

Solução de Arquitetura e Configuração



PsychoManager
iidh.com.br

Fabio Tassinari, Arquiteto.

Janeiro de 2018

Sumário

Informações deste Documento 3

- Originador 3**
- Propriedade 3**
- Copyright 3**
- Distribuição 3**
- Confidencialidade 3**
- Cópias Adicionais 3**
- Publico Alvo 3**
- Detalhes do Projeto 4**
- Informações de Contato 4**
- Histórico de Versão 4**
- Documentos relacionados 4**

INTRODUÇÃO 5

Definição de Arquitetura 5

Especificações Técnicas 6

- Servidores 6**
- Consumo inicial 6**
- Rede 6**
- DNS 6**
- Firewall 7**
- Domino 7**
- Outros 7**
- Autenticação 7**

Informações essenciais 8

Roteiro de Instalação 9

- Infra 9**
- Aplicação 12**

Informações deste Documento

Originador

Facta Sistemas

Propriedade

Integração - Instituto de Desenvolvimento Humano.

Copyright

Este documento contém informações confidenciais que são de uso exclusivo da IIDH e não devem ser compartilhadas com pessoas que não pertençam a esta companhia. Este documento e qualquer parte dele não podem ser copiados, reproduzido, fotocopiado, armazenado eletronicamente em um sistema de recuperação, ou transmitido sem o consentimento expresso por escrito da IIDH. A Facta Sistemas não garante que este documento esteja livre de erros ou omissões. A Facta Sistemas se reserva o direito de fazer correções, atualizações, revisões ou alterações nas informações aqui contidas.

Distribuição

Não envie ou copie sem a permissão por escrito da IIDH.
As cópias deste documento estão restritas a:

- Integração - Instituto de Desenvolvimento Humano

Confidencialidade

Todas as informações descritas neste documento são consideradas confidenciais.

Cópias Adicionais

Cópias adicionais deste documento podem ser obtidas com os representantes da IIDH.

Publico Alvo

Este documento destina-se à equipe técnica da IIDH, responsáveis pelo suporte e manutenção de todo e qualquer sistema ou software relacionado ao seu compromisso com a Facta Sistemas.

Detalhes do Projeto

Cliente, Local	IIDH
Local	São Paulo – SP – Brasil
Duração	
Autor	Fabio Cesar Tassinari
Arquitetos	Fabio Cesar Tassinari

Informações de Contato

Nome	Fabio Cesar Tassinari
Cargo	Arquiteto
Email	fctassinari@gmail.com

Histórico de Versão

Versão	Data	Autor	Descrição
1.0	25/01/2018	Fabio C. Tassinari	Versão Inicial

Documentos relacionados

Esta documentação faz referências às informações disponíveis nos sites abaixo:

- Google Cloud:

<https://cloud.google.com>

https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address?&_ga=2.13208244.-262534153.1497114097#promote_ephemeral_ip

<https://cloud.google.com/dns/quickstart>

- Bitnami

<https://docs.bitnami.com/general/faq/get-started/access-ssh-tunnel/>

<https://docs.bitnami.com/google/infrastructure/wildfly/>

<https://docs.bitnami.com/general/how-to/generate-install-lets-encrypt-ssl/>

- WildFly

<http://wildfly.org/news/2014/02/06/GlassFish-to-WildFly-migration/>

INTRODUÇÃO

Este documento descreve a arquitetura pré-definida para o ambiente de infra, englobando os produtos:

- Google Cloud Platform
- Debian 9
- Apache 2.4.29
- Java 1.8.0_151
- MySQL 5.6.38
- OpenSSL 1.0.2n
- WildFly 11.0.0

A documentação descreve também a arquitetura pré-definida para o desenvolvimento do sistema PsychoManager englobando os produtos:

- Log4j-1.2.17
- Jasperreports-6.1.0
- Fonts (proprietário)
- Primefaces-6.0
- MySql-5.1.44
- Hibernate-4.3.8
- Spring-5.0.2
- Poi-3.11
- JDK 1.8
- URL iidh.com.br

Definição de Arquitetura

O objetivo desta definição de arquitetura é a alta disponibilidade e segurança do site IIDH.COM.BR e do sistema PsychoManager.

Arquitetura de Infra:

Atualmente o sistema necessita de uma infra simples e mínima. Foi criada uma máquina virtual no Google Cloud para suportar o site e a aplicação

Arquitetura de sistema:

O Site: Foi desenvolvido de forma estática baseado em um template para posteriormente ser utilizado um gerenciador de conteúdo.

O Sistema: Foi desenvolvido em Java no modelo MVC e banco de dados MySql, midlware WildFly e webserver Apache utilizando as tecnologias descritas na Introdução.

A arquitetura de infra descrita acima é o necessário para suportar esta aplicação mais banco de dados por ser simples e de baixo consumo.

Faz-se necessária acompanhamento/monitoração contínua para análise de consumo e desempenho.

Especificações Técnicas

Esta seção descreve os detalhes técnicos necessários como requerimentos para início da implantação.

Servidores

Diante das arquiteturas acima, foram detalhadas informações de Hardware para a VM que compõem a estrutura do sistema.

Machine type	g1-small (1 vCPU, 1,7 GB memory)		
CPU platform	Intel Ivy Bridge		
Zone	us-central1-f		
Boot disk and local disks			
Name	Size (GB)	Type	Mode
psycho-vm	10	SSD persistent disk	Boot, read/write

Consumo inicial

Por ser um sistema de porte simples, não são necessários grandes recursos. Os descritos na sessão Servidores são os suficientes para suportar o sistema e o site.

Rede

A rede interna utilizada no Google Cloud é a default e foi configurado um IP fixo externo para utilização da URL iidh.com.br

DNS

Seguir a documentação para criar o DNS e atender ao site e ao sistema IIDH-Zone (Cloud DNS)

<https://cloud.google.com/dns/quickstart>

iidh-zone				
DNS name: iidh.com.br.				
Zona para o sistema/site IIDH				
Record sets				
<button>Add record set</button> <button>Delete record sets</button>				
<input type="checkbox"/> DNS name ^	Type	TTL (seconds)	Data	
<input type="checkbox"/> iidh.com.br.	A	300	35.193.163.21	
iidh.com.br.	NS	21600	ns-cloud-a1.googledomains.com. ns-cloud-a2.googledomains.com. ns-cloud-a3.googledomains.com. ns-cloud-a4.googledomains.com.	
iidh.com.br.	SOA	21600	ns-cloud-a1.googledomains.com. cloud-dns-hostmaster.google.com. 1 21600 3600 259200 300	
<input type="checkbox"/> iidh.com.br.	TXT	300	"google-site-verification=iTRgXhS2ew43-Muaxj0soaQysWKFa0tyMslRdf73_aM"	
<input type="checkbox"/> www.iidh.com.br.	CNAME	300	c.storage.googleapis.com.	

Firewall

Utilizar o default definido na criação da máquina virtual

Domino

Tanto para o site como para o sistema será utilizado o domínio **iidh.com.br**

O domínio está registrado no <https://registro.br/> com as seguintes informações.

IIDH.COM.BR		PUBLICADO	03/01/2015	03/01/2019 ESTENDER
DADOS TITULAR				
Fabio Cesar Tassinari CPF: 152.261.018-99 Rua dos Capitães Mores, 346, ap 101 bl b 03167-030 - São Paulo - SP (11) 91318744				
CONTATOS				
ADMINISTRATIVO FCT23 - Fabio Cesar Tassinari				
TÉCNICO FCT23 - Fabio Cesar Tassinari				
COBRANÇA FCT23 - Fabio Cesar Tassinari				
DNS				
MASTER NS-CLOUD-A1.GOOGLEDOMAINS.COM				
SLAVE 1 NS-CLOUD-A2.GOOGLEDOMAINS.COM				
SLAVE 2 NS-CLOUD-A3.GOOGLEDOMAINS.COM				
SLAVE 3 NS-CLOUD-A4.GOOGLEDOMAINS.COM				

Outros

Autenticação

A autenticação do sistema é feita no banco de dados

Informações essenciais

Info	Valor
Winscp / putty	
Protocolo	SFTP
Host	35.193.163.21
Port	22
User	bitnami
Pass	faka1609
Mysql	
Protocolo	SFTP
Host	localhost *Necessário estar com o putty conectado com tunneling
Port	3306
User	root
Pass	zcxySB3Kn4jc
WildFly	
Url	http://localhost:9990/console / http://35.193.163.21/index2.html
User	manager
Pass	zcxySB3Kn4jc

Roteiro de Instalação

Infra

1 - Criar Máquina virtual GoogleCloud

No menu Launcher escolher pilhas do desenvolvedor e procurar por WildFly

Criar a máquina

Assign a static external IP address to your VM instance

https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address?&_ga=2.13208244.-262534153.1497114097#promote_ephemeral_ip

Após criada seguir a documentação para acessa-la

<https://docs.bitnami.com/google/infrastructure/wildfly/>
<https://docs.bitnami.com/google/components/mysql/>

Acertar a data do sistema: **tzselect**

Atenção: só consegui acessar o Mysql com o Workbench qdo fiz um tunel com o putty conforme demonstrado na documentação do wildfly

2 - MySql

Se o sistema for migrado de Postgres para MySql:

- Instalar o Postgres e MySql em uma máquina qualquer
- Restaurar os dados para o Postgres
- Importar dados do Postgres para o MySql com o programa PSQLtoMYSQL.java
- Gerar um dump do MySql
- Ex: Windows backupMysql.cmd

PATH = %PATH%;"C:\Program Files\MySQL\MySQL Server 5.7\bin"

mysqldump --user=root --password=12345678 --host=localhost --databases psychomanager --add-drop-database > dump.sql

Se o sistema já roda com o MySql:

- Gerar dump do banco original
- Ex: Linux backupMysql.sh

#!/bin/sh

/opt/bitnami/mysql/bin/mysqldump --defaults-extra-file=/sistemasIIDH/config.cnf --socket=/opt/bitnami/mysql/tmp/mysql.sock --host=localhost --databases psychomanager --add-drop-database >

/sistemasIIDH/shared/backup/PsychoManager/bkpPsychoManager_`date "+%Y_%m_%d_%H_%M_%S"`.sql

Arquivo de senha: config.cnf

[client]

user = root

password = zcxySB3Kn4jc

- Restaurar na máquina nova
- Ex: Linux restoreMysql.sh

mysql --user=root --password=zcxySB3Kn4jc --host=localhost < dump.sql

Obs: Campos boolean Postgres true e false -> MySql 0 = false, 1 = true

3 - WildFly

Criar usuario admin para o WildFly

add-user.bat

Configurar WildFly: <http://wildfly.org/news/2014/02/06/GlassFish-to-WildFly-migration/>

Trocar o acesso dos sistemas de localhost para qq endereço
substituir o trecho abaixo

```
<interface name="public">
  <inet-address value="{jboss.bind.address:127.0.0.1}"/>
</interface>
```

por

```
<interface name="public">
```

```
<any-address/>
</interface>
```

Deploy driver mysql (a máquina do Google Cloud já vem com o driver deployado)
Criar modulo

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.44-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Dentro do standalone.xml ficará assim

```
<deployments>
  <deployment name="mysql-connector-java-5.1.44-bin.jar" runtime-
    name="mysql-connector-java-5.1.44-bin.jar">
    <content sha1="c5c76a610587bde4af517e457bc66f8f0339caac"/>
  </deployment>
</deployments>
```

Criar datasource pelo console no standalone.xml ficará assim:

```
<datasource jta="true" jndi-name="java:/jdbc/PSychoDS" pool-name="PSychoDS"
  enabled="true" use-ccm="true">
  <connection-url>
    jdbc:mysql://127.0.0.1:3306/psychomanager?autoReconnect=true&useSSL=false
  </connection-url>
  <driver-class>com.mysql.jdbc.Driver</driver-class>
  <driver>mysql-connector-java-5.1.44-bin.jar_com.mysql.jdbc.Driver_5_1</driver>
  <pool>
    <min-pool-size>1</min-pool-size>
    <initial-pool-size>1</initial-pool-size>
    <max-pool-size>2</max-pool-size>
  </pool>
  <security>
    <user-name>root</user-name>
    <password>12345678</password>
  </security>
  <validation>
    <valid-connection-checker class-name=
      "org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"/>
    <background-validation>true</background-validation>
    <exception-sorter class-name=
      "org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"/>
  </validation>
</datasource>

<drivers>
  <driver name="h2" module="com.h2database.h2">
    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
  </driver>
  <driver name="mysql" module="com.mysql">
    <driver-class>com.mysql.jdbc.Driver</driver-class>
  </driver>
</drivers>
```

Criar realm

Gerar senha

```
cd C:\Users\Tassinari\Documents\Ferramentas\wildfly-
10.1.0\modules\system\layers\base\org\picketbox\main
```

```
java -cp ./picketbox-4.9.6.Final.jar org.jboss.security.Base64Encoder <senha>
SHA-256
```

Com esta senha, Incluir pelo menos um usuário do arquivo QueriesMySQL.sql

```
insert into TB_AUT_USER (userid,password,ds_nome) values('katia','<senha
gerada>','Katia Gonçalves');
```

No standalone.xml ficará:

```
<security-domain name="psycho" cache-type="default">
<authentication>
  <login-module code="Database" flag="required">
    <module-option name="dsJndiName" value="java:/jdbc/PsychoDS"/>
    <module-option name="principalsQuery" value="select password from
tb_aut_user where userid=?"/>
    <module-option name="rolesQuery" value="select groupid, 'Roles' from
tb_aut_group ug where ug.userid = ?"/>
    <module-option name="hashAlgorithm" value="SHA-256"/>
    <module-option name="hashEncoding" value="BASE64"/>
    <module-option name="unauthenticatedIdentity" value="guest"/>
  </login-module>
  <login-module code="RoleMapping" flag="required">
    <module-option name="rolesProperties"
value="file:${jboss.server.config.dir}/psycho.properties"/>
    <module-option name="replaceRole" value="false"/>
  </login-module>
</authentication>
</security-domain>
```

Criar o arquivo **psycho.properties** em **wildfly-10.1.0\standalone\configuration** com o conteúdo

```
vi psycho.properties
```

```
Adm=Adm
User=User
```

```
chown wildfly:wildfly psycho.properties
```

3.1 – Página Welcome Wildfly / Apache

Arquivo de configuração do Apache HTTP

```
/opt/bitnami/wildfly/conf/wildfly.conf
```

Diretorio de conteúdo

```
/opt/bitnami/wildfly/welcome-content
```

4 - How To Start Or Stop The Services?

Each Bitnami stack includes a control script that lets you easily stop, start and restart services. The script is located at `/opt/bitnami/ctlscript.sh`. Call it without any service name arguments to start all services:

```
sudo /opt/bitnami/ctlscript.sh start
```

Or use it to restart a single service, such as Apache only, by passing the service name as argument:

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

Use this script to stop all services:

```
sudo /opt/bitnami/ctlscript.sh stop
```

Restart the services by running the script without any arguments:

```
sudo /opt/bitnami/ctlscript.sh restart
```

Obtain a list of available services and operations by running the script without any arguments:

```
sudo /opt/bitnami/ctlscript.sh
```

SSL

Para configurar o SSL para o site e para o sistema seguir a documentação do link abaixo

<https://docs.bitnami.com/general/how-to/generate-install-lets-encrypt-ssl/>
~~As chaves geradas ficam em /etc/letsencrypt/live/iidh.com.br~~
As chaves geradas ficam em /opt/bitnami/letsencrypt/certificates/

Conteúdo do arquivo /opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf

```
<VirtualHost _default_:443>
    ServerAdmin localhost
    DocumentRoot "/opt/bitnami/apache2/htdocs/iidh"
    ServerName iidh.com.br

    SSLEngine on
    SSLCertificateFile "/opt/bitnami/apache2/conf/server.crt"
SSLCertificateKeyFile "/opt/bitnami/apache2/conf/server.key"
    SSLCertificateFile "/opt/bitnami/apache2/conf/iidh.com.br.crt"
    SSLCertificateKeyFile "/opt/bitnami/apache2/conf/iidh.com.br.key"

    <Directory "/opt/bitnami/apache2/htdocs/iidh">
        Options FollowSymLinks MultiViews
        AddLanguage en en
        LanguagePriority en
        ForceLanguagePriority Prefer Fallback

        AllowOverride All
        <IfVersion < 2.3 >
            Order allow,deny
            Allow from all
        </IfVersion>
        <IfVersion >= 2.3 >
            Require all granted
        </IfVersion>
    </Directory>

    # Error Documents
    ErrorDocument 503 /503.html

    ProxyPass /PsychoManager http://localhost:8080/PsychoManager
    ProxyPassReverse /PsychoManager http://localhost:8080/PsychoManager

</VirtualHost>

<VirtualHost _default_:80>
    ServerName iidh.com.br
    # Redirect permanent / https://iidh.com.br

    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    RewriteRule ^/?(.*) https://iidh.com.br/$1 [R,L]

</VirtualHost>
```

Para renovar o certificado (já está no crontab do usuário)

```
/opt/bitnami/ctlscript.sh stop apache
/opt/bitnami/letsencrypt/certbot auto-renew
ls -l /etc/letsencrypt/live/iidh.com.br

sudo /opt/bitnami/letsencrypt/lego --path /opt/bitnami/letsencrypt --
email="fetassinari@gmail.com" --http --http-timeout 30 --http-webroot
/opt/bitnami/apps/letsencrypt --domains iidh.com.br renew && sudo
/opt/bitnami/apache2/bin/httpd -f /opt/bitnami/apache2/conf/httpd.conf -k graceful #
bncert autorenew
```

<https://docs.bitnami.com/general/how-to/generate-install-lets-encrypt-ssl/#alternative-approach>

```
sudo /opt/bitnami/bncert-tool
```

NOTE: The Bitnami HTTPS Configuration Tool will automatically create a cron job to renew your certificate(s). By default, if the *bitnami* user account exists on the system, the cron jobs will be added under that user account. To view and modify the cron job, use the command *sudo crontab -u bitnami -l*.

Para renovar tem o script sendo chamado no crontab do usuário Tassinari
/sistemasIIDH/renew-certificate.sh

/opt/bitnami/ctlscript.sh start apache

Obs: se renovar e parar de funcionar verificar o arquivo **/opt/bitnami/apache2/conf/httpd.conf**
Se tiver a configuração **ServerName iidh.com.br:80** trocar para **ServerName localhost:80**

Aplicação

1 - Criar as pastas abaixo:

/sistemasIIDH
shared
appfiles
PsychoManager
pdf
reportfiles
libs
PsychoManager
logs
PsychoManager

Executar o seguinte comando apos a criação das pastas
chown -R wildfly.wildfly /sistemasIIDH/

2 - Na pasta /sistemasIIDH/shared/appfiles/PsychoManager/pdf

Copiar os arquivos pdf de ficha dos pacientes escaneadas

3 - Na pasta /sistemasIIDH/shared/libs/PsychoManager/

Criar o arquivo Config.properties com o conteúdo abaixo:

pdf_path=/sistemasIIDH/shared/appfiles/PsychoManager/pdf
timeZone=America/Sao_Paulo
file_report_path=/sistemasIIDH/shared/appfiles/PsychoManager/reportfiles
qtd_cols_dash=4
tempo_notificacao=2000

Criar o arquivo log4j.properties com o conteúdo abaixo:

Use Three appenders,
#stdout - is used for write to console
#R - is used for write to file
#ALL < DEBUG < INFO < WARN < ERROR < FATAL < OFF.
log4j.rootLogger=DEBUG, stdout, rootLogFile

Print only messages of level WARN or above in the package
#This is use for debugging mode
log4j.logger.testlogging=DEBUG

Appender writes to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %5p

(%F:%L) - %m%n


```

##### Appender writes to a file
log4j.appender.rootLogFile=org.apache.log4j.RollingFileAppender
log4j.appender.rootLogFile.File=/sistemasIIDH/shared/logs/PsychoManager/PsychoManager.log
log4j.appender.rootLogFile.MaxFileSize=5000MB
log4j.appender.rootLogFile.MaxBackupIndex=10
log4j.appender.rootLogFile.layout=org.apache.log4j.PatternLayout
log4j.appender.rootLogFile.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %5p
(%F:%L) - %m%n
#log4j.appender.rootLogFile.layout.ConversionPattern=%p %t %c - %m%n

##### Appender writes to a file
log4j.appender.modelLogFile=org.apache.log4j.DailyRollingFileAppender
log4j.appender.modelLogFile.File=/sistemasIIDH/shared/logs/PsychoManager/modelLogFile.log
log4j.appender.modelLogFile.DatePattern='.'yyyy-MM-dd
log4j.appender.modelLogFile.layout=org.apache.log4j.PatternLayout
log4j.appender.modelLogFile.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss}
%5p - %m%n
#log4j.appender.modelLogFile.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss}
%5p [%t] (%F:%L) - %m%n
#log4j.appender.modelLogFile.layout.ConversionPattern=%p %t %c - %m%n
# Print only messages of priority WARN or higher for your category
log4j.category.modelLogFile=DEBUG, modelLogFile
log4j.additivity.modelLogFile=false

# Hibernate logging options (INFO only shows startup messages)
log4j.logger.org.hibernate=ERROR

# Log JDBC bind parameter runtime arguments
log4j.logger.org.hibernate.type=TRACE

```

4 - Na pasta /sistemasIIDH

Criar o arquivo **backupMysql.sh** com o conteúdo abaixo:

```

#!/bin/sh
mysqldump --defaults-extra-file=/sistemasIIDH/config.cnf --
socket=/opt/bitnami/mysql/tmp/mysql.sock --host=localhost --databases psychomanager --add-drop-database
> /sistemasIIDH/shared/backup/PsychoManager/bkpPsychoManager_`date
"+%Y_%m_%d_%H_%M_%S"`.sql

```

Criar o arquivo **restoreMysql.sh** com o conteúdo abaixo:

```

#!/bin/sh
mysql --user=root --password=zcxySB3Kn4jc --host=localhost < <nome arq para
restore>.sql

```

5 - Configurar crontab para bkp do banco

Editar arquivo **/etc/crontab** e adicionar a linha

```
00 01 * * * wildfly /sistemasIIDH/backupMysql.sh
```

6 - Criar o modulo do MYSQL

Criar pasta **../wildfly-10.1.0/modules/com/mysql/main**

Criar arquivo **module.xml** com o conteúdo abaixo:

```

<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.3" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.44-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
    <module name="javax.servlet.api" optional="true"/>
  </dependencies>
</module>

```

Copiar o Jar do mysql para esta pasta

7 - Criar o modulo do JASPERREORT

Criar pasta `..\wildfly-10.1.0\modules\net\jasperreports\6.1.0.RELEASE`
Criar arquivo `module.xml` com o conteúdo abaixo:

```
<?xml version="1.0" encoding="UTF-8"?>

<module xmlns="urn:jboss:module:1.3" name="net.jasperreports" slot="6.1.0.RELEASE">
  <resources>
    <resource-root path="commons-beanutils-1.9.2.jar"/>
    <resource-root path="commons-collections-3.2.1.jar"/>
    <resource-root path="commons-digester-2.1.jar"/>
    <resource-root path="commons-logging-1.2.jar"/>
    <resource-root path="groovy-all-2.0.1.jar"/>
    <resource-root path="iText-2.1.7.js2.jar"/>
    <resource-root path="jasperreports-6.1.0.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Copiar o Jar do jasperreport para esta pasta

8 - Criar o modulo do PRIMEFACES

Criar pasta `..\wildfly-10.1.0\modules\org\primefaces\6.RELEASE`
Criar arquivo `module.xml` com o conteúdo abaixo:

```
<?xml version="1.0" encoding="UTF-8"?>

<module xmlns="urn:jboss:module:1.3" name="org.primefaces" slot="6.RELEASE" >
  <resources>
    <resource-root path="all-themes-1.0.10.jar"/>
    <resource-root path="primefaces-6.0.jar"/>
  </resources>
  <dependencies>
    <module name="javax.faces.api" />
    <module name="javax.api"/>
  </dependencies>
</module>
```

Copiar o Jar do primefaces para esta pasta

9 - Criar o modulo do SPRING

Criar pasta `..\wildfly-10.1.0\modules\`
Criar arquivo `module.xml` com o conteúdo abaixo:

```
<?xml version="1.0" encoding="UTF-8"?>

<module xmlns="urn:jboss:module:1.3" name="org.primefaces" slot="6.RELEASE" >
  <resources>
    <resource-root path="all-themes-1.0.10.jar"/>
    <resource-root path="primefaces-6.0.jar"/>
  </resources>
  <dependencies>
    <module name="javax.faces.api" />
    <module name="javax.api"/>
  </dependencies>
</module>
```

Copiar o Jar do Spring para esta pasta

10 - Configurações APACHE

Adicionar o trecho abaixo no arquivo `/opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf` para redirecionar para o site e redirecionar para a aplicação

```
# Bitnami applications installed in a Virtual Host
```



```

<VirtualHost _default_:443>
  ServerAdmin localhost
  DocumentRoot "/opt/bitnami/apache2/htdocs/iidh"
  ServerName iidh.com.br

  SSLEngine on
  SSLCertificateFile "/opt/bitnami/apache2/conf/server.crt"
  SSLCertificateKeyFile "/opt/bitnami/apache2/conf/server.key"

  <Directory "/opt/bitnami/apache2/htdocs/iidh">
    Options FollowSymLinks MultiViews
    AddLanguage en en
    LanguagePriority en
    ForceLanguagePriority Prefer Fallback

    AllowOverride All
    <IfVersion < 2.3 >
      Order allow,deny
      Allow from all
    </IfVersion>
    <IfVersion >= 2.3 >
      Require all granted
    </IfVersion>
  </Directory>

  # Error Documents
  ErrorDocument 503 /503.html

  ProxyPass /PsychoManager http://localhost:8080/PsychoManager
  ProxyPassReverse /PsychoManager http://localhost:8080/PsychoManager

</VirtualHost>

<VirtualHost _default_:80>
  ServerName iidh.com.br
  # Redirect permanent / https://iidh.com.br

  RewriteEngine On
  RewriteCond %{HTTPS} !=on
  RewriteRule ^/?(.*) https://iidh.com.br/$1 [R,L]

</VirtualHost>

<VirtualHost _default_:80>
  ServerAdmin localhost
  DocumentRoot /opt/bitnami/apache2/htdocs/xfinder
  ServerName xfinderarchery.com

  <Directory "/opt/bitnami/apache2/htdocs/xfinder">
    Options FollowSymLinks MultiViews
    AddLanguage en en
    LanguagePriority en
    ForceLanguagePriority Prefer Fallback

    AllowOverride All
    <IfVersion < 2.3 >
      Order allow,deny
      Allow from all
    </IfVersion>
    <IfVersion >= 2.3 >
      Require all granted
    </IfVersion>
  </Directory>

  # Error Documents
  ErrorDocument 503 /503.html

  # ErrorLog logs/iidh.com.com-error_log

```

```
# CustomLog logs/iidh.com.-access_log common
```

```
</VirtualHost>
```

Criar a pasta **iidh** em **/opt/bitnami/apache2/htdocs**

Copiar o **site** para dentro da pasta **/opt/bitnami/apache2/htdocs/iidh**



Instalando Wordpress

Baixar a imagem

```
docker pull wordpress:latest
```

Descobrir o IP do servidor

```
ifconfig
```

Ir para a pasta /sistemasIIDH/shared e criar a pasta wp-data

```
cd /sistemasIIDH/shared  
mkdir wp-data
```

Instalar o wordpress

```
docker run --name wordpress \  
-e WORDPRESS_DB_HOST=10.128.0.2:3306 \  
-e WORDPRESS_DB_USER=root \  
-e WORDPRESS_DB_PASSWORD=zcxySB3Kn4jc \  
-e WORDPRESS_DB_NAME=wordpress \  
-p 8081:80 \  
--mount type=bind,source="$(pwd)/wp-data,target=/var/www/html \  
-d wordpress:latest
```

Verificar o log

```
docker logs -f wordpress
```

Verificar o conteúdo da pasta wp-data

```
ls -l wp-data
```

Verificar container

```
docker ps
```

Rodar o Script de tunel do wordpress

```
./tunnel-Wordpress.sh
```

Verificar se a porta 8081 esta no ar

```
sudo ss -i -P -n | grep 8081
```

Acessar a pagina

```
http://localhost:8081
```

Configurar Wordpress

The image displays three sequential screenshots of the WordPress installation interface. The first screenshot shows the language selection screen with 'Português do Brasil' highlighted. The second screenshot shows the 'Bem-vindo (a)' screen where site information is entered: 'Integração' for the title, 'lch' for the username, and a password. The third screenshot shows the 'Sucesso!' screen indicating successful installation.

Instalar o tema medical-spa

Copiar arquivos de imagem para o servidor

```
scp -i ~/Documents/Google\ Cloud/iidh-gcloud.pem * tassinari@35.193.163.21:/tmp/tassinari  
cd /sistemasIIDH/shared/wp-data/wp-content  
mkdir uploads  
mv /tmp/tassinari /sistemasIIDH/shared/wp-data/wp-content/uploads
```

Alterar owner

```
chown -R www-data:www-data /sistemasIIDH/shared/wp-data/wp-content/uploads
```

Acessar container local do banco para gerar mysqldump

```
docker exec -it mysql /bin/bash
echo "[client]
user = root
password = teresa07" > config.cnf
mysqldump --defaults-extra-file=config.cnf --host=localhost --databases wp_db --add-drop-database >
wp_db-`date +%Y_%m_%d_%H_%M_%S`.sql
```

Editar o arquivo de dump e substituir

```
Substitua                por
utf8mb4                  utf8
utf8_unicode_520_ci      utf8_general_ci
```

No arquivo wp-config.php

```
define('DB_CHARSET', 'utf8');
```

Copiar dump de banco para servidor

```
scp -i ~/Documents/Google\ Cloud/iidh-gcloud.pem wp_db-2020_08_23_15_21_58.sql
tassinari@35.193.163.21:/tmp/tassinari
```

Carregar dump

```
mysql --user=root --password=zcxysB3Kn4jc --host=localhost < wp_db-2020_08_23_15_21_58.sql
```

Atualizar a url no banco de dados

```
UPDATE `wordpress`.`wp_options` SET `option_value`='http://localhost:8081' WHERE `option_id`='1';
UPDATE `wordpress`.`wp_options` SET `option_value`='http://localhost:8081' WHERE `option_id`='2';
```

Restartar container

```
docker restart wordpress
```

Acessar a pagina

```
http://localhost:8081/wp-login.php
Atualizar banco de dados
```



Necessária a atualização do banco de dados

WordPress atualizado! Antes de prosseguir, temos que atualizar seu banco de dados para a versão mais recente.

O processo de atualização pode demorar um pouco, portanto seja paciente.

[Atualizar o banco de dados do WordPress](#)

Instalar os plugins

```
contact-form-7
WP Mail SMTP by WPForms
```

Configurar Apache para acessar o container Wordpress

Editar o arquivo `/opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf` e acrescentar o trecho abaixo em destaque para fazer redirect para o wordpress

```
<VirtualHost _default_:443>
    ServerAdmin localhost
    DocumentRoot "/opt/bitnami/apache2/htdocs/iidh"
    ServerName iidh.com.br

    SSLEngine on
    SSLCertificateFile "/opt/bitnami/apache2/conf/iidh.com.br.crt"
    SSLCertificateKeyFile "/opt/bitnami/apache2/conf/iidh.com.br.key"

    <Directory "/opt/bitnami/apache2/htdocs/iidh">
        Options FollowSymLinks MultiViews
        AddLanguage en en
        LanguagePriority en
        ForceLanguagePriority Prefer Fallback

        AllowOverride All
        <IfVersion < 2.3 >
            Order allow,deny
            Allow from all
        </IfVersion>
        <IfVersion >= 2.3 >
            Require all granted
        </IfVersion>
    </Directory>

    # Error Documents
    ErrorDocument 503 /503.html

    RequestHeader set X-Forwarded-Proto "https"
    RequestHeader set X-Forwarded-Port "443"

    ProxyPreserveHost On
    ProxyRequests Off

    ProxyPass /PsychoManager http://localhost:8080/PsychoManager
    ProxyPassReverse /PsychoManager http://localhost:8080/PsychoManager

    ProxyPass / http://localhost:8081/
    ProxyPassReverse / http://localhost:8081/

</VirtualHost>

<VirtualHost _default_:80>
    ServerName iidh.com.br

    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    # RewriteRule ^/?(.*) https://iidh.com.br/$1 [R,L]
    RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
```

Restartar o apache

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

Editar o arquivo `/sistemasIIDH/shared/wp-data/wp-config.php` e acrescentar o trecho abaixo para funcionar o redirect.

```
define( 'WP_DEBUG', false );
define( 'FORCE_SSL_ADMIN', true );
define( 'WP_HOME', 'https://iidh.com.br' );
define( 'WP_SITEURL', 'https://iidh.com.br' );
```

