

## A. Apêndice

### A.1. Script de Instalação do CouchDB

```
1 #!/bin/sh
2 sudo apt-get update || true
3 sudo apt-get --no-install-recommends -y install \
4 build-essential pkg-config runit erlang \
5     libc6-dev libmozjs185-dev libcurl4-openssl-dev
6 wget http://apache-mirror.rbc.ru/pub/apache/couchdb/source/2.1.1/apache
7     -couchdb-2.1.1.tar.gz
8 tar -xvzf apache-couchdb-2.1.1.tar.gz
9 cd apache-couchdb-2.1.1/
10 ./configure && make release
11 sudo adduser --system \
12     --no-create-home \
13     --shell /bin/bash \
14     --group --gecos \
15     "CouchDB Administrator" couchdb
16 sudo cp -R rel/couchdb /home/couchdb
17 sudo chown -R couchdb:couchdb /home/couchdb
18 sudo find /home/couchdb -type d -exec chmod 0770 {} \;
19 sudo sh -c 'chmod 0644 /home/couchdb/etc/*'
20 sudo mkdir /var/log/couchdb
21 sudo chown couchdb:couchdb /var/log/couchdb
22 sudo mkdir /etc/sv/couchdb
23 sudo mkdir /etc/sv/couchdb/log
24 cat > run << EOF
25     #!/bin/sh
26     export HOME=/home/couchdb
27     exec 2>&1
28     exec chpst -u couchdb /home/couchdb/bin/couchdb
29 EOF
30 cat > log_run << EOF
31     #!/bin/sh
32     exec svlogd -tt /var/log/couchdb
33 EOF
34
35 sudo mv ./run /etc/sv/couchdb/run
36 sudo mv ./log_run /etc/sv/couchdb/log/run
37 sudo chmod u+x /etc/sv/couchdb/run
38 sudo chmod u+x /etc/sv/couchdb/log/run
39 sudo ln -s /etc/sv/couchdb/ /etc/service/couchdb
40 sleep 5
41 sudo sv status couchdb
```

Listing 1. Script de instalação do Banco de Dados.

### A.2. Configurar o CouchDB em Cluster

É necessário iniciar o serviço

```
#!/etc/init.d/couchdb start
```

Depois, criar um usuário administrador, pois o cluster não funcionará se não houver um administrador definido. Para este trabalho, o login definido é “*admin*” e a senha “*couchdb*”.

```
#curl -X PUT http://127.0.0.1:5984/_node/couchdb@<IP_Servidor>/_config/  
admins/admin -d '<senha>'
```

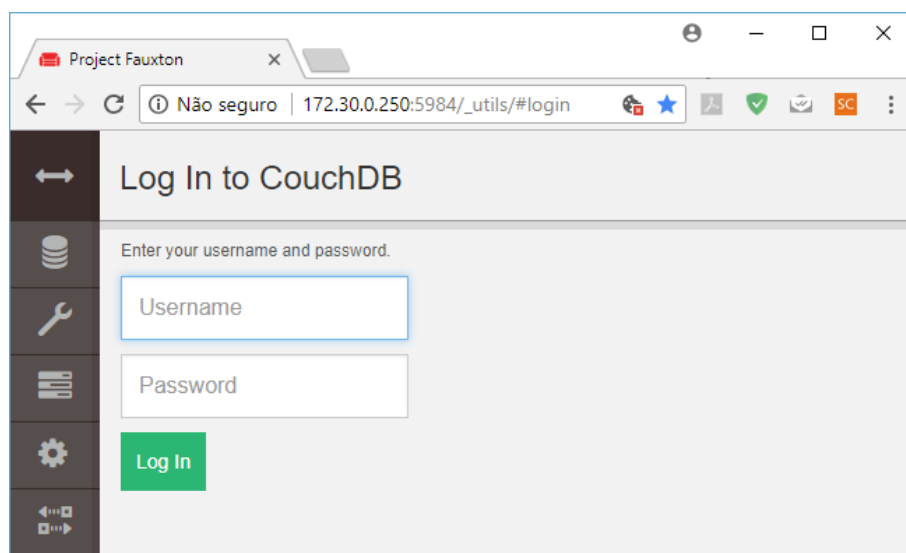
Em seguida, atribuo que o *CouchDB* escute todas as interfaces de rede, porque é obrigatório que um cluster tenha nós capazes de se comunicar uns com os outros.

```
#curl -X PUT http://<usuario_admin>:<senha>@127.0.0.1:5984/_node/  
couchdb@<IP_Servidor>/_config/httpsd/bind_address -d '0.0.0.0'
```

Observação! É preciso repetir essas três etapas (inicialização do servidor, criação da conta de administrador e configuração do endereço de ligação do servidor) nos dois nós do cluster. Todos os nós do cluster devem ter a mesma conta de administrador (o mesmo login e senha).

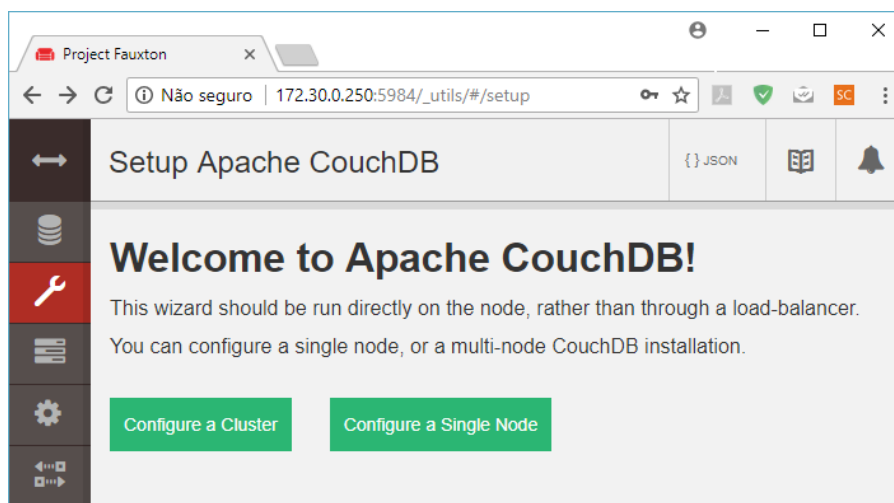
Ao acessar a interface de administração *Web* do *Fauxton*, para configurar o *cluster*, a interface do *Fauxton* pode ser acessada através da URL:

```
http://<IP_Servidor>:5984/_utils/
```



**Figura 12. Tela de Autenticação do CouchDB.**

Efetue o *login* (Figura 12) usando a conta de administrador e clique em “Setup” do menu esquerdo. Ao acessar a página, vá em “Configurar *Cluster*” (Figura 13) e depois em Create Admin Credentials insira o usuário e senha com credenciais de administrador, em *Bind Address the node will listen on* insira 0.0.0.0, em *Port that the node will use* 5984 e em *Number of Nodes to be added to the cluster (Including this one)*, insira o mesmo número de nós existentes, neste trabalho são 2.



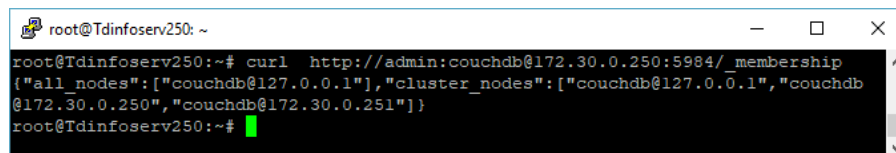
**Figura 13. Configuração do *Cluster*.**

Na parte inferior do formulário, clique em “*Add Node*” (Figura 14).

**Figura 14. Interface de Configuração do *Cluster*.**

É preciso repetir este processo no outro nó. É possível verificar a configuração de *cluster* usando o endpoint REST `_membership`.

```
# curl http://<usuario>:<senha>@<IP_Servidor>:5984/_membership
```

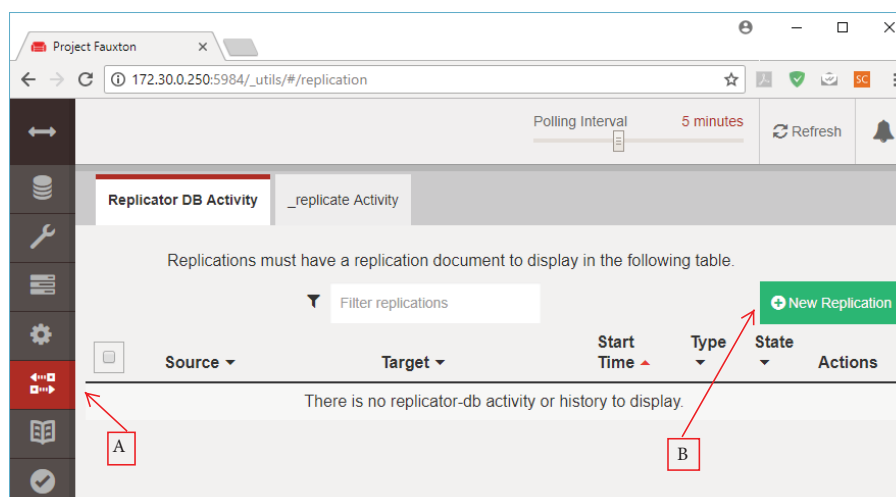


```
root@Tdinfoserv250: ~  
root@Tdinfoserv250:~# curl http://admin:couchdb@172.30.0.250:5984/_membership  
{  
  "all_nodes": ["couchdb@127.0.0.1"],  
  "cluster_nodes": ["couchdb@127.0.0.1", "couchdb@172.30.0.250", "couchdb@172.30.0.251"]  
}
```

**Figura 15. Verificar a Configuração de *Cluster*.**

### A.3. Configurar o Serviço de Replicação no *CouchDB*

Ao solicitar que o *CouchDB* replique um banco de dados para outro, ele irá comparar os dois bancos de dados para descobrir quais documentos na origem são diferentes do destino e enviar para depois enviar os documentos alterados para o destino até que todas as alterações sejam transferidas. As alterações incluem novos documentos, documentos alterados e documentos excluídos. Documentos que já existem no destino na mesma versão não são transferidos, apenas as versões mais recentes. Para acessar o serviço de replicação de dados do *CouchDB* acesse a aba no lado esquerdo (Figura 16 - A) da interface de administração *Web* do *Fauxton*.



**Figura 16. Interface do Serviço de Replicação.**

A seguir, clique no botão ao lado direito *New Replication* (Figura 16 - B), a interface de configuração do serviço de replicação será apresentada (Figura 17).

The screenshot shows a web browser window with the address bar displaying "172.30.0.250:5984/\_utils/#/replication/\_create". The page title is "Job Configuration". On the left, a sidebar menu includes "Databases", "Setup", "Active Tasks", "Configuration", "Replication" (highlighted in red), "Documentation", "Verify", and "Your Account". At the bottom of the sidebar, it says "CouchDB", "Fauxton on Apache CouchDB v. 2.1.1", and "Log Out admin". The main content area contains four configuration fields: "Replication Source:" with a dropdown menu showing "Select source"; "Replication Target:" with a dropdown menu showing "Select target"; "Replication Type:" with a dropdown menu showing "One time"; and "Replication Document:" with a text input field containing "Custom ID (optional)" and a clear button (X). Below these fields are two buttons: a green "Start Replication" button and a red "Clear" button.

**Figura 17. Configuração do Serviço de Replicação.**

A Figura 18 apresenta o formulário já preenchido, em *Replication Source* informe o local da base de dados de origem, *Source Name* informe o nome da base de dados a ser replicada, em *Replication Target* informe se será replicado para uma base de dados local ou remota já existente, ou será replicado para uma nova base de dados local ou remota. Em *Replication Type* informe se o processo de replicação será realizado somente uma vez ou se é contínuo, e no último campo *Replication Document* deve-se informar o ID do documento. Contudo, este último campo é opcional a inserção de dados. Para executar, clique em *Start Replication*.

Project Fauxton

172.30.0.250:5984/\_utils/#/replication/\_create

### Job Configuration

Replication Source: Local database

Source Name: sistema-saude-medicamento

Replication Target: New remote database

New Database: http://172.30.0.51:5984/sistema-saude-medicamento

https://\$USERNAME:\$PASSWORD@\$REMOTE\_SERVER/\$DATABASE  
If a "new" database already exists, data will replicate into that existing database.

Replication Type: Continuous

Replication Document: Custom ID (optional)

**Start Replication** Clear

**CouchDB**

Fauxton on Apache CouchDB v. 2.1.1

Log Out admin

**Figura 18. Formulário Preenchido.**

O processo de replicação será realizado conforme é apresentado na Figura 19. A replicação sincroniza duas cópias da mesma base de dados, e permite que os usuários tenham acesso com baixa latência aos dados, independentemente da sua localização. Essas bases de dados podem estar localizadas no mesmo servidor ou em servidores diferentes. Ao alterar uma cópia da base de dados, a replicação enviará essas alterações para a outra cópia.

Replicator DB Activity

Replications must have a replication document to display in the following table.

Filter replications

Source	Target	Start Time	Type	State	Actions
http://172.30.0.250:5984/sistema-saude	http://172.30.0.251:5984/sistema-saude	Apr 23rd, 9:01 pm	Continuous	Running	

**New Replication**

**Figura 19. Replicação em Execução.**

A base de dados no *CouchDB* possui um número de sequência que é incrementado toda vez que o banco de dados é alterado. O *CouchDB* armazena quais alterações foram realizadas e com qual número de sequência. Dessa forma, o *CouchDB* encontra diferenças entre as bases de dados de maneira mais eficiente, o que aumenta também a robustez da replicação.

## B. Instalação e Configuração do *Apache Kafka* e *Zookeeper*

Nesta seção são apresetados os comandos que foram utilizados para instalar e configurar o *Apache Kafka* e *Zookeeper*. Para maiores informações acesse o *site* <http://kafka.apache.org> e <https://zookeeper.apache.org>. Inicialmente é necessário efetuar a instalação do Java, caso sua máquina não tenha esta aplicação já instalada. Para realizar a instalação utilize o seguinte comando:

```
#apt-get install default-jdk
```

Após a instalação do Java, é necessário instalar o *Apache Zookeeper*, a versão utilizada neste trabalho foi a *release 3.4.9-3*.

```
#apt-get install zookeeper
```

Para configurar o *Zookeeper*, acesse o diretório “/etc/zookeeper/conf”, depois edite o arquivo “zoo.cfg” e insira (caso necessário) as seguintes linhas de comando.

```
1 #tickTime=2000
2 #initLimit=10
3 #syncLimit=5
4 #dataDir=/var/lib/zookeeper
5 #clientPort=2181
6 #server.1=Tdinfoserv252:2888:3888
7 #server.2=Tdinfoserv253:2888:3888
8 #server.3=Tdinfoserv254:2888:3888
```

Após as modificações no arquivo “zoo.cfg”, é necessário criar as identificações dos *brokers* em cada um dos 3 servidores. Neste trabalho, aos servidores serão atribuídos os seguintes nomes: *Tdinfoserv252*, *Tdinfoserv253* e *Tdinfoserv254*.

- No servidor “*Tdinfoserv252*”:

```
#echo "1" > /etc/zookeeper/myid
```

- No servidor “*Tdinfoserv253*”:

```
#echo "2" > /etc/zookeeper/myid
```

- No servidor “*Tdinfoserv254*”:

```
#echo "3" > /etc/zookeeper/myid
```

Também é necessário instalar o *Apache Kafka*, a versão utilizada foi a *release 2.12-1.1.0*

```
1 #cd /opt
2 #wget http://ftp.unicamp.br/pub/apache/kafka/1.1.0/kafka_2.12-1.1.0.tgz
3 #tar -zxvf kafka_2.12-1.1.0.tgz
4 #cd kafka_2.12-1.1.0/config
```

Edite o arquivo “server.properties”:

```
#nano server.properties
```

Neste arquivo é necessário ajustar os parâmetros do primeiro *Broker*:

```
1 #broker.id=0
2 #port=9092
3 #log.dir=/tmp/kafka0-logs
4 #host.name=Tdinfoserv252
5 #zookeeper.connect=Tdinfoserv252:2181,Tdinfoserv253:2181,Tdinfoserv254:2181
```

Após alterado faça duas cópias deste arquivo:

```
1 #cp server.properties server-1.properties
2 #cp server.properties server-2.properties
```

Por fim, edite os 2 novos arquivos (“server-1.properties” e “server-2.properties”). Abra o primeiro arquivo “server-1.properties” com um editor de textos, depois edite as seguintes linhas:

```
1 #broker.id=1
2 #port=9093
3 #log.dir=/tmp/kafka1-logs
4 #host.name=Tdinfoserv253
5 #zookeeper.connect=Tdinfoserv252:2181,Tdinfoserv253:2181,Tdinfoserv254:2181
```

Logo após, repita este processo no segundo arquivo “server-2.properties” com um editor de textos, depois edite as seguintes linhas:

```
1 #broker.id=2
2 #port=9094
3 #log.dir=/tmp/kafka2-logs
4 #host.name=Tdinfoserv254
5 #zookeeper.connect=Tdinfoserv252:2181,Tdinfoserv253:2181,Tdinfoserv254:2181
```

Após alterar os arquivos (“server-1.properties” e “server-2.properties”), efetue o mesmo procedimento nos outros 2 servidores para criar o *cluster*. Por fim, é necessário iniciar os serviços do *Zookeeper* nos 3 servidores.

```
#/usr/share/zookeeper/bin/zkServer.sh start
```

É necessário iniciar cada *Broker* do *Kafka* em cada servidor.

- Servidor 1: Tdinfoserv252

```
1 #/opt/kafka_2.12-1.1.0/bin/kafka-server-start.sh /opt/kafka_2.12-1.1.0/config/server.properties
```

- Servidor 2: Tdinfoserv253

```
1 #/opt/kafka_2.12-1.1.0/bin/kafka-server-start.sh /opt/kafka_2.12-1.1.0/config/server-1.properties
```

- Servidor 3: Tdinfoserv254



```
1 #/opt/kafka_2.12-1.1.0/bin/kafka-server-start.sh /opt/kafka_2.12-1.1.0/
   config/server-2.properties
```

Ao término desta etapa, já é possível executar as funcionalidades do *Apache Kafka*, conforme descrito na Seção 6.

### C. Instalação do *Zabbix Server*

Para o procedimento de instalação do *Zabbix Server* versão 3.4, adicione o repositório *Zabbix* e instale o pacote de configuração do repositório. Este pacote contém arquivos de configuração do “apt”(gerenciador de pacotes de *software*) para o *Ubuntu* 14.04.

```
1 #wget http://repo.zabbix.com/zabbix/3.4/ubuntu/pool/main/z/zabbix-
   release/zabbix-release_3.4-1+trusty_all.deb
2 #dpkg -i zabbix-release_3.4-1+trusty_all.deb
3 #apt-get update
```

A seguir é necessário efetuar a instalação do banco de dados e do *frontend*. Para instalar o servidor *Zabbix* com suporte ao SGBD (Sistema de Gerenciamento de Banco de Dados) *MySQL*, execute:

```
#apt-get install zabbix-server-mysql
```

E para instalar o *frontend* do *Zabbix* execute o comando:

```
#apt-get install zabbix-frontend-php
```

O próximo passo é criar uma base de dados no banco de dados *MySQL*. Esta base de dados é necessária para poder armazenar todos os dados que serão coletados pelo *Zabbix*, desta maneira é possível ter um histórico dos eventos ocorridos na rede de computadores.

```
shell>mysql -u root -p
```

**Observação! Aqui é solicitado a senha de usuário root do sistema.**

```
1 mysql>create database zabbixdb character set utf8 collate utf8_bin;
2 mysql>grant all privileges on zabbixdb.* to zabbixuser@localhost
   identified by '<password>';
3 mysql>FLUSH PRIVILEGES;
4 mysql>quit;
```

**Observação! Caso queira alterar a senha da base de dados criado.**

```
1 mysql>UPDATE mysql.user SET Password=PASSWORD('NOVA.SENHA') WHERE User
   ='USUARIO' AND Host='localhost';
```

Após criar a base de dados, é preciso importar o esquema inicial e os dados para o servidor com o *MySQL*:

```
1 #cd /usr/share/doc/zabbix-server-mysql/
2 #zcat create.sql.gz | mysql -u root -p zabbixdb
```

**Observação! É necessário inserir sua senha de banco de dados recém-criada.**

O próximo procedimento é configurar o arquivo “/etc/zabbix/zabbix\_server.conf”. Este arquivo possui os parâmetros de configuração para conexão com a base de dados “zabbixdb” que foi criada.

```
1 DBName = zabbixdb
2 DBUser = zabbixuser
3 DBPassword = <senha>
```

**Observação! No DBPassword, use a senha do banco de dados do Zabbix para MySQL.**

Com o arquivo “zabbix\_server.conf” configurado, inicie o processo “zabbix-server” e atribua que o serviço inicie com o sistema.

```
1 #service zabbix-server start
2 #update-rc.d zabbix-server enable
```

Logo a seguir, é necessário alterar o arquivo de configuração do servidor *Web* com *Apache*. Esta modificação é necessária para que seja inicializado sem problemas o *frontend* do *Zabbix* que está localizado em “/etc/apache2/conf-enabled/zabbix.conf”. Algumas configurações do PHP (*Hypertext Preprocessor*) já estão configuradas, é necessário descomentar a linha de configuração “date.timezone” e definir o fuso horário que corresponde ao fuso horário do estado e país no qual o servidor *Web* está localizado.

Antes:

```
1 php_value max_execution_time 300
2 php_value memory_limit 128M
3 php_value post_max_size 16M
4 php_value upload_max_filesize 2M
5 php_value max_input_time 300
6 php_value always_populate_raw_post_data -1
7 # php_value date.timezone Europa/Riga
```

Depois:

```
1 php_value max_execution_time 300
2 php_value memory_limit 128M
3 php_value post_max_size 16M
4 php_value upload_max_filesize 2M
5 php_value max_input_time 300
6 php_value always_populate_raw_post_data -1
7 php_value date.timezone America/Sao_Paulo
```

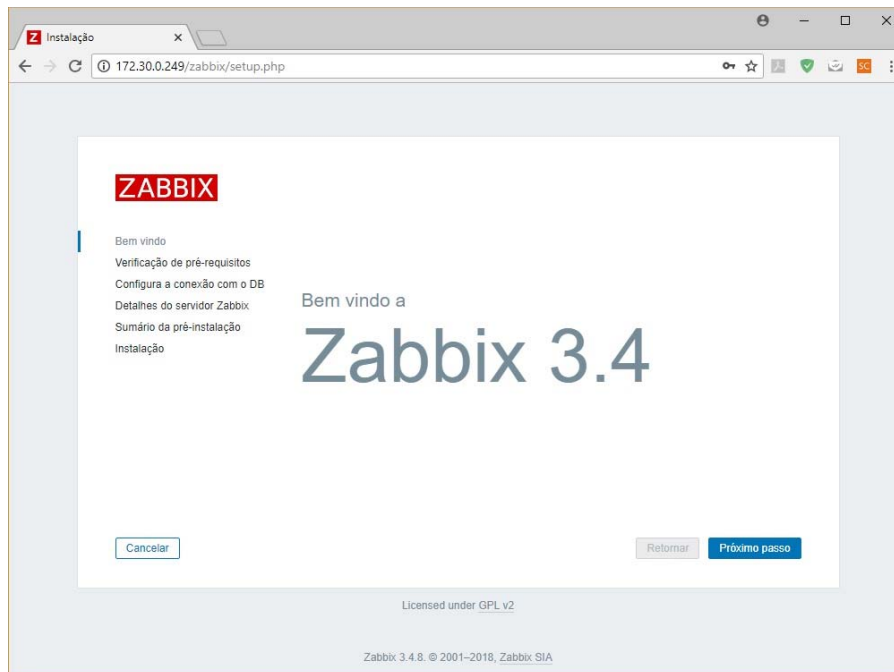
Com a configuração do *frontend* do *zabbix server* concluída, é necessário reiniciar o serviço “*apache2*” do servidor *Web*. As alterações que foram realizadas serão carregadas com o serviço.

```
#\etc\init.d\apache2 restart
```

O próximo passo agora é com as etapas de instalação do *frontend* do *Zabbix server* que permitirão o acesso ao *framework* de gerenciamento do *Zabbix*. Acesse o navegador *Web*, e abra a URL (*Uniform Resource Locator*):

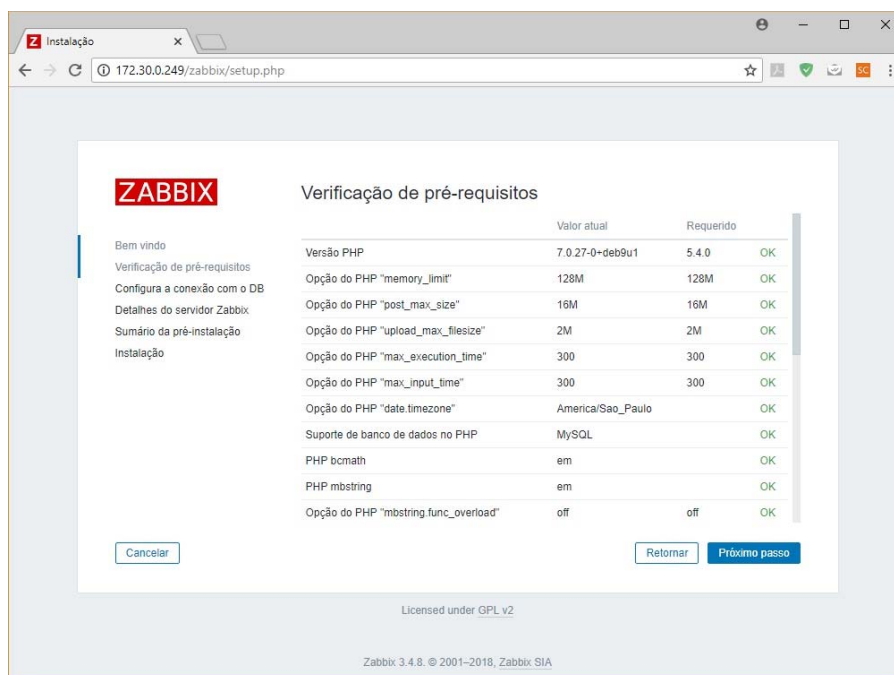
`http://<ip_do_servidor_ou_nome>/zabbix`

A Figura 20 apresenta a primeira tela do assistente de instalação do *frontend*.



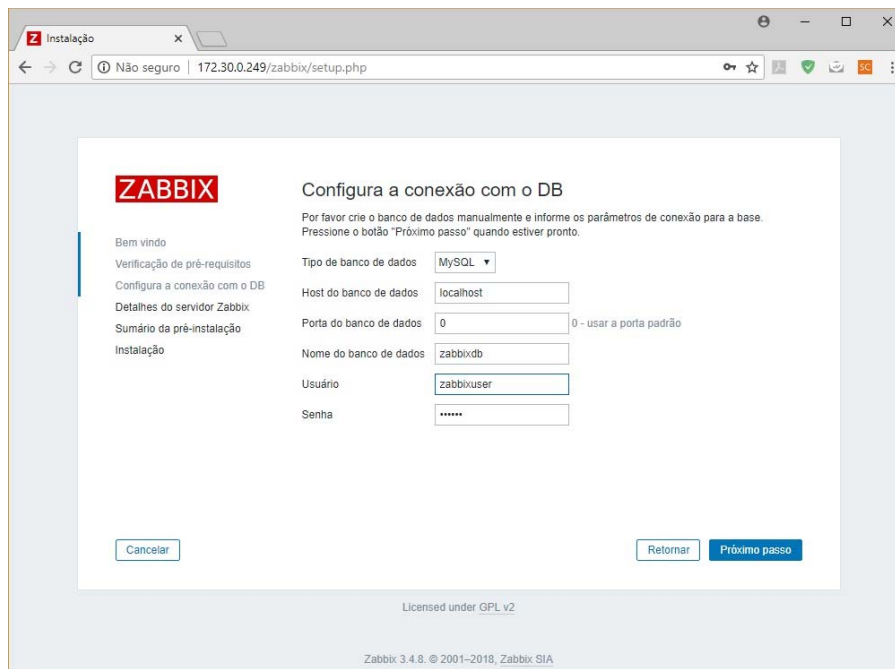
**Figura 20. Tela de Configuração do *Frontend* do Zabbix.**

A Figura 21 apresenta se todos os pré-requisitos de *software* foram atendidos.



**Figura 21. Tela de Configuração dos Pré-requisitos.**

A Figura 22 apresenta a tela de configuração para se conectar ao banco de dados MySQL que foi instalado, é necessário informar o nome da base de dados, usuário e senha. É importante lembrar que o banco de dados do *Zabbix* já deve estar criado.



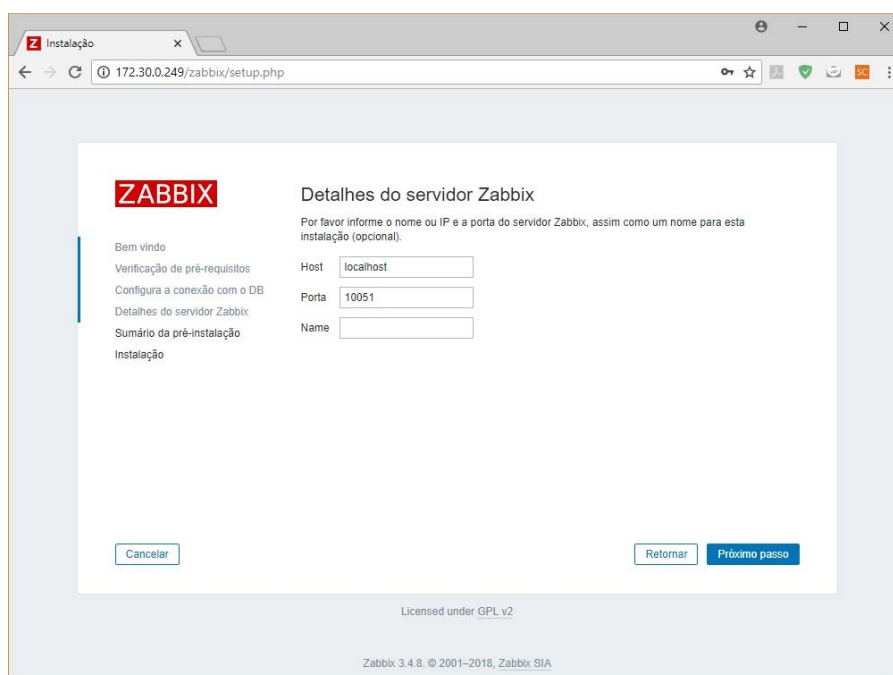
The screenshot shows a web browser window with the title "Instalação" and the address bar displaying "172.30.0.249/zabbix/setup.php". The page content is titled "ZABBIX" and "Configura a conexão com o DB". It includes a sidebar with navigation links: "Bem vindo", "Verificação de pré-requisitos", "Configura a conexão com o DB", "Detalhes do servidor Zabbix", "Sumário de pré-instalação", and "Instalação". The main content area contains the following fields and options:

- Tipo de banco de dados:
- Host do banco de dados:
- Porta do banco de dados:  0 - usar a porta padrão
- Nome do banco de dados:
- Usuário:
- Senha:

At the bottom of the form are three buttons: "Cancelar", "Retornar", and "Próximo passo". Below the form, it says "Licensed under GPL v2" and "Zabbix 3.4.8. © 2001-2018, Zabbix SIA".

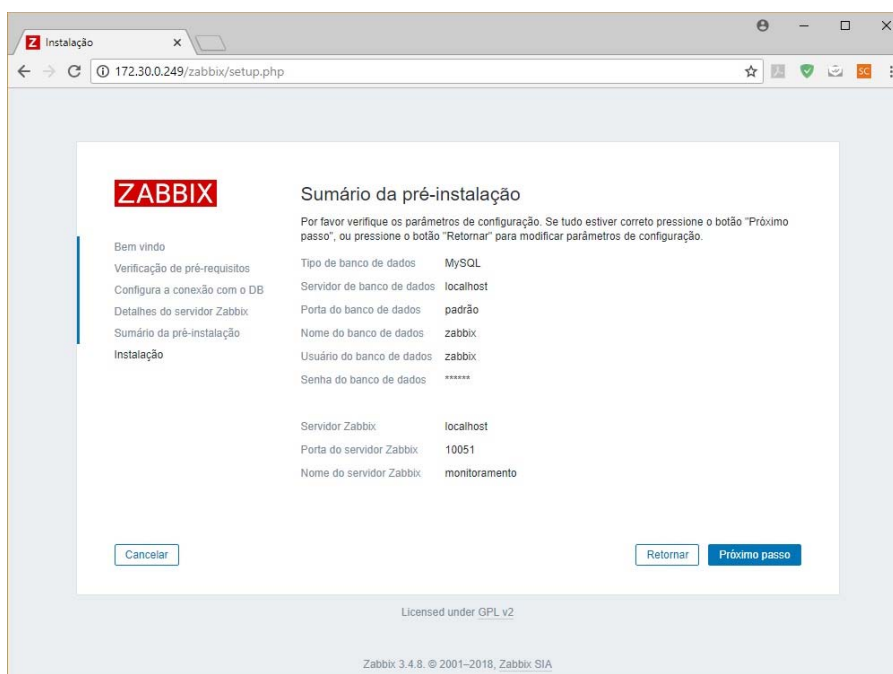
**Figura 22. Tela de Configuração do Banco de Dados.**

A Figura 23 apresenta os detalhes do servidor *Zabbix*. É necessário informar o nome do *Host* do servidor, porta de conexão utilizada e qual o nome será atribuído ao servidor *Zabbix*.



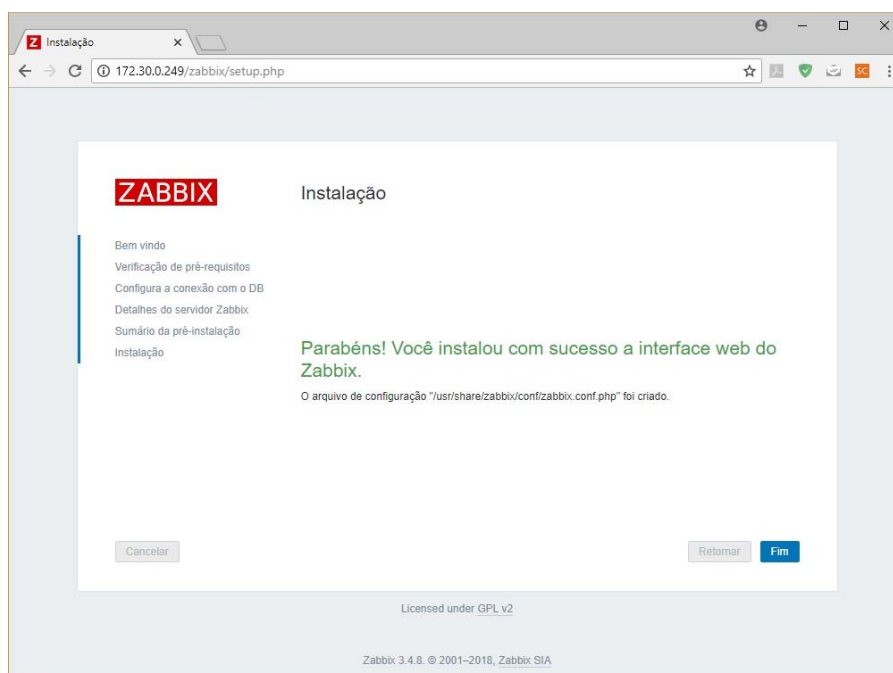
**Figura 23. Tela de Configuração da Porta de Conexão do Servidor Zabbix.**

A Figura 24 apresenta um resumo das configurações, caso seja necessário retorne e efetue as alterações necessárias.



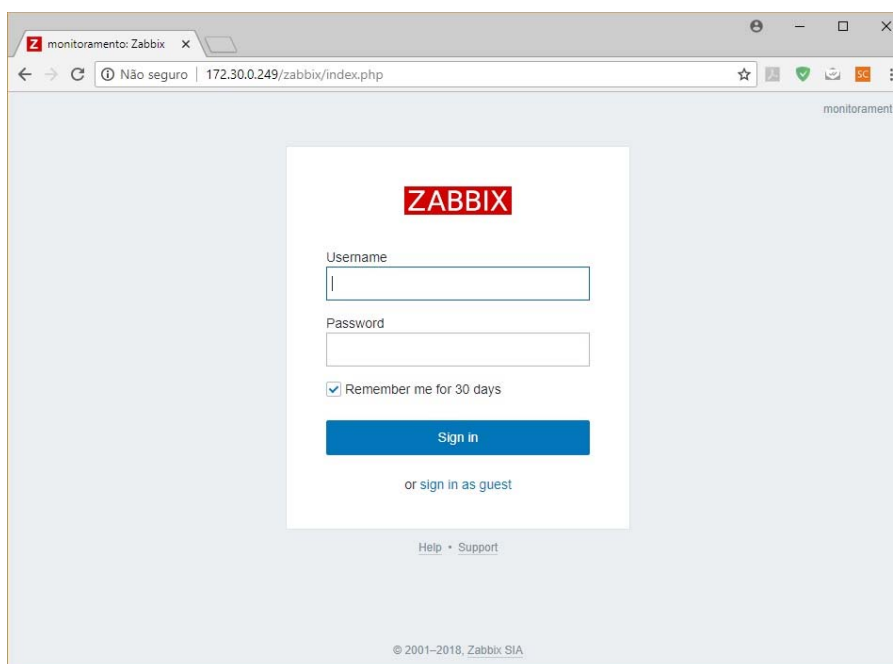
**Figura 24. Tela com o Resumo das Configurações.**

A Figura 25 apresenta a tela de conclusão da instalação, caso todas as etapas tenham ocorrido com sucesso.



**Figura 25. Conclusão da Instalação.**

A Figura 26 mostra o *frontend* do *Zabbix* pronto para uso. A nome de usuário padrão é “Admin” e a senha “zabbix”.



**Figura 26. Tela Inicial de Acesso ao Zabbix.**

## C.1. Configuração do agente *Zabbix*

Inicialmente é necessário verificar qual é a versão do sistema operacional da máquina na qual será instalado o agente *zabbix*. Essa informação é essencial para que não seja instalado uma versão incompatível com a versão em execução no servidor, para isso execute o seguinte comando.

```
#lsb_release -a
```

Ao obter a informação referente a versão do sistema operacional, baixe a versão do agente *zabbix* correspondente. Neste trabalho o sistema operacional utilizado corresponde ao Ubuntu 14.04 LTS. Desta forma, a primeira linha destaca o repositório para instalar o pacote necessário para o agente *zabbix* que é voltado a esta versão do sistema operacional. A segunda linha de código informa para instala o pacote que foi baixado e a terceira linha atualiza a base de dados do sistema operacional.

```
1 #wget http://repo.zabbix.com/zabbix/3.4/ubuntu/pool/main/z/zabbix-  
   release/zabbix-release_3.4-1+trusty_all.deb  
2 #dpkg -i zabbix-release_3.4-1+trusty_all.deb  
3 #sudo apt-get update
```

Após atualizar o repositório do sistema, é necessário instalar o agente *zabbix* com o seguinte comando.

```
#apt-get install zabbix-agent
```

Logo a seguir, é necessário alterar o arquivo de configuração do agente *zabbix* que estar localizado em: “/etc/zabbix/zabbix\_agentd.conf”. Altere e retire o comentário “#” quando necessário das seguintes linhas de comando:

```
1 Server=IP_do_SERVIDOR_ZABBIX  
2 ListenPort=10050  
3 ServerActive= IP_do_SERVIDOR_ZABBIX  
4 Hostname=NOME_da_ESTACAO_CLIENTE_LOCAL
```

Ao sair do arquivo de configuração do agente *zabbix*, inicie o serviço com o comando:

```
#/etc/init.d/zabbix-agent start
```

Para verificar o status do serviço do agente *zabbix*, utiliza a seguinte linha de comando.

```
#/etc/init.d/zabbix-agent status
```

A Figura 27, mostra o *status* do serviço do agente *zabbix*. Neste exemplo, o serviço estar em execução.

```
172.30.0.249 - PuTTY
root@Tdfinfoserv249:~# /etc/init.d/zabbix-agent status
● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2018-05-25 19:39:04 -03; 3 weeks 1 days ago
     Process: 24802 ExecStop=/bin/kill -SIGTERM $MAINPID (code=exited, status=0/SUCCESS)
    Process: 24806 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited, status=0/SUCCESS)
   Main PID: 24808 (zabbix_agentd)
      Tasks: 6 (limit: 4915)
   CGroup: /system.slice/zabbix-agent.service
           └─24808 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
             └─24809 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
               └─24810 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                 └─24811 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                   └─24812 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                     └─24813 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

mai 25 19:39:04 Tdfinfoserv249 systemd[1]: Starting Zabbix Agent...
mai 25 19:39:04 Tdfinfoserv249 systemd[1]: zabbix-agent.service: PID file /run/zabbix/zabbix_age...ctory
mai 25 19:39:04 Tdfinfoserv249 systemd[1]: Started Zabbix Agent.
Hint: Some lines were ellipsized, use -l to show in full.
root@Tdfinfoserv249:~#
```

Figura 27. Status do serviço do agente Zabbix.

Por fim, é necessário atribuir ao agente *zabbix* para iniciar automaticamente com sistema operacional ubuntu durante o processo de *boot* do sistema.

```
1 #update-rc.d zabbix-agent defaults
2 #update-rc.d zabbix-agent enable
```

O procedimento realizado nesta seção, deverá ser realizado para todas as máquinas a serem monitoradas pelo *zabbix*.

## C.2. Configuração das *Triggers* no Zabbix

Nesta seção é demonstrado como configurar uma *Trigger* para referenciar dados coletados, horário atual e outros fatores. Em outras palavras, uma *trigger* possibilita avaliar automaticamente os dados coletados com uma expressão que define o limite aceitável para o dado recebido. A Figura 28 mostra a tela para adicionar um novo *host* a ser monitorado, para isso, clique no menu “Configuração” e depois submenu “Hosts”.

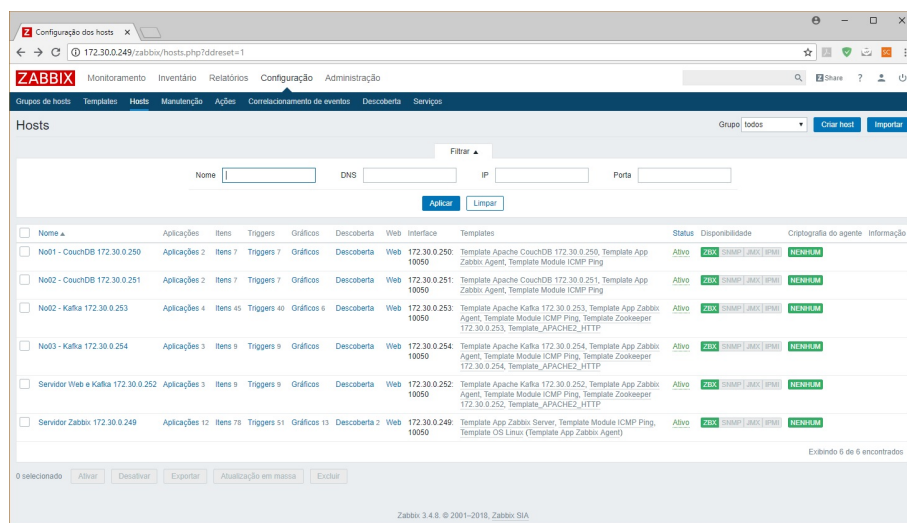
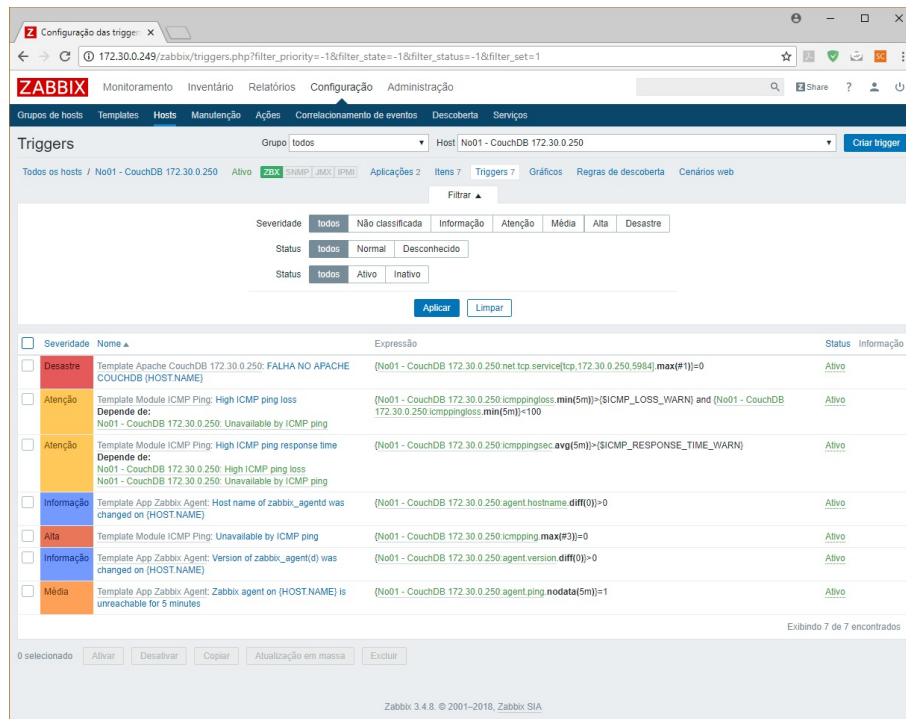


Figura 28. Tela de Configuração dos Hosts.

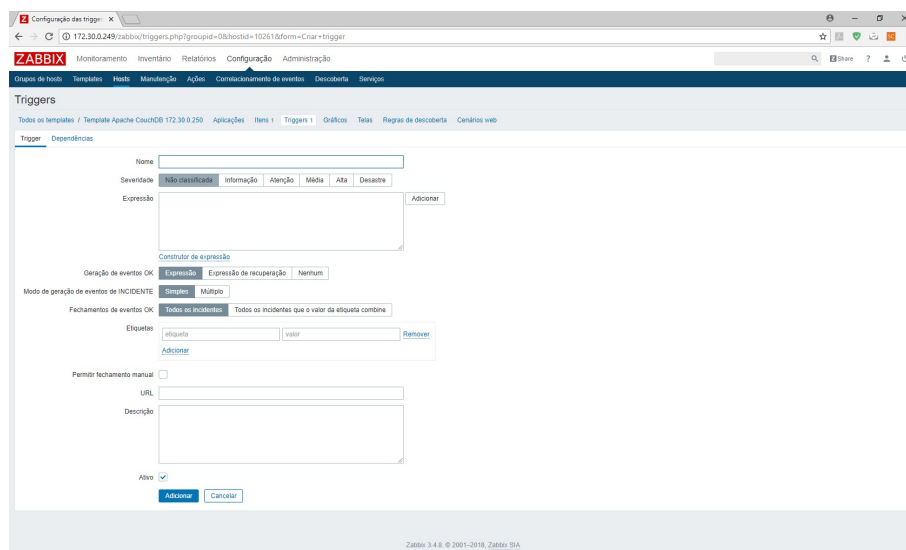


Ao ser mostrado os *hosts* já adicionados para monitoramento, clique no item “*Triggers*”correspondente ao *host* ao qual deseja-se criar/alterar uma *trigger*. A Figura 29 apresenta as 7 *triggers* existentes por grau de severidade para o *host* “No01 - CouchDB 172.30.0.250”. Para adicionar uma nova regra, clique no botão “Criar *trigger*”de cor azul com fonte branca, localizado no canto superior esquerdo.



**Figura 29. Tela de Configuração das *Triggers*.**

Na Figura 30, é apresentado o *template* com os campos a serem preenchidos para criar uma nova *trigger*.



**Figura 30. Criar uma Nova *Trigger*.**

Os seguintes campos foram preenchidos com os respectivos parâmetros:

- Nome: FALHA NO APACHE COUCHDB HOST.NAME
- Severidade: DESASTRE
- Expressão: Template Apache CouchDB 172.30.0.250:net.tcp.service[tcp,172.30.0.250,5984].max(1)=0
- Geração de eventos OK: Expressão
- Modo de geração de eventos de INCIDENTE: Simples
- Fechamentos de eventos OK: Todos os incidentes

As demais expressões que foram utilizadas para cada servidor são apresentadas na Tabela 2.

**Tabela 2. Triggers por Servidor**

Servidor	Nome	Expressão
No01 - CouchDB 172.30.0.250	FALHA NO APACHE COUCHDB {HOST.NAME}	{Template Apache CouchDB 172.30.0.250: net.tcp.service[tcp,172.30.0.250,5984].max(#1)}=0
No02 - CouchDB 172.30.0.251	FALHA NO APACHE COUCHDB {HOST.NAME}	{Template Apache CouchDB 172.30.0.251: net.tcp.service[tcp,172.30.0.251,5984].max(#1)}=0
Servidor Web e Kafka 172.30.0.252	FALHA NO APACHE2.HTTP {HOST.NAME}	{Template _APACHE2.HTTP:net.tcp.service[http].max(#1)}=0
	FALHA NO APACHE KAFKA {HOST.NAME}	{Template Apache Kafka 172.30.0.252:net.tcp.service[tcp,172.30.0.252,9092].max(#1)}=0
	FALHA NO ZOOKEEPER {HOST.NAME}	{Template Zookeeper 172.30.0.252:net.tcp.service[tcp,172.30.0.252,2181].max(#1)}=0
No02 - Kafka 172.30.0.253	FALHA NO APACHE2.HTTP {HOST.NAME}	{Template _APACHE2.HTTP:net.tcp.service[http].max(#1)}=0
	FALHA NO APACHE KAFKA {HOST.NAME}	{Template Apache Kafka 172.30.0.253:net.tcp.service[tcp,172.30.0.253,9093].max(#1)}=0
	FALHA NO ZOOKEEPER {HOST.NAME}	{Template Zookeeper 172.30.0.253:net.tcp.service[tcp,172.30.0.253,2181].max(#1)}=0
No03 - Kafka 172.30.0.254	FALHA NO APACHE2.HTTP {HOST.NAME}	{Template _APACHE2.HTTP:net.tcp.service[http].max(#1)}=0
	FALHA NO APACHE KAFKA {HOST.NAME}	{Template Apache Kafka 172.30.0.254:net.tcp.service[tcp,172.30.0.254,9094].max(#1)}=0
	FALHA NO ZOOKEEPER {HOST.NAME}	{Template Zookeeper 172.30.0.254:net.tcp.service[tcp,172.30.0.254,2181].max(#1)}=0

## D. Instalação e Configuração do *Gatling*

*Gatling* é uma ferramenta de teste de carga e estresse baseada no Scala<sup>3</sup> e construída para alto desempenho. Antes de iniciar com a instalação do *gatling*, efetue a instalação do JAVA. Para instalar o JAVA execute o comando:

```
#apt-get install default-jdk
```

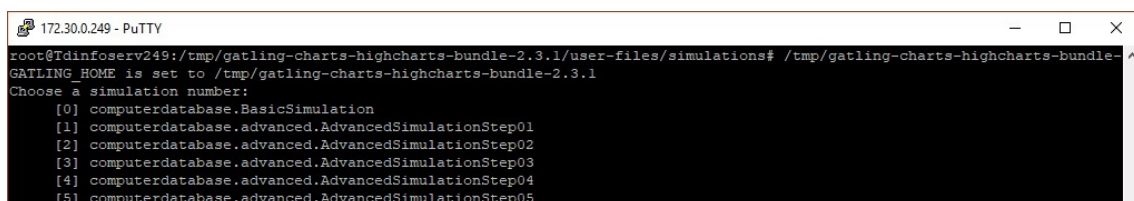
Após a instalação do JAVA, acesse o diretório “/tmp”, baixe o *gatling* versão 2.3.1, descompacte o arquivo no mesmo diretório “/tmp”. Acesse o diretório “gatling-charts-highcharts-bundle-2.3.1-bundle” que foi criado ao descompactar o arquivo “.zip”, acesse o diretório “/bin” e execute o arquivo “gatling.sh”, para executá-lo em modo *shell* (Figura 31).

<sup>3</sup>Scala é uma linguagem de programação de propósito geral projetada para expressar padrões de programação comuns de uma forma concisa, elegante e de tipagem segura (*type-safe*).

```

1 #cd /tmp/
2 #wget https://repo1.maven.org/maven2/io/gatling/highcharts/gatling-
   charts-highcharts-bundle/2.3.1/gatling-charts-highcharts-bundle
   -2.3.1-bundle.zip
3 #unzip gatling-charts-highcharts-bundle-2.3.1-bundle.zip
4 #cd gatling-charts-highcharts-bundle-2.3.1-bundle
5 #cd bin/
6 #./gatling.sh

```



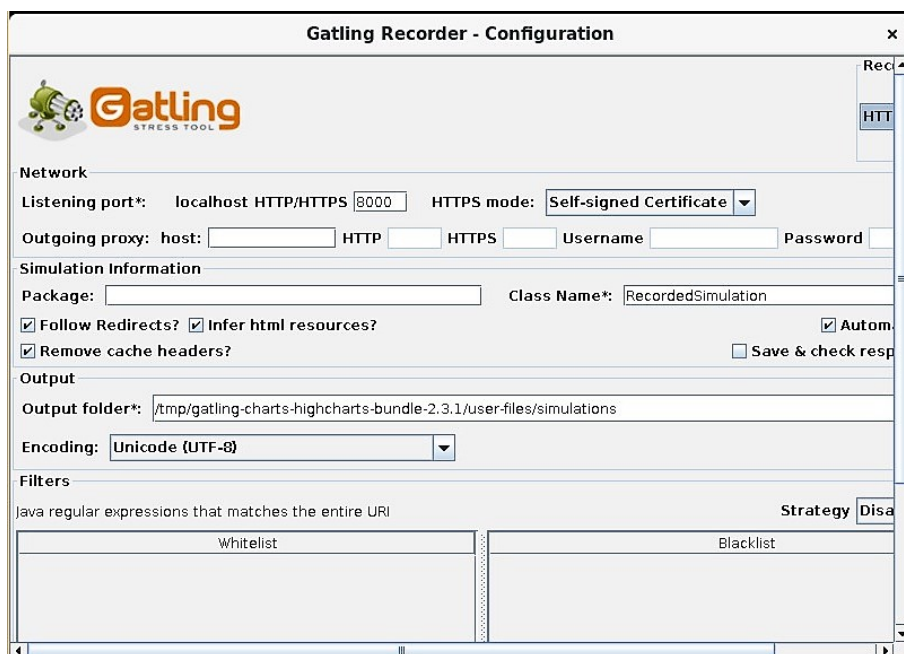
```

172.30.0.249 - PuTTY
root@Tdfoserv249:/tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations# ./tmp/gatling-charts-highcharts-bundle-
GATLING_HOME is set to /tmp/gatling-charts-highcharts-bundle-2.3.1
Choose a simulation number:
[0] computerdatabase.BasicSimulation
[1] computerdatabase.advanced.AdvancedSimulationStep01
[2] computerdatabase.advanced.AdvancedSimulationStep02
[3] computerdatabase.advanced.AdvancedSimulationStep03
[4] computerdatabase.advanced.AdvancedSimulationStep04
[5] computerdatabase.advanced.AdvancedSimulationStep05

```

**Figura 31. Tela de *Shell* do *Gatling*.**

A Figura 31 apresenta em tela em *Shell* do *gatling*. Para o *gatling* com o *frontend* em JAVA, execute o arquivo “recorder.sh”, conforme apresenta a Figura 32.



**Gatling Recorder - Configuration**

**Network**

Listening port\*: localhost HTTP/HTTPS 8000 HTTPS mode: Self-signed Certificate

Outgoing proxy: host: HTTP HTTPS Username Password

**Simulation Information**

Package: Class Name\*: RecordedSimulation

☒ Follow Redirects? ☒ Infer html resources? ☒ Autom.

☒ Remove cache headers? ☐ Save & check resp

**Output**

Output folder\*: /tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations

Encoding: Unicode (UTF-8)

**Filters**

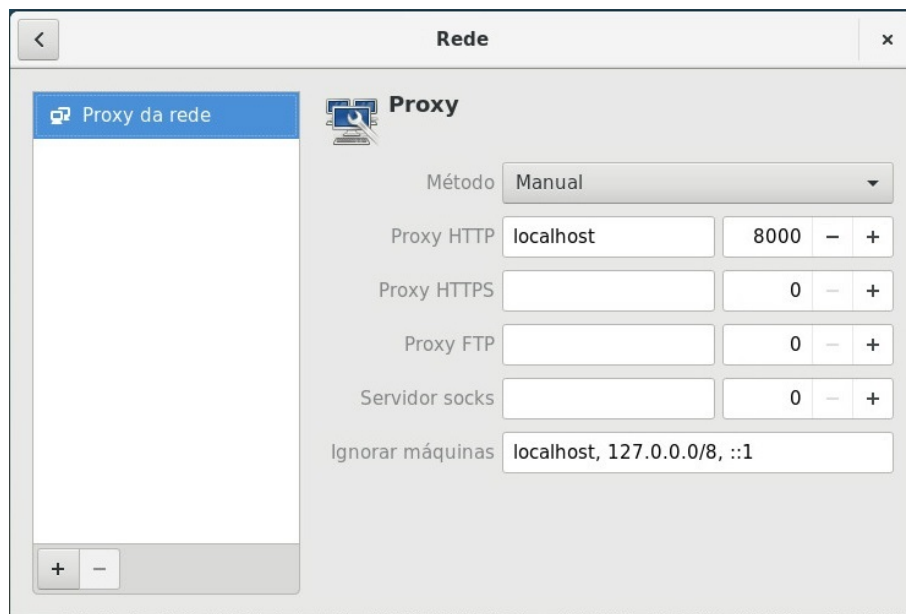
Java regular expressions that matches the entire URI

Whitelist Blacklist

Strategy Disa

**Figura 32. Tela de *Frontend* do *Gatling*.**

Com a instalação do serviço já realizada, o próximo passo é configurar o *gatling* para realização dos experimentos. Para isso é necessário configurar o navegador *Web* para redirecionar sua comunicação via *proxy* com uma porta de comunicação específica para captura do dados pelo *gatling*. A Figura 33 mostra que o endereço de *proxy* informado foi *localhost* e a porta utilizada foi a 8000. Independentemente do navegador *Web* utilizado, o *gatling* vai conseguir capturar os dados recebidos.



**Figura 33. Tela de configuração do *Proxy*.**

A Figura 34 apresenta a tela com as configurações básicas e essenciais para o funcionamento do *gatling*. Nesta janela altere as configurações:

- Recorder mode: selecione *HTTP Proxy*;
- Digite o número da porta na caixa do *host* local (neste trabalho foi utiliza a porta 8000 ).
- Digite o nome do pacote a ser criado: distribuídos.

**Observação!** Escolha o nome que preferir.

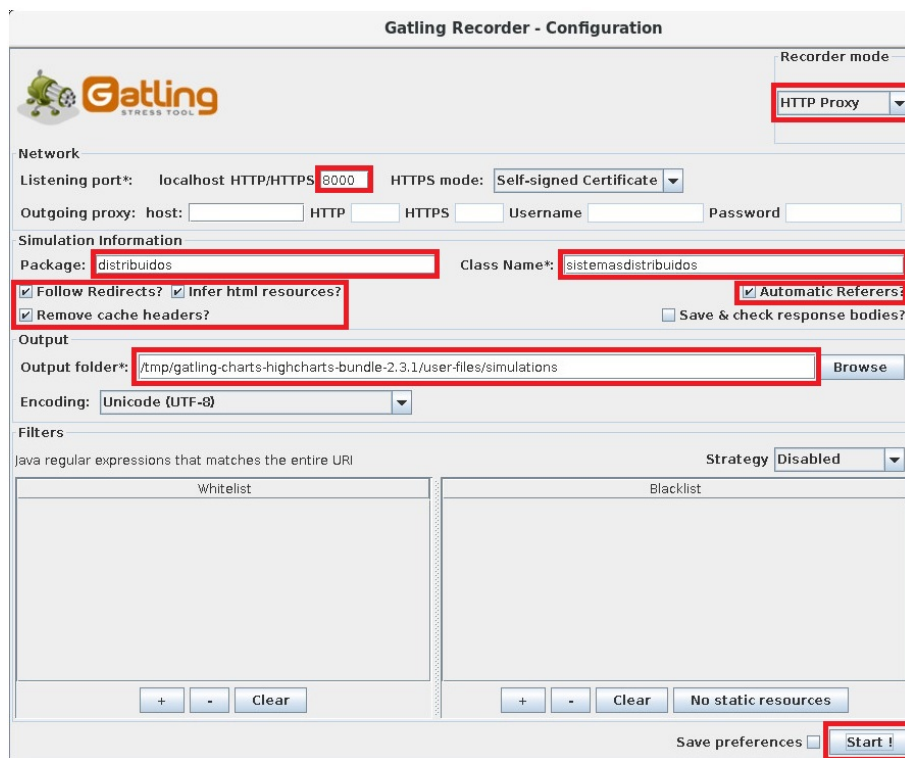
- Digite o nome da classe a ser criada para o pacote: sistemasdistribuidos

**Observação!** Escolha o nome que preferir.

- Habilite as opções: Follow Redirects?; Infer html resources?; Remove cache headers? e Automatic Referers?
- Informe o caminho dos arquivos de saída. neste trabalho foi utilizado: /tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations

**Observação!** Este caminho é onde o *script* será armazenado após a gravação. Você pode alterar o local, mas se fizer isso, não se esqueça de copiá-lo para a pasta de simulações do *Gatling*. Para facilitar as coisas, é recomendável deixar o caminho como padrão.

- Clique no botão Start.



**Gatling Recorder - Configuration**

Recorder mode: HTTP Proxy

Network  
 Listening port\*: localhost HTTP/HTTPS 8000 HTTPS mode: Self-signed Certificate  
 Outgoing proxy: host: HTTP HTTPS Username Password

Simulation Information  
 Package: distribuidos Class Name\*: sistemasdistribuidos  
☒ Follow Redirects? ☒ Infer html resources? ☒ Automatic Referers  
☒ Remove cache headers? ☐ Save & check response bodies?

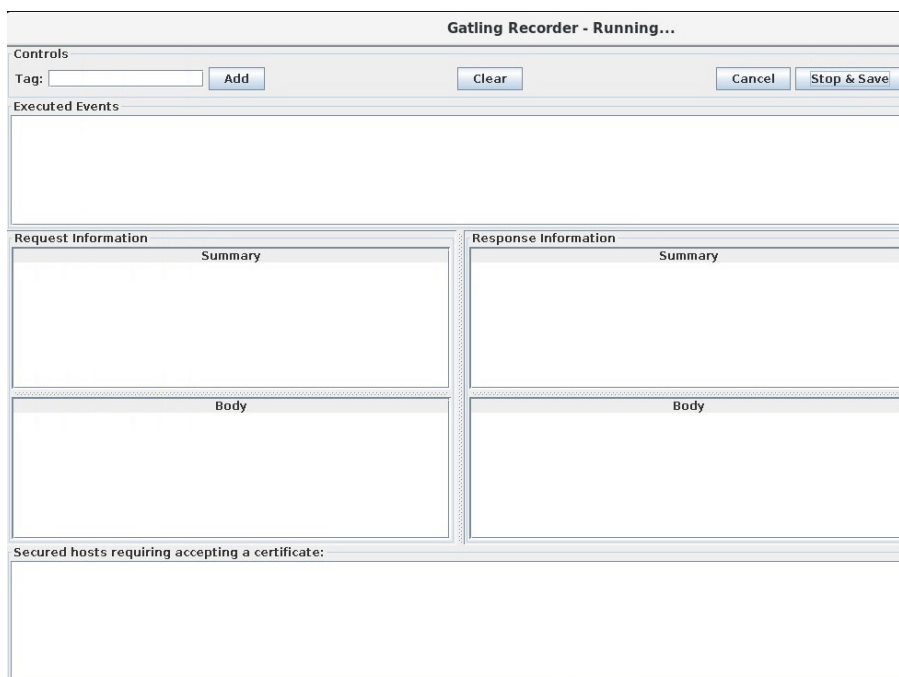
Output  
 Output folder\*: /tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations Browse  
 Encoding: Unicode (UTF-8)

Filters  
 java regular expressions that matches the entire URI Strategy Disabled  
 Whitelist Blacklist  
 + - Clear + - Clear No static resources

Save preferences ☐ Start !

**Figura 34. Configuração do *Frontend* do Gatling.**

Ao iniciar a captura dos dados pelo *frontend* do *gatling* (Figura 34), uma nova janela será exibida (Figura 35) referente a captura dos dados a serem obtidos do *link* do *site* a ser acessado no navegador *Web*.



**Gatling Recorder - Running...**

Controls  
 Tag: Add Clear Cancel Stop & Save

Executed Events

Request Information	Response Information
Summary	Summary
Body	Body

Secured hosts requiring accepting a certificate:

**Figura 35. Tela de Captura dos Dados.**

36). Retorne ao navegador *Web* e acesse o *site* ao qual queira capturar os dados (Figura

	CNPJ	Razão Social	Nome do Medicamento	Número Registro(A)
1	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130040
2	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130040
3	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130024
4	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130067
5	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130105
6	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130172
7	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130040
8	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130024
9	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130105
10	02685377000157	SANOFI-AVENTIS FARMACÊUTICA LTDA	CEWIN	1130010130172
11	49324221000104	FRESENIUS KABI BRASIL LTDA	1021 CHD C/3,5 MEQ/L CA NA 136	1134400240011
12	07643687000122	ENDOMED LABORATORIO FARMACEUTICO LT...	1021 CHD C/3,5 MEQ/L CA NA 136	SI/NC

Figura 36. Site do Sistema de Saúde.

Automaticamente na tela de captura de dados (Figura 37) serão apresentadas as informações capturadas ao acessar o respectivo *site*. Feche o navegador, clique no botão “Stop & Save” para finalizar a captura dos dados e salvar o que foi obtido. A janela *Gatling Recorder* será fechada.



Figura 37. Captura dos Dados pelo Gatling.

A Figura 38 mostra um arquivo de registrador chamado “sistemasdistribuidos.scala”foi criado e salvo no diretório “/tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations/distribuidos”. Este *script* scala é criado automaticamente ao término da gravação. O nome do arquivo é o nome da classe com a extensão “.scala”e está localizado em uma pasta chamada “distribuidos” que é o mesmo nome do pacote que foi inserido na janela do registrador. Essa estrutura de pastas será seguida para testes de carga.

```
172.30.0.249 - PuTTY
root@Tdfinfoserv249:/tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations/distribuidos# ls -las
total 12
4 drwxr-xr-x 2 root root 4096 jun 27 17:43 .
4 drwxr-xr-x 5 root root 4096 jun 27 17:09 ..
4 -rw-r--r-- 1 root root 1535 jun 27 17:43 sistemasdistribuidos.scala
root@Tdfinfoserv249:/tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations/distribuidos#
```

**Figura 38. Arquivo com Extensão Scala Criado.**

O arquivo de *script* estar escrito na linguagem de programação Scala. É neste arquivo que tem-se as configurações referente ao número de usuários e como eles são executados no *script*. Também é possível adicionar uma pausa global, configurar protocolos HTTP ou HTTPS por meio desse *script*. O método `setUp()` é usado para configurar o usuário. Neste exemplo, ele está configurado para 1 usuário e usa o protocolo HTTP para teste de carga: `setUp(scenario.inject(atOnceUsers(1)))`. Protocolos (`httpProtocol`). Para realização dos testes de carga é necessário alterar neste arquivo o número de usuários, foi definido um quantitativo de 250 usuários para a realização dos testes de carga neste trabalho.

```
1 package distribuidos
2
3 import scala.concurrent.duration._
4
5 import io.gatling.core.Predef._
6 import io.gatling.http.Predef._
7 import io.gatling.jdbc.Predef._
8
9 class sistemasdistribuidos extends Simulation {
10
11     val httpProtocol = http
12         .baseUrl("http://172.30.0.252")
13         .inferHtmlResources()
14         .acceptHeader("*/")
15         .acceptEncodingHeader("gzip, deflate")
16         .acceptLanguageHeader("pt-BR, pt;q=0.9, en-US;q=0.8, en;q=0.7")
17         .userAgentHeader("Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36")
18
19     val headers_0 = Map(
20         "Accept" -> "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8",
21         "Upgrade-Insecure-Requests" -> "1")
22
23     val headers_3 = Map("X-Requested-With" -> "XMLHttpRequest")
24
25     val url = "http://172.30.0.252/public"
26
27     val scn = scenario("sistemasdistribuidos")
28         .exec(http("request_0")
29             .get("/public/"))
```



```

30         .headers(headers_0))
31     .pause(1)
32     .exec(http("request_1")
33         .get("/public/")
34         .headers(headers_0))
35     .pause(2)
36     .exec(http("request_2")
37         .get("/public/src//form/ListarMedicamentos.js?_dc=1530130603692")
38         .resources(http("request_3")
39             .get("/public/medicamento/?m=getTodosMedicamentos&_dc=1530130603708&page=1&start=0&limit=200&sort=NO_PRODUTO&dir=ASC")
40             .headers(headers_3) ,
41             http("request_4")
42                 .get("/public/medicamento/?m=getTodosMedicamentos&_dc=1530130603839&page=1&start=0&limit=200&sort=NO_PRODUTO&dir=ASC")
43                 .headers(headers_3)))
44
45     setUp(scen.inject(atOnceUsers(1))).protocols(httpProtocol)
46 }

```

Para executar um teste de carga com o *gatling* acesse o diretório “/tmp/gatling-charts-highcharts-bundle-2.3.1/bin”, depois execute o arquivo “gatling.sh”. Este é o arquivo de execução do *Gatling*, que executa o teste, neste caso “sistemasdistribuidos.scala”.

```

1 #cd /tmp/gatling-charts-highcharts-bundle-2.3.1-bundle/bin/
2 #./gatling.sh

```

```

172.30.0.249 - PuTTY
root@Tdfinfoserv249: /tmp/gatling-charts-highcharts-bundle-2.3.1/user-files/simulations# /tmp/gatling-charts-highcharts-bundle-2.3.1
GATLING_HOME is set to /tmp/gatling-charts-highcharts-bundle-2.3.1
Choose a simulation number:
[0] computerdatabase.BasicSimulation
[1] computerdatabase.advanced.AdvancedSimulationStep01
[2] computerdatabase.advanced.AdvancedSimulationStep02
[3] computerdatabase.advanced.AdvancedSimulationStep03
[4] computerdatabase.advanced.AdvancedSimulationStep04
[5] computerdatabase.advanced.AdvancedSimulationStep05
[6] distribuidos.sistemasdistribuidos
[7] sistemasdistribuidos.RecordedSimulation

```

**Figura 39. Tela de *Shell* do *Gatling*.**

A Figura 39 mostra um *shell* de comandos que é apresentado, em seguida as informações de teste de carregamento são exibidas na janela: variável *GATLING\_HOME*, arquivo de execução Java, e os diretórios com os *Scripts in simulations*.



```
root@Tdinfoserv249:/tmp/gatling-charts-highcharts-bundle-2.3.1/bin# ./gatling.sh
GATLING_HOME is set to /tmp/gatling-charts-highcharts-bundle-2.3.1
Choose a simulation number:
  [0] computerdatabase.BasicSimulation
  [1] computerdatabase.advanced.AdvancedSimulationStep01
  [2] computerdatabase.advanced.AdvancedSimulationStep02
  [3] computerdatabase.advanced.AdvancedSimulationStep03
  [4] computerdatabase.advanced.AdvancedSimulationStep04
  [5] computerdatabase.advanced.AdvancedSimulationStep05
  [6] distribuidos.sistemasdistribuidos
  [7] sistemasdistribuidos.RecordedSimulation
6
Select simulation id (default is 'sistemasdistribuidos'). Accepted characters are a-z, A-Z, 0-9, - and _
sistemasdistribuidos
Select run description (optional)

Simulation distribuidos.sistemasdistribuidos started...

=====
2018-06-29 10:41:23                               5s elapsed
----- Requests -----
> Global (OK=22 KO=0 )
> request_0 (OK=11 KO=0 )
> ext-all.css (OK=11 KO=0 )
----- sistemasdistribuidos -----
```

Figura 40. Tela de Carga com o Gatling.

A Figura 40 mostra que foi selecionado a opção 6 que corresponde ao arquivo “sistemasdistribuidos” e pressione a tecla “Enter”. Depois é informado o ID de simulação padrão “sistemasdistribuidos”, insira este nome e pressione a tecla “Enter”. Será solicitado uma descrição de execução opcional, pressione a tecla “Enter” para prosseguir e a simulação será iniciada.

**Observação!** Na maioria dos casos, a descrição da execução será o objetivo ou título dos casos de teste e também o número de usuários e a descrição da carga. A descrição destina-se a explicar o objetivo do teste de carga para qualquer pessoa que esteja lendo esta descrição. Este é um passo opcional, para pular pressionando “Enter”.

Quando a simulação estiver concluída, será apresentado um *link* que gera o relatório no formato HTML, conforme apresentado na Figura 41.

```
-----
--- Global Information ---
> request count 1910 (OK=1742 KO=168 )
> min response time 0 (OK=0 KO=0 )
> max response time 59869 (OK=59869 KO=0 )
> mean response time 9043 (OK=9915 KO=0 )
> std deviation 10799 (OK=10919 KO=0 )
> response time 50th percentile 5724 (OK=7918 KO=0 )
> response time 75th percentile 15206 (OK=15507 KO=0 )
> response time 95th percentile 32370 (OK=33184 KO=0 )
> response time 99th percentile 52118 (OK=53647 KO=0 )
> mean requests/sec 23.012 (OK=20.988 KO=2.024 )
--- Response Time Distribution ---
> t < 800 ms 598 ( 31%)
> 800 ms < t < 1200 ms 2 ( 0%)
> t > 1200 ms 1142 ( 60%)
> failed 168 ( 9%)
--- Errors ---
> j.n.ConnectException: connection timed out: /172.30.0.252:80 149 (88,69%)
> j.u.c.TimeoutException: Request timeout to /172.30.0.252:80 af 19 (11,31%)
ter 60000 ms
-----

Reports generated in 0s.
Please open the following file: /tmp/gatling-charts-highcharts-bundle-2.3.1/results/sistemasdistribuidos-1530279677947/index.html
root@Tdinfoserv249:/tmp/gatling-charts-highcharts-bundle-2.3.1/bin#
```

Figura 41. Local do Relatório.

O arquivo gerado pode ser aberto em qualquer navegador *Web*.