junqueira Martins, Fernando Mário, JAVA6 e Programação Orientada pelos Objectos, FCA, 2009
- Rodrigues, António, Oracle 10g e 9i Para Profissionais, FCA, 2005
- Dias Damas, Luís Manuel, SQL - Structured Query Language, CFA, 2005
- Goya, Milton, PL/SQL - Procedures e Funções, 20/04/2011,  
[www.linhadecodigo.com.br](http://www.linhadecodigo.com.br)
- Oracle timestamps, 23/04/2011, <<http://psoug.org/reference/timestamp.html>>
- Oracle PL/SQL Tutorial, 11/03/2011, <<http://www.java2s.com>>

## 8. Anexo

### 8.1. XML

```
<?xml version="1.0"?>  
  
<parametro>  
    <ind_fil>  
        <indicador>71</indicador>  
        <filtro>filterv</filtro>  
        <auxiliar>4</auxiliar>  
    </ind_fil>  
</parametro>
```

### 8.2. JAVA

#### 8.2.1 Classe Ind\_Filt\_Op

```
public class Ind_Filt_Op {  
  
    /*Attribute declaration */  
  
    public Integer indicador, aux;  
    public String filtro;  
  
    public Integer getIndicador() {  
        return indicador;  
    }  
    public void setIndicador(Integer indicador) {  
        this.indicador = indicador;  
    }  
    public Integer getAux() {
```

```

        return aux;
    }
    public void setAux(Integer aux) {
        this.aux = aux;
    }
    public String getFiltro() {
        return filtro;
    }
    public void setFiltro(String filtro) {
        this.filtro = filtro;
    }
}

```

### 8.2.2.Classe Lembrete

```

import java.util.Calendar;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;

public class Lembrete {

    /*Attribute declaration */

    Timer timer;
    XmlReader xmlreader;

    /*The Lembrete method defines the time(hour, minute and second) that the tasks will be
    performed*/
    public Lembrete(){

        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY, 19);
        calendar.set(Calendar.MINUTE, 54);
        calendar.set(Calendar.SECOND, 0);
        Date time = calendar.getTime();

        timer = new Timer();
        timer.schedule(new TarefasAgendadas(), time);

    }

    /*This class defines the method responsible for the execution of the scheduled tasks*/
}

```

```

class TarefasAgendadas extends TimerTask {
    @SuppressWarnings("static-access")

    /*Method responsible for call the class XmlReader*/

    public void run() {

        xmlreader = new XmlReader();

        xmlreader.run();
        xmlreader.const_call_pro();

        timer.cancel();
    }
}

public static void main(String args[]) {
    new Lembrete();
    System.out.println("Tarefas Agendadas");
}
}

```

### 8.2.3.Classe XmlReader

```

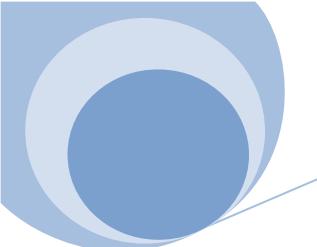
import java.io.File;
import java.util.ArrayList;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class XmlReader extends Thread{

    @SuppressWarnings("unused")
    private long checkTruncInterval;

    /* Creating an ArrayList which is of type the class Ind_Filt_Op */

```



```

static ArrayList<Ind_Filt_Op> list = new ArrayList<Ind_Filt_Op>();

/*This method receive an XML file, reads and places these data an object of ArrayList*/
public void run() {

    try {
        File file = new File("c:\\parametros.xml");
        DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document doc = db.parse(file);
        doc.getDocumentElement().normalize();
        System.out.println("Root element " +
doc.getDocumentElement().getTagName());
        NodeList nodeLst =
doc.getElementsByTagName("ind_fil");
        System.out.println("Indicadores - Filtros - Auxiliar");

        for (int s = 0; s < nodeLst.getLength(); s++) {

            Node indNode = nodeLst.item(s);

            if (indNode.getNodeType() ==
Node.ELEMENT_NODE) {

                Element indElmnt = (Element) indNode;
                NodeList indElmntLst =
indElmnt.getElementsByTagName("indicador");
                Element indNmElmnt = (Element)
indElmntLst.item(0);
                NodeList indicador =
indNmElmnt.getChildNodes();

                System.out.println("O indicador : " +
((Node) indicador.item(0)).getNodeValue().toString());

                NodeList fltNmElmntLst =
indElmnt.getElementsByTagName("filtro");
                Element fltNmElmnt = (Element)
fltNmElmntLst.item(0);

                NodeList filtro =
fltNmElmnt.getChildNodes();

                System.out.println("O filtro : " + ((Node)
filtro.item(0)).getNodeValue().toString());
            }
        }
    }
}

```

```

        NodeList auxNmElmntLst =
indElmnt.getElementsByTagName("auxiliar");
                           Element auxNmElmnt = (Element)
auxNmElmntLst.item(0);

        NodeList auxiliar =
auxNmElmnt.getChildNodes();

        System.out.println("O auxiliar : " + ((Node)
auxiliar.item(0)).getNodeValue().toString());

        Ind_Filt_Op obj = new Ind_Filt_Op();

        obj.setIndicador(Integer.parseInt(indicador.item(0).getNodeValue()));

        obj.setAux(Integer.parseInt((auxiliar.item(0)).getNodeValue()));

        obj.setFiltro((filtro.item(0)).getNodeValue());

        list.add(obj);

    }

} catch (Exception e) {
    e.printStackTrace();
}

const_call_pro();
}
/*This method runs the list and retrieves the data contained.
 * Makes the connection to the database and call the procedures SQL*/

```

```

public static void const_call_pro(){

    while(true){

        for(int i=0; i<list.size(); i++){

            Ind_Filt_Op ifa = list.get(i);

            Integer ind = ifa.getIndicador();
            Integer aux = ifa.getAux();
            String filt = ifa.getFiltro();

            String url = "jdbc:oracle:thin:@innowave-
sdi.office-on-the.net:1521:dbdev";

```

```
try {
    Connection conn =
DriverManager.getConnection(url, "slg_engine", "inwslg_engine");
    conn.setAutoCommit(false);
    CallableStatement cs =
conn.prepareCall("begin PRO_GENERAL(:1,:2, :3); end;");
    cs.setInt(1,ind);
    cs.setString(2, filt);
    cs.setInt(3, aux);
    cs.execute();
    cs.close();
    conn.close();
} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
try{Thread.sleep(86400000);

}catch (InterruptedException e) {

e.printStackTrace();
}
}
}
```

### 8.3. Oracle

#### 8.3.1 Procedure General

create or replace

```
PROCEDURE PRO_GENERAL
```

```
(
```

```
    INDICATORV NUMBER,
```

```
    FILTERV varchar2,
```

```
    AUX NUMBER
```

```
) AS
```

```
BEGIN
```

```
    IF AUX = 1 THEN
```

```
        SP_ACIDENTES_ESTRADAS_MÊS(INDICATORV,FILTERV);
```

```
    ELSIF AUX = 2 THEN
```

```
        SP_MIN_HOUR_DAY_WEEDAY_MONTH(INDICATORV,FILTERV);
```

```
    ELSIF AUX = 3 THEN
```

```
        SP_MONTHDAYBL(INDICATORV,FILTERV);
```

```
    ELSIF AUX = 4 THEN
```

```
        SP_WEEKDAYBL_TC1(INDICATORV,FILTERV);
```

```
    ELSIF AUX = 5 THEN
```

```
        SP_ACIDENTES_ESTRADAS_HORAS(INDICATORV,FILTERV);
```

```
    ELSE
```

```
        DBMS_OUTPUT.PUT_LINE('INTRODUZA UM NÚMERO ENTRE 1 E 5...');
```

```
    END IF;
```

```
END PRO_GENERAL;
```

### 8.3.2. Procedure Acidentes\_Estradas\_Horas

create or replace

```
PROCEDURE SP_ACIDENTES_ESTRADAS_HORAS
```

```
(
```

```
    INDICATORV IN VARCHAR2
```

```
, FILTERV IN VARCHAR2
```

```
) AS
```

```
VSTR VARCHAR2(1000);
```

```
BEGIN
```

```
VSTR:='DELETE FROM SLG_IA_MODEL_EXEMPLO_02 WHERE INDICATOR ='||  
INDICATORV||'
```

```
AND FILTER ='||FILTERV||';
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(VSTR);
```

```
EXECUTE IMMEDIATE VSTR;
```

```
COMMIT;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
ROLLBACK;
```

```
END ;
```

```
VSTR:='INSERT INTO SLG_ACIDENTES_HORAS_ESTRADAS SELECT
seq_id_ia_tb1.NEXTVAL, a.* FROM (
    SELECT TO_CHAR(DESC_STATUS) ACIDENTES,
TO_CHAR(SITUATIONDESCRIPTION) ESTRADAS, "MONTH_BL" TYPE,
TO_CHAR(TIMESTAMP,"HH24") HOUR,
SYSDATE LAST_UPD, "BACTHBL" USR_UPD, ""|| FILTERV || " FILTER , ' ||
INDICATORV ||' INDICATOR FROM SLG_DATA_FACT_E0_T100
GROUP BY TO_CHAR(DESC_STATUS),TO_CHAR(SITUATIONDESCRIPTION),
TO_CHAR(TIMESTAMP,"HH24")
ORDER BY TO_CHAR(DESC_STATUS),TO_CHAR(SITUATIONDESCRIPTION),
TO_CHAR(TIMESTAMP,"HH24"))a';
DBMS_OUTPUT.PUT_LINE(VSTR);
EXECUTE IMMEDIATE VSTR;
COMMIT;
END SP_ACIDENTES_ESTRADAS_HORAS;
```

### 8.3.3. Procedure Acidentes\_Estradas\_Mês

```
create or replace
PROCEDURE SP_ACIDENTES_ESTRADAS_MÊS
(
    INDICATORV IN NUMBER,
    FILTERV IN VARCHAR2
) AS
VSTR VARCHAR2(1000);
```

BEGIN

VSTR:='DELETE FROM SLG\_ACIDENTES\_ESTRADAS\_MÊS';

BEGIN

DBMS\_OUTPUT.PUT\_LINE(VSTR);

EXECUTE IMMEDIATE VSTR;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END ;

VSTR:='INSERT INTO SLG\_ACIDENTES\_ESTRADAS\_MÊS SELECT  
seq\_id\_ia\_tb1.NEXTVAL, a.\* FROM (

select count(acidentes)ACIDENTES, A.estrad

from (

select situationdescription AS acidentes, roadname as estradas

from slg\_data\_fact\_e0\_t100

where desc\_status="Novo" and timestamp <=(sysdate - 30)

) A

```
group by A.estradas ) a';  
  
DBMS_OUTPUT.PUT_LINE(VSTR);  
EXECUTE IMMEDIATE VSTR;  
  
COMMIT;  
  
END SP_ACIDENTES_ESTRADAS_MÊS;
```

#### 8.3.4. Procedure Acidentes\_min\_hour\_day\_weekday\_month

create or replace

```
PROCEDURE SP_MIN_HOUR_DAY_WEEDAY_MONTH  
(  
    INDICATORV IN NUMBER  
    , FILTERV IN VARCHAR2  
    ) AS  
    VSTR VARCHAR2(1000);  
  
BEGIN  
  
    VSTR:='DELETE FROM SLG_IA_MODEL_EXEMPLO_02 WHERE INDICATOR ='||  
    INDICATORV||'  
    AND FILTER ='"||FILTERV||"';  
  
    BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(VSTR);
EXECUTE IMMEDIATE VSTR;
COMMIT;

EXCEPTION
WHEN OTHERS THEN
ROLLBACK;

END ;

VSTR:='INSERT INTO SLG_IA_MODEL_EXEMPLO_02 SELECT
seq_id_ia_tb1.NEXTVAL, a.* FROM (
SELECT ROUND((SUM(VALOR))) VAL, "MONTH_BL" TYPE,
TO_CHAR(TIMESTAMP,"MI") MIN,
TO_CHAR(TIMESTAMP,"HH24") HOUR,
TO_CHAR(TIMESTAMP,"DD") DAY,
TO_CHAR(TIMESTAMP,"MM") MONTH,
TO_CHAR(TIMESTAMP,"D") WEEKDAY,
SYSDATE LAST_UPD, "BACTHBL" USR_UPD, ""|| FILTERV || " FILTER , " ||
INDICATORV || INDICATOR FROM TESTES
GROUP BY TO_CHAR(TIMESTAMP,"MM"),TO_CHAR(TIMESTAMP,"DD"),
TO_CHAR(TIMESTAMP,"D"),TO_CHAR(TIMESTAMP,"HH24"),
TO_CHAR(TIMESTAMP,"MI")
ORDER BY TO_CHAR(TIMESTAMP,"MM"),TO_CHAR(TIMESTAMP,"DD"),
TO_CHAR(TIMESTAMP,"D"),TO_CHAR(TIMESTAMP,"HH24"),
TO_CHAR(TIMESTAMP,"MI")) a';
DBMS_OUTPUT.PUT_LINE(VSTR);
```

```
EXECUTE IMMEDIATE VSTR;
```

```
COMMIT;
```

```
END SP_MIN_HOUR_DAY_WEEDAY_MONTH;
```

#### 8.3.4. Procedure Month\_DayBL

create or replace

```
PROCEDURE SP_MONTHDAYBL
```

```
(
```

```
    INDICATORV IN NUMBER,
```

```
    FILTERV IN VARCHAR2
```

```
) AS
```

```
    VSTR VARCHAR2(1000);
```

```
BEGIN
```

```
    VSTR:='DELETE FROM SLG_IA_MODEL_EXEMPLO WHERE INDICATOR ='||  
    INDICATORV||'
```

```
    AND FILTER ='"||FILTERV||"';
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE(VSTR);
```

```
    EXECUTE IMMEDIATE VSTR;
```

```
    COMMIT;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

ROLLBACK;

END ;

VSTR:='INSERT INTO SLG\_IA\_MODEL\_EXEMPLO SELECT seq\_id\_ia\_tb1.NEXTVAL,  
a.\* FROM (

SELECT ROUND((SUM(VALUE))) VAL, "MONTH\_BL" TYPE, NULL MIN, NULL  
HOUR,

TO\_CHAR(TIMESTAMP,"DD") DAY,

TO\_CHAR(TIMESTAMP,"MM") MONTH,

TO\_CHAR(TIMESTAMP,"D") WEEKDAY,

SYSDATE LAST\_UPD, "BACTHBL" USR\_UPD, ""|| FILTERV || " FILTER , '||  
INDICATORV ||' INDICATOR FROM SLG\_DATA\_IND\_ST\_E0\_I71\_A2

GROUP BY TO\_CHAR(TIMESTAMP,"MM"), TO\_CHAR(TIMESTAMP,"DD"),  
TO\_CHAR(TIMESTAMP,"D")

ORDER BY TO\_CHAR(TIMESTAMP,"MM"), TO\_CHAR(TIMESTAMP,"DD"),  
TO\_CHAR(TIMESTAMP,"D")) a';

DBMS\_OUTPUT.PUT\_LINE(VSTR);

EXECUTE IMMEDIATE VSTR;

COMMIT;

END SP\_MONTHDAYBL;

### 8.3.5. Procedure WeekDay

create or replace

```
PROCEDURE SP_WEEKDAYBL_TC1(
    INDICATORV NUMBER,
    FILTERV  VARCHAR2)
AS
    VSTR VARCHAR2(1000);
BEGIN
    VSTR := 'DELETE FROM SLG_IA_MODEL_TB1 WHERE INDICATOR = ' ||
    INDICATORV|| ' AND FILTER = "' || FILTERV || '"';
    BEGIN
        DBMS_OUTPUT.PUT_LINE(VSTR);
        EXECUTE IMMEDIATE VSTR;
        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
    END;
    VSTR := 'INSERT INTO slg_ia_model_tb1
        SELECT seq_id_ia_tb1.NEXTVAL, a.*
        FROM (SELECT  ROUND (SUM (VALUE)) VAL, "WEEKDAY_BL" TYPE, NULL MIN,
                    TO_CHAR (TIMESTAMP, "HH24") HOUR, NULL DAY, NULL MONTH,
                    TO_CHAR (TIMESTAMP,"D") WEEKDAY, SYSDATE UPDATE_DT,
                    "BACTHBL" UPDATE_USR,
                    ""|| FILTERV || "" FILTER , ''|| INDICATORV ||' INDICATOR
        FROM slg_data_ind_st_e0_i71_a2
        group by TO_CHAR (TIMESTAMP, "D"), TO_CHAR (TIMESTAMP, "HH24")'
```

```
ORDER BY TO_CHAR (TIMESTAMP, "D"), TO_CHAR (TIMESTAMP, "HH24")) a';  
  
DBMS_OUTPUT.PUT_LINE(VSTR);  
EXECUTE IMMEDIATE VSTR;  
  
COMMIT;  
END SP_WEEKDAYBL_TC1;
```