

# Infraestrutura

- Pré-Requisitos de Instalação
- Instalação do Sistema
- Instalação do Sistema

# Pré-Requisitos de Instalação

# Pré-Requisitos de Instalação

É recomendável que a equipe técnica responsável pela implantação no sistema Polare na instituição tenha alguma experiência com as tecnologias a seguir.

**Java.** O sistema Polare está sendo escrito em [Java](#) e portanto alguma experiência com essa linguagem pode ser útil, mas não estritamente necessário para implantação do sistema.

**Maven.** O sistema Polare utiliza uma ferramenta construção de projeto e gerenciamento de dependências chamada [Maven](#). Não é necessário ter conhecimento aprofundado sobre essa ferramenta, visto que para gerar os artefatos `.war` do sistema Polare é preciso apenas ter o Maven instalado e executar comandos pré-definidos no *shell* do sistema.

**Git.** Git é um sistema de versionamento (controle de versão) distribuído. Necessário para obtenção do código fonte do sistema Polare. Não é necessário ter conhecimento aprofundado sobre essa ferramenta, visto que para obter o código fonte do sistema Polare é preciso apenas ter o Git instalado e executar comandos pré-definidos no *shell* do sistema.

**Web.** Pode ser necessário editar algumas páginas *web* do sistema Polare de forma a customizar a identidade visual da instituição. O sistema Polare utiliza o *framework web* [Thymeleaf](#), mas algum conhecimento básico de html, css e javascript deve ser suficiente para efetuar alterações em páginas *web* do sistema.

**Linux.** Experiência fundamental. Nessa implantação utilizamos o [Ubuntu](#) linux por conveniência. Outras *distros* linux podem ser utilizadas de acordo com a necessidade da instituição.

**Docker.** Nesta implantação optamos por utilizar [Docker](#), embora cada instituição possa definir como o sistema Polare deverá ser implantado internamente. Entretanto, recomendamos Docker como uma forma moderna e robusta de executar esse tipo de implantação.

**Postgresql.** Experiência com bancos de dados é importante em virtualmente qualquer sistema moderno de *software*. Não é estritamente necessária para a implantação do sistema Polare, mas é útil para entender e resolver certos problemas relacionados com migrações e *schemas*, bem como configuração de usuários e acesso ao banco de dados.

# Software e Hardware necessários

**Infraestrutura de API.** É necessário instalar uma infraestrutura de que permita acessar a API de sistemas. Essa API de sistemas é responsável por fornecer ao sistema Polare informações do usuário servidor, como por exemplo, lotação e hierarquia funcional. Tecnicamente, essa infraestrutura é um conjunto de sistemas a parte que fornece informações armazenadas nos bancos de dados do SIG via API [Restful](#).

A UFRN pode disponibilizar um ambiente contendo essa infraestrutura. Acesse a documentação oficial (é necessário login) para mais detalhes:

[https://docs.info.ufrn.br/doku.php?id=cooperacao:tutoriais:infraestrutura:api:montagem\\_infraestrutura\\_api](https://docs.info.ufrn.br/doku.php?id=cooperacao:tutoriais:infraestrutura:api:montagem_infraestrutura_api).

images/visao-geral-ambiente-api.png

Visão geral do ambiente

## Note

A infraestrutura de API utilizada pelo Polare no IFPA foi instalada anteriormente para ser utilizada pelo SIGAA Mobile. Nesse caso a infraestrutura de API foi reutilizada.

## Warning

Alguns erros de API no sistema Polare podem ser corrigidos atualizando as imagens Docker de uma infraestrutura de API previamente instalada. O problema decorre da defasagem entre a versão da infraestrutura de API instalada e o Polare.

**Maven.** Instalar o Maven. Basicamente fazer o download da ferramenta e acessar os executáveis contidos no diretório bin via linha de comando. Para mais detalhes acesse

<https://maven.apache.org/download.cgi>

**Git.** Instalar o Git. Para mais detalhes acesse <https://git-scm.com/download/linux>

**Servidor Linux.** Instalar um servidor Linux com seguintes configurações:

- 4 GB de memória Ram
- 100 GB de armazenamento
- 4 núcleos de processador

- Ubuntu 22.04.1 LTS

**Docker e Docker-compose.** Instalar no servidor Linux as versões mais atuais do [Docker](#) e [Docker-compose](#).

[Previous](#)[Next](#)

---

Sponsored by [EthicalAds](#)  
Image source and link type are unknown

**Reach specific developers** on the open source, privacy-first ad network: **EthicalAds**

Ad by [EthicalAds](#) · [i](#)

© Copyright 2023, UFRN/IFPA. Revision `ce91cf46`.

Built with [Sphinx](#) using a [theme](#) provided by [Read the Docs](#).

# Instalação do Sistema

- - Instalação do Sistema
  - [Edit on GitHub](#)
- 

# Instalação do Sistema

O sistema Polare e o banco de dados [Postgresql](#) serão executados em contêiner (Docker) conforme descrito em [Pré-Requisitos de Instalação](#).

No momento que esta documentação é escrita, o sistema Polare precisa ser distribuído em um servidor Java *web*. Neste guia utilizaremos o [Tomcat](#) versão 9.

O procedimento de implantação do sistema Polare consiste em criar uma imagem Tomcat 9 e em seguida executar essa imagem via Docker. Essa imagem deverá conter as configurações locais da instituição (conexões com o banco de dados, senhas, mapeamento das rotas da infraestrutura de API, etc.) e o sistema Polare (arquivos `.war` construídos utilizando a ferramenta Maven a partir do código fonte).

## Construção dos artefatos .war (Polare)

O procedimento para gerar os arquivos `.war` consiste em clonar o código fonte do sistema Polare via Git e executar o Maven para construir esses arquivos `.war`.

**Clonar o código fonte.** O seguinte comando deve ser executado a partir de um diretório na linha de comando para clonar o código fonte do sistema Polare (é necessário informar as suas credenciais de acesso):

```
git clone https://gitcooperacao.info.ufrn.br/referencia/polare.git
cd polare
git checkout release_0.5.0 -b release_0.5.0
```

### ⚠ Warning

O comando `git checkout release_0.5.0 -b release_0.5.0` faz *checkout* na tag `release_0.5.0` e imediatamente cria uma *branch* com o mesmo nome.

### Warning

O comando `git checkout release_0.5.0 -b release_0.5.0` faz *checkout* na tag `release_0.5.0` e imediatamente cria uma *branch* com o mesmo nome.

**Construir os arquivos .war utilizando Maven.** O seguinte comando deve ser executado a partir do diretório polare (que contém o código fonte do sistema):

```
./mvnw clean install -Pwar -B -Dmaven.test.skip=true -Dmaven.javadoc.skip=true -Ddependency-check.skip=true
```

A ferramenta Maven fará o *download* de todas as dependências necessárias para o projeto finalizando com a construção dos arquivos `.war`.

Caso ocorra o erro:

```
Error: Could not find or load main class org.apache.maven.wrapper.MavenWrapperMain
Caused by: java.lang.ClassNotFoundException: org.apache.maven.wrapper.MavenWrapperMain
```

É necessário executar o comando seguinte para configurar o `maven-wrapper`:

```
mvn -N io.takari:maven:wrapper
```

Dois arquivos `.war` serão gerados:

1. polare/polare-frontend-publico/target/polare-frontend-publico.war
2. polare/polare-frontend-restrito/target/polare-frontend-restrito.war

# Criação da Imagem Tomcat

Crie um diretório contendo os seguintes arquivos:

1. apache-tomcat-9.0.65 (diretório)
2. copiar-ears-e-iniciar-tomcat.sh
3. Dockerfile

O conteúdo desses arquivos é descrito a seguir.

### apache-tomcat-9.0.65

Esse é o diretório contendo o Tomcat 9 descompactado que pode ser obtido no link <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.65/bin/apache-tomcat-9.0.65.zip>

### copiar-ears-e-iniciar-tomcat.sh

```
#!/bin/sh
cp /polare/deploy/* /opt/tomcat/webapps/
/opt/tomcat/bin/catalina.sh run
```

### Dockerfile

```
FROM openjdk:17-alpine

RUN mkdir -p /opt/tomcat/
RUN mkdir -p /polare/deploy

COPY ./apache-tomcat-9.0.65 /opt/tomcat
COPY ./copiar-ears-e-iniciar-tomcat.sh /polare/

EXPOSE 8080 8443 8009 9999 8787
env CATALINA_HOME /opt/tomcat

ENTRYPOINT ["/polare/copiar-ears-e-iniciar-tomcat.sh"]
```

Os arquivos 1 e 2 serão inseridos na imagem que será criada através do Dockerfile. Neste guia o nome da imagem Tomcat será tomcat-9. Para criar a imagem execute o comando a seguir a partir do diretório contendo os arquivos descritos anteriormente:

```
docker build -t tomcat-9 .
```

#### Note

O nome da imagem (tomcat-9) é referenciado no arquivo docker-compose.yml

# Execução do Ambiente

Crie um diretório contendo os seguintes arquivos:

1. wars (diretório)
2. docker-compose.yml
3. nginx.conf
4. catalina.sh

O conteúdo desses arquivos é descrito a seguir.

**wars (diretório).** É necessário copiar os arquivos `.war` gerados anteriormente para este diretório. Nesta implantação os arquivos `.war` foram renomeados para `polare.war` (referente ao arquivo `polare-frontend-restrito.war`) e `polare-publico.war` (referente ao arquivo `polare-frontend-publico.war`).

## Warning

O Tomcat atribui o caminho do contexto da aplicação em função do nome do arquivo `.war` contido no diretório `apache-tomcat-9.0.65/webapps/` por padrão. Por exemplo, um arquivo (aplicação) chamado `polare.war`, pode ser acessado via <http://localhost:8080/polare>.

**docker-compose.yml.** É necessário fazer ajustes nesse arquivo em função das configurações do ambiente local (as linhas com o comentário `# ALTERAR` devem ser modificadas).

```
version: "3.2"
services:
  nginx:
    image: nginx
    hostname: nginx
    restart: unless-stopped
    ports:
      - "80:80"
      - "443:443"
    environment:
      - TZ=America/Belem # ALTERAR
    volumes:
      - type: bind
        source: ./nginx.conf
        target: /etc/nginx/conf.d/default.conf
```



```

    read_only: true
networks:
  - rede
depends_on:
  - tomcat9
polare-db:
  container_name: polare-db
  image: postgres:12
  hostname: polare-db
  ports:
    - "5432:5432"
  environment:
    - POSTGRES_USER=postgres
    - POSTGRES_PASSWORD=postgres
  healthcheck:
    test: [ "CMD-SHELL", "pg_isready -U postgres" ]
    interval: 10s
    timeout: 5s
    retries: 5
  restart: unless-stopped
  volumes:
    - /home/administrador/polare-db:/var/lib/postgresql/data
  networks:
    - rede
tomcat9:
  image: tomcat:9
  hostname: tomcat9
  restart: unless-stopped
  ports:
    - "8080:8080"
  volumes:
    - type: bind
      source: ./catalina.sh
      target: /opt/tomcat/bin/catalina.sh
      read_only: true
    - type: bind
      source: ${WAR_POLARE}/polare.war # WAR_POLARE é o caminho da pasta "wars"
      target: /polare/deploy/polare.war
      read_only: true
    - type: bind
      source: ${WAR_POLARE}/polare-publico.war # WAR_POLARE é o caminho da pasta "wars"
      target: /polare/deploy/polare-publico.war
      read_only: true
  networks:
    - rede
networks:
  rede:
    driver: bridge

```

## ⚠ Warning

É necessário fazer a configuração SSL no nginx para distribuir a aplicação via HTTPS.

**nginx.conf** Configuração do nginx. É necessário fazer ajustes nesse arquivo em função das configurações do ambiente local (as linhas com o comentário `# ALTERAR` devem ser modificadas).

```
proxy_cache_path /tmp/NGINX_treinamento_cache/ keys_zone=backcache:10m; # ALTERAR

upstream polare {
    ip_hash;
    server tomcat9:8080;
}

server {
    listen 80;
    server_name polare-treinamento.ifpa.edu.br; # ALTERAR

    client_max_body_size 128M;

    access_log /var/log/nginx/polare-treinamento.ifpa.edu.br-80-access.log; # ALTERAR
    error_log /var/log/nginx/polare-treinamento.ifpa.edu.br-80-error.log; # ALTERAR

    # Redirect all HTTP to HTTPS
    location / {
        return 301 https://$server_name$request_uri;
    }
}

server {
    listen 443 ssl http2;
    server_name polare-treinamento.ifpa.edu.br; # ALTERAR

    client_max_body_size 128M;

    access_log /var/log/nginx/polare-treinamento.ifpa.edu.br-443-access.log; # ALTERAR
    error_log /var/log/nginx/polare-treinamento.ifpa.edu.br-443-error.log; # ALTERAR

    # certificados da instituição
    ssl_certificate /etc/nginx/ssl/nginx.crt; # ALTERAR
    ssl_certificate_key /etc/nginx/ssl/nginx.key; # ALTERAR

    ssl_session_cache shared:SSL:1m;
    ssl_prefer_server_ciphers on;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDH+AESGCM:ECDH+AES256:ECDH+AES128:DH+3DES:!ADH:!AECDH:!MD5;

    location = / {
```

```

        return 302 /polare;
    }

    location = /polare-publico/ {
        return 302 /polare-publico/relatorios;
    }

    location /polare/ {
        proxy_pass http://polare;
        proxy_cache backcache;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Real-Port $server_port;
        proxy_set_header X-Real-Scheme $scheme;
    }

    location /polare-publico/ {
        proxy_pass http://polare;
        proxy_cache backcache;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Real-Port $server_port;
        proxy_set_header X-Real-Scheme $scheme;
    }
}

```

Para mais detalhes sobre a configuração no nginx acesse

<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-22-04>

**catalina.sh** Arquivo fundamental referente aos parâmetros de configuração do sistema Polare. É necessário fazer ajustes nesse arquivo em função das configurações do ambiente local (entre as linhas 334 e 380).

```

1# /bin/sh
2
3# Licensed to the Apache Software Foundation (ASF) under one or more
4# contributor license agreements. See the NOTICE file distributed with
5# this work for additional information regarding copyright ownership.
6# The ASF licenses this file to You under the Apache License, Version 2.0
7# (the "License"); you may not use this file except in compliance with
8# the License. You may obtain a copy of the License at
9#
10# http://www.apache.org/licenses/LICENSE-2.0
11#

```

```
12# Unless required by applicable law or agreed to in writing, software
13# distributed under the License is distributed on an "AS IS" BASIS,
14# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15# See the License for the specific language governing permissions and
16# limitations under the License.
17
18# -----
19# Control Script for the CATALINA Server
20#
21# For supported commands call "catalina.sh help" or see the usage section at
22# the end of this file.
23#
24# Environment Variable Prerequisites
25#
26# Do not set the variables in this script. Instead put them into a script
27# setenv.sh in CATALINA_BASE/bin to keep your customizations separate.
28#
29# CATALINA_HOME May point at your Catalina "build" directory.
30#
31# CATALINA_BASE (Optional) Base directory for resolving dynamic portions
32# of a Catalina installation. If not present, resolves to
33# the same directory that CATALINA_HOME points to.
34#
35# CATALINA_OUT (Optional) Full path to a file where stdout and stderr
36# will be redirected.
37# Default is $CATALINA_BASE/logs/catalina.out
38#
39# CATALINA_OUT_CMD (Optional) Command which will be executed and receive
40# as its stdin the stdout and stderr from the Tomcat java
41# process. If CATALINA_OUT_CMD is set, the value of
42# CATALINA_OUT will be used as a named pipe.
43# No default.
44# Example (all one line)
45# CATALINA_OUT_CMD="/usr/bin/rotatelog -f $CATALINA_BASE/logs/catalina.out
46#
47# CATALINA_OPTS (Optional) Java runtime options used when the "start",
48# "run" or "debug" command is executed.
49# Include here and not in JAVA_OPTS all options, that should
50# only be used by Tomcat itself, not by the stop process,
51# the version command etc.
52# Examples are heap size, GC logging, JMX ports etc.
53#
54# CATALINA_TMPDIR (Optional) Directory path location of temporary directory
55# the JVM should use (java.io.tmpdir). Defaults to
56# $CATALINA_BASE/temp.
57#
58# JAVA_HOME Must point at your Java Development Kit installation.
59# Required to run the with the "debug" argument.
60#
61# JRE_HOME Must point at your Java Runtime installation.
62# Defaults to JAVA_HOME if empty. If JRE_HOME and JAVA_HOME
63# are both set, JRE_HOME is used.
```

```

64#
65#  JAVA_OPTS      (Optional) Java runtime options used when any command
66#                  is executed.
67#                  Include here and not in CATALINA_OPTS all options, that
68#                  should be used by Tomcat and also by the stop process,
69#                  the version command etc.
70#                  Most options should go into CATALINA_OPTS.
71#
72#  JAVA_ENDORSED_DIRS (Optional) Lists of of colon separated directories
73#                  containing some jars in order to allow replacement of APIs
74#                  created outside of the JCP (i.e. DOM and SAX from W3C).
75#                  It can also be used to update the XML parser implementation.
76#                  This is only supported for Java <= 8.
77#                  Defaults to $CATALINA_HOME/endorsed.
78#
79#  JPDA_TRANSPORT  (Optional) JPDA transport used when the "jpda start"
80#                  command is executed. The default is "dt_socket".
81#
82#  JPDA_ADDRESS    (Optional) Java runtime options used when the "jpda start"
83#                  command is executed. The default is localhost:8000.
84#
85#  JPDA_SUSPEND    (Optional) Java runtime options used when the "jpda start"
86#                  command is executed. Specifies whether JVM should suspend
87#                  execution immediately after startup. Default is "n".
88#
89#  JPDA_OPTS       (Optional) Java runtime options used when the "jpda start"
90#                  command is executed. If used, JPDA_TRANSPORT, JPDA_ADDRESS,
91#                  and JPDA_SUSPEND are ignored. Thus, all required jpda
92#                  options MUST be specified. The default is:
93#
94#                  -agentlib:jdwp=transport=$JPDA_TRANSPORT,
95#                  address=$JPDA_ADDRESS,server=y,suspend=$JPDA_SUSPEND
96#
97#  JSSE_OPTS       (Optional) Java runtime options used to control the TLS
98#                  implementation when JSSE is used. Default is:
99#                  "-Djdk.tls.ephemeralDHKeySize=2048"
100#
101#  CATALINA_PID    (Optional) Path of the file which should contains the pid
102#                  of the catalina startup java process, when start (fork) is
103#                  used
104#
105#  CATALINA_LOGGING_CONFIG (Optional) Override Tomcat's logging config file
106#                  Example (all one line)
107#                  CATALINA_LOGGING_CONFIG="-Djava.util.logging.config.file=$CATALINA_BASE/c
108#
109#  LOGGING_CONFIG  Depreciated
110#                  Use CATALINA_LOGGING_CONFIG
111#                  This is only used if CATALINA_LOGGING_CONFIG is not set
112#                  and LOGGING_CONFIG starts with "-D..."
113#
114#  LOGGING_MANAGER (Optional) Override Tomcat's logging manager
115#                  Example (all one line)

```

```

116#          LOGGING_MANAGER="-Djava.util.logging.manager=org.apache.juli.ClassLoaderL
117#
118#  UMASK          (Optional) Override Tomcat's default UMASK of 0027
119#
120#  USE_NOHUP      (Optional) If set to the string true the start command will
121#                  use nohup so that the Tomcat process will ignore any hangup
122#                  signals. Default is "false" unless running on HP-UX in which
123#                  case the default is "true"
124# -----
125
126# OS specific support.  $var _must_ be set to either true or false.
127cygwin=false
128darwin=false
129os400=false
130hpux=false
131case "`uname`" in
132  CYGWIN*)  cygwin=true;;
133  Darwin*)  darwin=true;;
134  OS400*)   os400=true;;
135  HP-UX*)   hpux=true;;
136  esac
137
138# resolve links - $0 may be a softlink
139PRG="$0"
140
141while [ -h "$PRG" ]; do
142  ls=`ls -ld "$PRG"`
143  link=`expr "$ls" : '.*-> \(.*\)$'`
144  if expr "$link" : '/.*' > /dev/null; then
145    PRG="$link"
146  else
147    PRG=`dirname "$PRG""/"$link"
148  fi
149done
150
151# Get standard environment variables
152PRGDIR=`dirname "$PRG"`
153
154# Only set CATALINA_HOME if not already set
155[ -z "$CATALINA_HOME" ] && CATALINA_HOME=`cd "$PRGDIR/.." >/dev/null; pwd`
156
157# Copy CATALINA_BASE from CATALINA_HOME if not already set
158[ -z "$CATALINA_BASE" ] && CATALINA_BASE="$CATALINA_HOME"
159
160# Ensure that any user defined CLASSPATH variables are not used on startup,
161# but allow them to be specified in setenv.sh, in rare case when it is needed.
162CLASSPATH=
163
164if [ -r "$CATALINA_BASE/bin/setenv.sh" ]; then
165  . "$CATALINA_BASE/bin/setenv.sh"
166elif [ -r "$CATALINA_HOME/bin/setenv.sh" ]; then
167  . "$CATALINA_HOME/bin/setenv.sh"

```

```
168fi
169
170# For Cygwin, ensure paths are in UNIX format before anything is touched
171if $cygwin; then
172 [ -n "$JAVA_HOME" ] && JAVA_HOME=`cygpath --unix "$JAVA_HOME"`
173 [ -n "$JRE_HOME" ] && JRE_HOME=`cygpath --unix "$JRE_HOME"`
174 [ -n "$CATALINA_HOME" ] && CATALINA_HOME=`cygpath --unix "$CATALINA_HOME"`
175 [ -n "$CATALINA_BASE" ] && CATALINA_BASE=`cygpath --unix "$CATALINA_BASE"`
176 [ -n "$CLASSPATH" ] && CLASSPATH=`cygpath --path --unix "$CLASSPATH"`
177fi
178
179# Ensure that neither CATALINA_HOME nor CATALINA_BASE contains a colon
180# as this is used as the separator in the classpath and Java provides no
181# mechanism for escaping if the same character appears in the path.
182case $CATALINA_HOME in
183  *:*) echo "Using CATALINA_HOME:   $CATALINA_HOME";
184      echo "Unable to start as CATALINA_HOME contains a colon (:) character";
185      exit 1;
186esac
187case $CATALINA_BASE in
188  *:*) echo "Using CATALINA_BASE:   $CATALINA_BASE";
189      echo "Unable to start as CATALINA_BASE contains a colon (:) character";
190      exit 1;
191esac
192
193# For OS400
194if $os400; then
195 # Set job priority to standard for interactive (interactive - 6) by using
196 # the interactive priority - 6, the helper threads that respond to requests
197 # will be running at the same priority as interactive jobs.
198 COMMAND=' chgjob job('$JOBNAME') runpty(6)'
199 system $COMMAND
200
201 # Enable multi threading
202 export QIBM_MULTI_THREADED=Y
203fi
204
205# Get standard Java environment variables
206if $os400; then
207 # -r will Only work on the os400 if the files are:
208 # 1. owned by the user
209 # 2. owned by the PRIMARY group of the user
210 # this will not work if the user belongs in secondary groups
211 . "$CATALINA_HOME"/bin/setclasspath.sh
212else
213 if [ -r "$CATALINA_HOME"/bin/setclasspath.sh ]; then
214   . "$CATALINA_HOME"/bin/setclasspath.sh
215 else
216   echo "Cannot find $CATALINA_HOME/bin/setclasspath.sh"
217   echo "This file is needed to run this program"
218   exit 1
219 fi
```

```

220fi
221
222# Add on extra jar files to CLASSPATH
223if [ ! -z "$CLASSPATH" ] ; then
224  CLASSPATH="$CLASSPATH":
225fi
226CLASSPATH="$CLASSPATH"$CATALINA_HOME/bin/bootstrap.jar
227
228if [ -z "$CATALINA_OUT" ] ; then
229  CATALINA_OUT="$CATALINA_BASE"/logs/catalina.out
230fi
231
232if [ -z "$CATALINA_TMPDIR" ] ; then
233  # Define the java.io.tmpdir to use for Catalina
234  CATALINA_TMPDIR="$CATALINA_BASE"/temp
235fi
236
237# Add tomcat-juli.jar to classpath
238# tomcat-juli.jar can be over-ridden per instance
239if [ -r "$CATALINA_BASE/bin/tomcat-juli.jar" ] ; then
240  CLASSPATH=$CLASSPATH:$CATALINA_BASE/bin/tomcat-juli.jar
241else
242  CLASSPATH=$CLASSPATH:$CATALINA_HOME/bin/tomcat-juli.jar
243fi
244
245# Bugzilla 37848: When no TTY is available, don't output to console
246have_tty=0
247if [ -t 0 ]; then
248  have_tty=1
249fi
250
251# For Cygwin, switch paths to Windows format before running java
252if $cygwin; then
253  JAVA_HOME=`cygpath --absolute --windows "$JAVA_HOME"`
254  JRE_HOME=`cygpath --absolute --windows "$JRE_HOME"`
255  CATALINA_HOME=`cygpath --absolute --windows "$CATALINA_HOME"`
256  CATALINA_BASE=`cygpath --absolute --windows "$CATALINA_BASE"`
257  CATALINA_TMPDIR=`cygpath --absolute --windows "$CATALINA_TMPDIR"`
258  CLASSPATH=`cygpath --path --windows "$CLASSPATH"`
259  [ -n "$JAVA_ENDORSED_DIRS" ] && JAVA_ENDORSED_DIRS=`cygpath --path --windows "$JAVA_ENDORSE
260fi
261
262if [ -z "$JSSE_OPTS" ] ; then
263  JSSE_OPTS="-Djdk.tls.ephemeralDHKeySize=2048"
264fi
265JAVA_OPTS="$JAVA_OPTS $JSSE_OPTS"
266
267# Register custom URL handlers
268# Do this here so custom URL handles (specifically 'war:...') can be used in the security pol
269JAVA_OPTS="$JAVA_OPTS -Djava.protocol.handler.pkgs=org.apache.catalina.webresources"
270
271# Check for the deprecated LOGGING_CONFIG

```



```
272# Only use it if CATALINA_LOGGING_CONFIG is not set and LOGGING_CONFIG starts with "-D..."
273if [ -z "$CATALINA_LOGGING_CONFIG" ]; then
274  case $LOGGING_CONFIG in
275    -D*) CATALINA_LOGGING_CONFIG="$LOGGING_CONFIG"
276  esac
277fi
278
279# Set juli LogManager config file if it is present and an override has not been issued
280if [ -z "$CATALINA_LOGGING_CONFIG" ]; then
281  if [ -r "$CATALINA_BASE"/conf/logging.properties ]; then
282    CATALINA_LOGGING_CONFIG="-Djava.util.logging.config.file=$CATALINA_BASE/conf/logging.prop
283  else
284    # Bugzilla 45585
285    CATALINA_LOGGING_CONFIG="-Dnop"
286  fi
287fi
288
289if [ -z "$LOGGING_MANAGER" ]; then
290  LOGGING_MANAGER="-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager"
291fi
292
293# Set UMASK unless it has been overridden
294if [ -z "$UMASK" ]; then
295  UMASK="0027"
296fi
297umask $UMASK
298
299# Java 9 no longer supports the java.endorsed.dirs
300# system property. Only try to use it if
301# JAVA_ENDORSED_DIRS was explicitly set
302# or CATALINA_HOME/endorsed exists.
303ENDORSED_PROP=ignore.endorsed.dirs
304if [ -n "$JAVA_ENDORSED_DIRS" ]; then
305  ENDORSED_PROP=java.endorsed.dirs
306fi
307if [ -d "$CATALINA_HOME/endorsed" ]; then
308  ENDORSED_PROP=java.endorsed.dirs
309fi
310
311# Make the umask available when using the org.apache.catalina.security.SecurityListener
312JAVA_OPTS="$JAVA_OPTS -Dorg.apache.catalina.security.SecurityListener.UMASK=`umask`"
313
314if [ -z "$USE_NOHUP" ]; then
315  if $hpux; then
316    USE_NOHUP="true"
317  else
318    USE_NOHUP="false"
319  fi
320fi
321unset _NOHUP
322if [ "$USE_NOHUP" = "true" ]; then
323  _NOHUP="nohup"
```

```

324fi
325
326# Add the JAVA 9 specific start-up parameters required by Tomcat
327JDK_JAVA_OPTIONS="$JDK_JAVA_OPTIONS --add-opens=java.base/java.lang=ALL-UNNAMED"
328JDK_JAVA_OPTIONS="$JDK_JAVA_OPTIONS --add-opens=java.base/java.io=ALL-UNNAMED"
329JDK_JAVA_OPTIONS="$JDK_JAVA_OPTIONS --add-opens=java.base/java.util=ALL-UNNAMED"
330JDK_JAVA_OPTIONS="$JDK_JAVA_OPTIONS --add-opens=java.base/java.util.concurrent=ALL-UNNAMED"
331JDK_JAVA_OPTIONS="$JDK_JAVA_OPTIONS --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED"
332export JDK_JAVA_OPTIONS
333
334# configuração BANCO DE DADOS
335CATALINA_OPTS="$CATALINA_OPTS -Dspring.datasource.username=postgres"
336CATALINA_OPTS="$CATALINA_OPTS -Dspring.datasource.password=postgres"
337CATALINA_OPTS="$CATALINA_OPTS -Dspring.datasource.url=jdbc:postgresql://polare-db:5432/polare
338
339# configuração OAUTH
340CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn-api.provider=
341CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn-api.client-id=
342CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn-api.client-se
343CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn-api.authoriza
344CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn.client-id=ALT
345CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn.client-secret
346CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn.scope=read"
347CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn.authorization
348CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.ufrn.redirect-uri=
349CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.ufrn.authorization-uri
350CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.ufrn.token-uri=https:/
351CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.ufrn.user-info-uri=htt
352CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.ufrn.user-name-attribu
353CATALINA_OPTS="$CATALINA_OPTS -Dapp.auth.logout.ufrn.logout-uri=https://ALTERAR/authz-server/
354
355# configuração API serviços
356CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.api-key=ALTERAR" # ALTERAR
357CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.arquivos=https://ALTERAR/file/v1/arquiv
358CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.unidades=https://ALTERAR/unidade/v1/uni
359CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.responsaveis=https://ALTERAR/pessoa/v1/
360CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.servidor-localizacoes=https://ALTERAR/p
361CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.usuarios-sig=https://ALTERAR/usuario/v1
362CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.servidores=https://ALTERAR/pessoa/v1/se
363CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.unidades-lotacao=https://ALTERAR/pesso
364CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.unidades-exercicios=https://ALTERAR/pes
365CATALINA_OPTS="$CATALINA_OPTS -Dapp.api.ufrn.services.unidades-localizacao=https://ALTERAR/pe
366CATALINA_OPTS="$CATALINA_OPTS -Dapp.polare-url=http://ALTERAR/polare/login" # ALTERAR
367CATALINA_OPTS="$CATALINA_OPTS -Dapp.polare-publico-url=http://ALTERAR/polare-publico" # ALTER
368
369# configuração GOVBR
370CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.govbr.client-id=ALT
371CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.govbr.client-secre
372CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.govbr.scope=openic
373CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.govbr.authorizatio
374CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.registration.govbr.redirect-uri
375CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.govbr.authorization-ur

```

```

376CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.govbr.token-uri=https:
377CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.govbr.jwk-set-uri=http
378CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.govbr.user-info-uri=ht
379CATALINA_OPTS="$CATALINA_OPTS -Dspring.security.oauth2.client.provider.govbr.user-name-attri
380CATALINA_OPTS="$CATALINA_OPTS -Dapp.auth.logout.govbr.logout-uri=https://sso.staging.acesso.g
381export CATALINA_OPTS
382
383# ----- Execute The Requested Command -----
384
385# Bugzilla 37848: only output this if we have a TTY
386if [ $have_tty -eq 1 ]; then
387  echo "Using CATALINA_BASE:    $CATALINA_BASE"
388  echo "Using CATALINA_HOME:    $CATALINA_HOME"
389  echo "Using CATALINA_TMPDIR:  $CATALINA_TMPDIR"
390  if [ "$1" = "debug" ] ; then
391    echo "Using JAVA_HOME:        $JAVA_HOME"
392  else
393    echo "Using JRE_HOME:         $JRE_HOME"
394  fi
395  echo "Using CLASSPATH:         $CLASSPATH"
396  echo "Using CATALINA_OPTS:    $CATALINA_OPTS"
397  if [ ! -z "$CATALINA_PID" ]; then
398    echo "Using CATALINA_PID:      $CATALINA_PID"
399  fi
400fi
401
402if [ "$1" = "jpda" ] ; then
403  if [ -z "$JPDA_TRANSPORT" ]; then
404    JPDA_TRANSPORT="dt_socket"
405  fi
406  if [ -z "$JPDA_ADDRESS" ]; then
407    JPDA_ADDRESS="localhost:8000"
408  fi
409  if [ -z "$JPDA_SUSPEND" ]; then
410    JPDA_SUSPEND="n"
411  fi
412  if [ -z "$JPDA_OPTS" ]; then
413    JPDA_OPTS="-agentlib:jdwp=transport=$JPDA_TRANSPORT,address=$JPDA_ADDRESS,server=y,suspen
414  fi
415  CATALINA_OPTS="$JPDA_OPTS $CATALINA_OPTS"
416  shift
417fi
418
419if [ "$1" = "debug" ] ; then
420  if $os400; then
421    echo "Debug command not available on OS400"
422    exit 1
423  else
424    shift
425    if [ "$1" = "-security" ] ; then
426      if [ $have_tty -eq 1 ]; then
427        echo "Using Security Manager"

```

```

428     fi
429     shift
430     eval exec "\"$_RUNJDB\"" "\"$CATALINA_LOGGING_CONFIG\"" $LOGGING_MANAGER "$JAVA_OPTS" "
431         -D$ENDORSED_PROP="$JAVA_ENDORSED_DIRS" \
432         -classpath "$CLASSPATH" \
433         -sourcepath "$CATALINA_HOME"/../.. /java \
434         -Djava.security.manager \
435         -Djava.security.policy=="$CATALINA_BASE"/conf/catalina.policy \
436         -Dcatalina.base="$CATALINA_BASE" \
437         -Dcatalina.home="$CATALINA_HOME" \
438         -Djava.io.tmpdir="$CATALINA_TMPDIR" \
439         org.apache.catalina.startup.Bootstrap "$@" start
440     else
441         eval exec "\"$_RUNJDB\"" "\"$CATALINA_LOGGING_CONFIG\"" $LOGGING_MANAGER "$JAVA_OPTS" "
442         -D$ENDORSED_PROP="$JAVA_ENDORSED_DIRS" \
443         -classpath "$CLASSPATH" \
444         -sourcepath "$CATALINA_HOME"/../.. /java \
445         -Dcatalina.base="$CATALINA_BASE" \
446         -Dcatalina.home="$CATALINA_HOME" \
447         -Djava.io.tmpdir="$CATALINA_TMPDIR" \
448         org.apache.catalina.startup.Bootstrap "$@" start
449     fi
450 fi
451
452 elif [ "$1" = "run" ]; then
453
454     shift
455     if [ "$1" = "-security" ] ; then
456         if [ $have_tty -eq 1 ]; then
457             echo "Using Security Manager"
458         fi
459         shift
460         eval exec "\"$_RUNJAVA\"" "\"$CATALINA_LOGGING_CONFIG\"" $LOGGING_MANAGER "$JAVA_OPTS" "$
461             -D$ENDORSED_PROP="\"$JAVA_ENDORSED_DIRS\"" \
462             -classpath "\"$CLASSPATH\"" \
463             -Djava.security.manager \
464             -Djava.security.policy=="\"$CATALINA_BASE/conf/catalina.policy\"" \
465             -Dcatalina.base="\"$CATALINA_BASE\"" \
466             -Dcatalina.home="\"$CATALINA_HOME\"" \
467             -Djava.io.tmpdir="\"$CATALINA_TMPDIR\"" \
468             org.apache.catalina.startup.Bootstrap "$@" start
469     else
470         eval exec "\"$_RUNJAVA\"" "\"$CATALINA_LOGGING_CONFIG\"" $LOGGING_MANAGER "$JAVA_OPTS" "$
471             -D$ENDORSED_PROP="\"$JAVA_ENDORSED_DIRS\"" \
472             -classpath "\"$CLASSPATH\"" \
473             -Dcatalina.base="\"$CATALINA_BASE\"" \
474             -Dcatalina.home="\"$CATALINA_HOME\"" \
475             -Djava.io.tmpdir="\"$CATALINA_TMPDIR\"" \
476             org.apache.catalina.startup.Bootstrap "$@" start
477     fi
478
479 elif [ "$1" = "start" ] ; then

```

```
480
481 if [ ! -z "$CATALINA_PID" ]; then
482     if [ -f "$CATALINA_PID" ]; then
483         if [ -s "$CATALINA_PID" ]; then
484             echo "Existing PID file found during start."
485             if [ -r "$CATALINA_PID" ]; then
486                 PID=`cat "$CATALINA_PID"`
487                 ps -p $PID >/dev/null 2>&1
488                 if [ $? -eq 0 ] ; then
489                     echo "Tomcat appears to still be running with PID $PID. Start aborted."
490                     echo "If the following process is not a Tomcat process, remove the PID file and t
491                     ps -f -p $PID
492                     exit 1
493                 else
494                     echo "Removing/clearing stale PID file."
495                     rm -f "$CATALINA_PID" >/dev/null 2>&1
496                     if [ $? != 0 ]; then
497                         if [ -w "$CATALINA_PID" ]; then
498                             cat /dev/null > "$CATALINA_PID"
499                         else
500                             echo "Unable to remove or clear stale PID file. Start aborted."
501                             exit 1
502                         fi
503                     fi
504                 fi
505             else
506                 echo "Unable to read PID file. Start aborted."
507                 exit 1
508             fi
509         else
510             rm -f "$CATALINA_PID" >/dev/null 2>&1
511             if [ $? != 0 ]; then
512                 if [ ! -w "$CATALINA_PID" ]; then
513                     echo "Unable to remove or write to empty PID file. Start aborted."
514                     exit 1
515                 fi
516             fi
517         fi
518     fi
519 fi
520
521 shift
522 if [ -z "$CATALINA_OUT_CMD" ] ; then
523     touch "$CATALINA_OUT"
524 else
525     if [ ! -e "$CATALINA_OUT" ]; then
526         if ! mkfifo "$CATALINA_OUT"; then
527             echo "cannot create named pipe $CATALINA_OUT. Start aborted."
528             exit 1
529         fi
530     elif [ ! -p "$CATALINA_OUT" ]; then
531         echo "$CATALINA_OUT exists and is not a named pipe. Start aborted."
```

```

532     exit 1
533 fi
534 $CATALINA_OUT_CMD <"$CATALINA_OUT" &
535 fi
536 if [ "$1" = "-security" ] ; then
537     if [ $have_tty -eq 1 ]; then
538         echo "Using Security Manager"
539     fi
540     shift
541     eval $ _NOHUP "\"$ _RUNJAVA\""" "\"$CATALINA_LOGGING_CONFIG\""" $LOGGING_MANAGER "$JAVA_OPTS"
542         -D$ENDORSED_PROP="\"$JAVA_ENDORSED_DIRS\""" \
543         -classpath "\"$CLASSPATH\""" \
544         -Djava.security.manager \
545         -Djava.security.policy=="\"$CATALINA_BASE/conf/catalina.policy\""" \
546         -Dcatalina.base="\"$CATALINA_BASE\""" \
547         -Dcatalina.home="\"$CATALINA_HOME\""" \
548         -Djava.io.tmpdir="\"$CATALINA_TMPDIR\""" \
549         org.apache.catalina.startup.Bootstrap "$@" start \
550         >> "$CATALINA_OUT" 2>&1 "&"
551
552 else
553     eval $ _NOHUP "\"$ _RUNJAVA\""" "\"$CATALINA_LOGGING_CONFIG\""" $LOGGING_MANAGER "$JAVA_OPTS"
554         -D$ENDORSED_PROP="\"$JAVA_ENDORSED_DIRS\""" \
555         -classpath "\"$CLASSPATH\""" \
556         -Dcatalina.base="\"$CATALINA_BASE\""" \
557         -Dcatalina.home="\"$CATALINA_HOME\""" \
558         -Djava.io.tmpdir="\"$CATALINA_TMPDIR\""" \
559         org.apache.catalina.startup.Bootstrap "$@" start \
560         >> "$CATALINA_OUT" 2>&1 "&"
561
562 fi
563
564 if [ ! -z "$CATALINA_PID" ]; then
565     echo $! > "$CATALINA_PID"
566 fi
567
568 echo "Tomcat started."
569
570 elif [ "$1" = "stop" ] ; then
571
572     shift
573
574     SLEEP=5
575     if [ ! -z "$1" ]; then
576         echo $1 | grep "[^0-9]" >/dev/null 2>&1
577         if [ $? -gt 0 ]; then
578             SLEEP=$1
579             shift
580         fi
581     fi
582
583     FORCE=0

```

```

584 if [ "$1" = "-force" ]; then
585     shift
586     FORCE=1
587 fi
588
589 if [ ! -z "$CATALINA_PID" ]; then
590     if [ -f "$CATALINA_PID" ]; then
591         if [ -s "$CATALINA_PID" ]; then
592             kill -0 `cat "$CATALINA_PID"` >/dev/null 2>&1
593             if [ $? -gt 0 ]; then
594                 echo "PID file found but either no matching process was found or the current user c
595                 exit 1
596             fi
597         else
598             echo "PID file is empty and has been ignored."
599         fi
600     else
601         echo "\$CATALINA_PID was set but the specified file does not exist. Is Tomcat running?
602         exit 1
603     fi
604 fi
605
606 eval "\"$_RUNJAVA\" \"$LOGGING_MANAGER\" \"$JAVA_OPTS\" \
607     -D$ENDORSED_PROP=\"\$JAVA_ENDORSED_DIRS\" \
608     -classpath \"\$CLASSPATH\" \
609     -Dcatalina.base=\"\$CATALINA_BASE\" \
610     -Dcatalina.home=\"\$CATALINA_HOME\" \
611     -Djava.io.tmpdir=\"\$CATALINA_TMPDIR\" \
612     org.apache.catalina.startup.Bootstrap \"$@" stop
613
614 # stop failed. Shutdown port disabled? Try a normal kill.
615 if [ $? != 0 ]; then
616     if [ ! -z "$CATALINA_PID" ]; then
617         echo "The stop command failed. Attempting to signal the process to stop through OS sign
618         kill -15 `cat "$CATALINA_PID"` >/dev/null 2>&1
619     fi
620 fi
621
622 if [ ! -z "$CATALINA_PID" ]; then
623     if [ -f "$CATALINA_PID" ]; then
624         while [ $SLEEP -ge 0 ]; do
625             kill -0 `cat "$CATALINA_PID"` >/dev/null 2>&1
626             if [ $? -gt 0 ]; then
627                 rm -f "$CATALINA_PID" >/dev/null 2>&1
628                 if [ $? != 0 ]; then
629                     if [ -w "$CATALINA_PID" ]; then
630                         cat /dev/null > "$CATALINA_PID"
631                         # If Tomcat has stopped don't try and force a stop with an empty PID file
632                         FORCE=0
633                     else
634                         echo "The PID file could not be removed or cleared."
635                     fi

```

```
636         fi
637         echo "Tomcat stopped."
638         break
639     fi
640     if [ $SLEEP -gt 0 ]; then
641         sleep 1
642     fi
643     if [ $SLEEP -eq 0 ]; then
644         echo "Tomcat did not stop in time."
645         if [ $FORCE -eq 0 ]; then
646             echo "PID file was not removed."
647         fi
648         echo "To aid diagnostics a thread dump has been written to standard out."
649         kill -3 `cat "$CATALINA_PID"`
650     fi
651     SLEEP=`expr $SLEEP - 1 `
652 done
653 fi
654 fi
655
656 KILL_SLEEP_INTERVAL=5
657 if [ $FORCE -eq 1 ]; then
658     if [ -z "$CATALINA_PID" ]; then
659         echo "Kill failed: \$CATALINA_PID not set"
660     else
661         if [ -f "$CATALINA_PID" ]; then
662             PID=`cat "$CATALINA_PID"`
663             echo "Killing Tomcat with the PID: $PID"
664             kill -9 $PID
665             while [ $KILL_SLEEP_INTERVAL -ge 0 ]; do
666                 kill -0 `cat "$CATALINA_PID"` >/dev/null 2>&1
667                 if [ $? -gt 0 ]; then
668                     rm -f "$CATALINA_PID" >/dev/null 2>&1
669                     if [ $? != 0 ]; then
670                         if [ -w "$CATALINA_PID" ]; then
671                             cat /dev/null > "$CATALINA_PID"
672                         else
673                             echo "The PID file could not be removed."
674                         fi
675                     fi
676                     echo "The Tomcat process has been killed."
677                     break
678                 fi
679                 if [ $KILL_SLEEP_INTERVAL -gt 0 ]; then
680                     sleep 1
681                 fi
682                 KILL_SLEEP_INTERVAL=`expr $KILL_SLEEP_INTERVAL - 1 `
683             done
684             if [ $KILL_SLEEP_INTERVAL -lt 0 ]; then
685                 echo "Tomcat has not been killed completely yet. The process might be waiting on
686             fi
687         fi
```



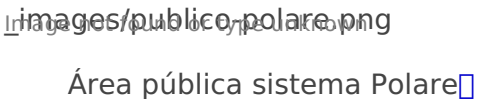
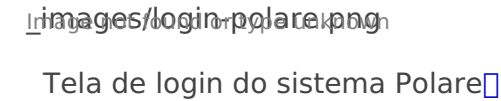
```

688     fi
689 fi
690
691 elif [ "$1" = "configtest" ] ; then
692
693     eval "\"$_RUNJAVA\" \"$LOGGING_MANAGER \"$JAVA_OPTS" \
694         -D$ENDORSED_PROP=\"$_JAVA_ENDORSED_DIRS\" \" \
695         -classpath \"$_CLASSPATH\" \" \
696         -Dcatalina.base=\"$_CATALINA_BASE\" \" \
697         -Dcatalina.home=\"$_CATALINA_HOME\" \" \
698         -Djava.io.tmpdir=\"$_CATALINA_TMPDIR\" \" \
699         org.apache.catalina.startup.Bootstrap configtest
700     result=$?
701     if [ $result -ne 0 ]; then
702         echo "Configuration error detected!"
703     fi
704     exit $result
705
706 elif [ "$1" = "version" ] ; then
707
708     "$_RUNJAVA" \
709         -classpath "$CATALINA_HOME/lib/catalina.jar" \
710         org.apache.catalina.util.ServerInfo
711
712 else
713
714     echo "Usage: catalina.sh ( commands ... )"
715     echo "commands:"
716     if $os400; then
717         echo "    debug           Start Catalina in a debugger (not available on OS400)"
718         echo "    debug -security Debug Catalina with a security manager (not available on OS400)"
719     else
720         echo "    debug           Start Catalina in a debugger"
721         echo "    debug -security Debug Catalina with a security manager"
722     fi
723     echo "    jpda start      Start Catalina under JPDA debugger"
724     echo "    run             Start Catalina in the current window"
725     echo "    run -security   Start in the current window with security manager"
726     echo "    start           Start Catalina in a separate window"
727     echo "    start -security Start in a separate window with security manager"
728     echo "    stop            Stop Catalina, waiting up to 5 seconds for the process to end"
729     echo "    stop n          Stop Catalina, waiting up to n seconds for the process to end"
730     echo "    stop -force     Stop Catalina, wait up to 5 seconds and then use kill -KILL if st
731     echo "    stop n -force   Stop Catalina, wait up to n seconds and then use kill -KILL if st
732     echo "    configtest      Run a basic syntax check on server.xml - check exit code for resu
733     echo "    version         What version of tomcat are you running?"
734     echo "Note: Waiting for the process to end and use of the -force option require that \"$_CATALINA_HOME\" be
735     exit 1
736
737 fi

```

O seguinte comando deve ser executado no diretório que contém os arquivos anteriores para criar os contêiners e infraestrutura necessária para o funcionamento do sistema Polare:

```
docker-compose up -d
```



Warning

O login no sistema Polare só poderá ser efetuado se uma infraestrutura OAUTH estiver instalada e configurada. As duas opções atualmente são a infraestrutura fornecida pela UFRN (descrita na sessão Pré-Requisitos de Instalação), ou o login único pelo GOVBR.

[Previous](#)[Next](#)

**Sponsored: Guardsquare**  
Secure your iOS and Android apps with multiple layers of code hardening and RASP. **Learn More**

**Ads by EthicalAds**

# Instalação do Sistema