# Instalación

Instalar ffmpeg, software necesario para procesar las imágenes de los avatars

Crear a base de datos con los scripts correspondientes (mysql o sqlserver)

Copiar el war al directorio de deploy de la configuración.

### Entregamos dos paquetes de instalación uno para Bases MySQL y otro para SQLSERVER

1) Mysql: mysql/LOJACK\_WEB.war

2) Sqlserver: sqlserver/LOJACK\_WEB.war

En los pasos de creación de base de datos, usuario y copiado de datasources cambiar los datos de la base y usuario de creerlo necesario.

### Crear la base de datos y el usuario:

1) Mysql: mysql/create-mysql.sql

2) Sqlserver: sqlserver/db-sqlserver.sql

### Crear los objetos con el usuario:

1) Mysql: mysql/create-mysql.sql

2) Sqlserver: sqlserver/create-sqlserver.sql

### Copiar la definición de datasource correspondiente al directorio de deploy:

1) Mysql: mysql/mysql-ds.xml

2) sqlserver: sqlserver/sqlserver-ds.xml

### Copiar el jar del driver al directorio lib de la configuración de jboss:

1) Mysql: mysql/mysql-connector-java-5.1.13-bin.jar

2) sqlserver: sqlserver/sqljdbc4.jar

## Configuración

Toda la configuración de la aplicación se realiza a través de datos de la tabla SYSPROPERTIES.

#### A continuación se listan las propiedades y su uso:

**front.server:** Se usa para generar links a la aplicación que son absolutos, debe tener el nombre con el cual el server que hostea el front se ve desde otros equipos, por ejemplo: www.lojackhome.com

**thalamus.server:** url del server de thalamus, por ejemplo http://localhost:8280/ThalamusWeb/ o https://testa-lojack-rest.thalamuslive.com/lojack

**thalamus.touchpoint.code:** código asignado para thalamus (REAL-LIFE-SMARTPHONE-APP)

**thalamus.touchpoint.token:** token asignado para thalamus

(k8exyct1v6edf9q50fryuq3r02upn0m98twst4etlbjgc9cnsq585879gxlcbu11)

**gis.server:** Ubicación del server de GIS, por ejemplo http://localhost:8180/GISWeb/gis/

**services.server:** Ubicación del server de servicios no GIS, por lo hablado debe ser el mismo.

**prevent.server:** Ubicación del server de prevent, por ejemplo http://www.lojackgis.com.ar/PreventWCFServices/GIS

### Service.svc

**prop.tmp.path:** Directorio temporal, aquí se extraerá la configuración del log4j y se escribirá el log, así como los blobs serán extraídos en dicho directorio.

**camera.mobile.refreshTime:** Tiempo de refresco de la versión Mobile de la cámara, representa la cantidad de milisegundos de refresco entre imagen e imagen.

**camera.applet.refreshTime:** Tiempo de refresco de la versión applet de la cámara, representa la cantidad de milisegundos de refresco entre imagen e imagen.

**job.refresh.time:** Tiempo de refresco del proceso del front que consulta los estados de los jobs asincrónicos. Se expresa en milisegundos.

**job.abort.time:** Tiempo pasado el cual se asume que un job no terminara. Pasado ese tiempo no se consultara más dicho job. Se expresa en milisegundos.

**job.client.refresh.time:** Tiempo de refresco del ajax del cliente que consulta los estados de los jobs. Se expresa en milisegundos.

**front.login.deplay:** Milisegundos de retraso del login.

**gui:** guid para usar en la comunicación con gis

**contactform.email:** Dirección de email a la cual se enviaran los formularios de contacto

## Mails

La configuración para envío de emails se encuentra en la tabla SYSPROPERTIES con el prefijo mail.

Se provee una configuración ejemplo que utiliza gmail para el envío (cuenta test.lojack.front@gmail.com/t3stl0j4k)

## Logging

La configuración del log de la aplicación se realiza a través de log4j. La aplicación extrae automáticamente el log4j al directorio prop.tmp.path, luego puede ser editado para configurarlo.

Si bien cada clase tiene su log, existen tres clases que pueden ser particularmente importantes y son las que acceden a servicios externos:

**com.tdil.thalamus.client.core.**

**ThalamusClient:** log de la clase que maneja la comunicación con thalamus. En modo debug muestra todas las invocaciones, incluyendo parámetros y resultados. En modo info muestra los tiempos de las invocaciones.

**com.tdil.lojack.prevent.PreventConnector:** log de la clase que maneja la comunicación con prevent. En modo debug muestra todas las invocaciones, incluyendo parámetros y resultados. En modo info muestra los tiempos de las invocaciones.

**com.tdil.lojack.gis.LoJackServicesConnector:** log de la clase que maneja la comunicación con el middleware. En modo debug muestra todas las invocaciones, incluyendo parámetros y resultados. En modo info muestra los tiempos de las invocaciones. Se recomienda quitar el uso de console appender en el ambiente productivo.