Projeto HUMV UFRB - Sistema de Gestão do Hospital Veterinário da UFRB

Laboratório de Sistemas de Software (LaSiS)

Centro de Ciências Exatas e Tecnológicas (CETEC)

Universidade Federal do Recôncavo da Bahia (UFRB)

**Instruções para configuração do ambiente de desenvolvimento**

1. Ferramentas utilizadas para o desenvolvimento
   1. Conta de usuário no site github.com
   2. Conta de usuário no site trello.com
   3. humv\_backend + humv\_desktop + banco de dados
      1. IDE Eclipse Mars (Java EE)
      2. Servidor Apache Tomcat 8.0.33
      3. Banco de dados PostgreSQL 9.4.7
      4. JDK 1.8 (Java 8)
      5. IDE Netbeans 8.1
2. Configurações
   1. Clonar o projeto humv\_backend através do programa Github Desktop (repositório humv)
   2. Para o humv\_backend + banco de dados
      1. Configurar Apache Tomcat no Eclipse (adicionar servidor)
      2. Instalar o PostgreSQL
         1. Criar um database chamado ***humv\_db***
         2. Criar um database chamado ***humv\_log***
         3. Criar um usuário chamado ***humv***
         4. A senha do usuário deverá ser ***lAs1s\_UFRB***
         5. Executar script (última página) no database **humv\_log**
         6. Altere a porta do banco de dados para **15432**
      3. Importar projeto humv\_backend no Eclipse
   3. Para o humv\_desktop
      1. Importar projeto humv\_desktop no Netbeans

**Servidor principal do ambiente de produção**

1. IP do servidor: 10.110.110.110
2. PostgreSQL: porta 15432
3. Apache Tomcat: porta 80
4. Jenkins: porta 18080
5. Backup do banco de dados armazenado em /home/backup-db.sql
6. Inserir o seguinte trecho de código no arquivo do Apache Tomcat <TOMCAT\_HOME>/conf/context.xml:

<Manager className="org.apache.catalina.session.PersistentManager" saveOnRestart="false" maxActiveSessions="-1" minIdleSwap="-1" maxIdleSwap="-1" maxIdleBackup="-1">

<Store className="org.apache.catalina.session.FileStore"/>

</Manager>

1. Criar usuário para o Apache Tomcat (username: humv / senha: lAs1s\_UFRB) no arquivo <TOMCAT\_HOME>/conf/tomcat-users.xml:

<role rolename="manager"/>

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="admin"/>

<role rolename="admin-gui"/>

<user username="humv" password="lAs1s\_UFRB" roles="manager,manager-gui,manager-script,admin,admin-gui"/>

OBS.: lembrar de sempre utilizá-lo para fazer deploy de novas versões da aplicação.

1. No arquivo /conf/server.xml, mudar o código:

<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

Para o trecho abaixo:

<Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" URIEncoding="UTF-8" compressableMimeType="text/html,text/xml,text/plain,text/css,text/ javascript,application/x-javascript,application/javascript"/>

-- Logback: the reliable, generic, fast and flexible logging framework.

-- Copyright (C) 1999-2010, QOS.ch. All rights reserved.

-- See http://logback.qos.ch/license.html for the applicable licensing

-- conditions.

-- This SQL script creates the required tables by ch.qos.logback.classic.db.DBAppender

-- It is intended for PostgreSQL databases.

-- DROP TABLE logging\_event\_property;

-- DROP TABLE logging\_event\_exception;

-- DROP TABLE logging\_event;

-- DROP SEQUENCE logging\_event\_id\_seq;

CREATE SEQUENCE logging\_event\_id\_seq MINVALUE 1 START 1;

CREATE TABLE logging\_event

(

timestmp BIGINT NOT NULL,

formatted\_message TEXT NOT NULL,

logger\_name VARCHAR(254) NOT NULL,

level\_string VARCHAR(254) NOT NULL,

thread\_name VARCHAR(254),

reference\_flag SMALLINT,

arg0 VARCHAR(254),

arg1 VARCHAR(254),

arg2 VARCHAR(254),

arg3 VARCHAR(254),

caller\_filename VARCHAR(254) NOT NULL,

caller\_class VARCHAR(254) NOT NULL,

caller\_method VARCHAR(254) NOT NULL,

caller\_line CHAR(4) NOT NULL,

event\_id BIGINT DEFAULT nextval('logging\_event\_id\_seq') PRIMARY KEY

);

CREATE TABLE logging\_event\_property

(

event\_id BIGINT NOT NULL,

mapped\_key VARCHAR(254) NOT NULL,

mapped\_value VARCHAR(1024),

PRIMARY KEY(event\_id, mapped\_key),

FOREIGN KEY (event\_id) REFERENCES logging\_event(event\_id)

);

CREATE TABLE logging\_event\_exception

(

event\_id BIGINT NOT NULL,

i SMALLINT NOT NULL,

trace\_line VARCHAR(254) NOT NULL,

PRIMARY KEY(event\_id, i),

FOREIGN KEY (event\_id) REFERENCES logging\_event(event\_id)

);

GRANT ALL PRIVILEGES ON TABLE logging\_event TO humv;

GRANT ALL PRIVILEGES ON TABLE logging\_event\_property TO humv;

GRANT ALL PRIVILEGES ON TABLE logging\_event\_exception TO humv;

GRANT ALL PRIVILEGES ON TABLE logging\_event\_id\_seq TO humv;

**Servidor de backup do ambiente de produção**

Backup do banco de dados armazenado em /home/backup-db.sql