DOCUMENTO TCP COTIZANTE AMICAR V2

28 de abril de 2015

Traspaso de Componentes a Certificación / Producción.

# Proyecto: Amicar Cotizante

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Autor** | **Revisado** | **Principales Cambios** |
| 2.000 | 09-02-2015 | Diego Pinto | A Sanmartín | Versión inicial. |
| 2.001 | 20-04-2015 | Diego Pinto |  | Se modifica app LandingPage para sacar la configuración de la carpeta principal del proyecto, y se agrega carpeta “BBDD” y “Configuration”. |
| 2.002 | 23-04-2015 | Diego Pinto |  | Se modifica aplicativo “App” por llegada nueva base de datos, y se modifican componentes; “Templates Body”, “BBDD”, “Configuration” y “Servlets” |
| 2.003 | 28-04-2015 | Diego Pinto |  | Se modifican los Servlets del proyecto, por fallas en QA. |

Índice de contenido.

[Proyecto: Amicar Cotizante 2](#_Toc417308778)

[Índice de contenido. 2](#_Toc417308779)

[Descripción: (marcar según corresponda). 4](#_Toc417308780)

[Detalles: (Sección donde se debe indicar requerimiento y solución implementada). 4](#_Toc417308781)

[Diagrama de Procesos. (Solo para nuevos desarrollos o cambios en módulos BAT). 4](#_Toc417308782)

[Casos de pruebas. (Solo para certificación). 4](#_Toc417308783)

[Evidencias de pruebas. 4](#_Toc417308784)

[1. Componentes 5](#_Toc417308785)

[2. Plan de implantación 5](#_Toc417308786)

[Preparación Ambiente 5](#_Toc417308787)

[Base de datos (BBDD) 9](#_Toc417308788)

[Configuración Aplicativo 9](#_Toc417308789)

[Configurar cotizanteAmicar.properties 9](#_Toc417308790)

[Servlets “Amicar Cotizante v2” 10](#_Toc417308791)

[Lecturas 10](#_Toc417308792)

[Clicks 10](#_Toc417308793)

[Desinscritos 10](#_Toc417308794)

[Aplicativo Web “Amicar Cotizante 2.0” 11](#_Toc417308795)

[Aplicativo **[app]** 11](#_Toc417308796)

[Base de datos **[database]** 11](#_Toc417308797)

[**Archivos LOGS [logs]** 11](#_Toc417308798)

[Automatización 11](#_Toc417308799)

[Reporte Mensual y Reporte Diario 11](#_Toc417308800)

[Aplicativos y estructura de trabajo 12](#_Toc417308801)

[Configuración del batch reporte mensual y reporte diario. 13](#_Toc417308802)

[Configuración del batch Amicar Cliente y Amicar Ejecutivo 14](#_Toc417308803)

[1.-AMICAR-CLIENTE.BAT 14](#_Toc417308804)

[1.-AMICAR-EJECUTIVO.BAT 15](#_Toc417308805)

[TaskEjecutivo.BAT 15](#_Toc417308806)

[Anexos 16](#_Toc417308807)

[Aplicaciones JAVA 16](#_Toc417308808)

[ArchivosAmicarFTP.jar 16](#_Toc417308809)

[Fecha 16](#_Toc417308810)

[ArchivoDiario 16](#_Toc417308811)

[AmicarAddFixBody 17](#_Toc417308812)

[CargaDatosAmicar2 17](#_Toc417308813)

[SvlAmicarClicks.war 18](#_Toc417308814)

[SvlAmicarLecturas.war 18](#_Toc417308815)

[SvlAmicarDesinscritos.war 18](#_Toc417308816)

[Bitácora. (Exclusivo operaciones) 19](#_Toc417308817)

# Descripción: (marcar según corresponda).

|  |  |  |
| --- | --- | --- |
| **Tipo de Traspaso.** | **Estado** | |
| **Certificación** | **Producción** |
| **Desarrollo.** |  |  |
| **Mantención.** |  |  |
| **Contingencia.** |  |  |

|  |  |
| --- | --- |
| **Sistema Afectado** | |
| **Amicar Cotizante v2** |  |
|  |  |

## Detalles: (Sección donde se debe indicar requerimiento y solución implementada).

Se genera documento TCP para llevar a cabo la implantación del aplicativo “Amicar Cotizante v2” para Amicar, donde queda detallado cada componente y pasos a seguir para lograr levantar el ambiente necesario de manera correcta. Diagramas de flujo. (Solo para nuevos desarrollos o cambios en módulos BAT).

## Diagrama de Procesos. (Solo para nuevos desarrollos o cambios en módulos BAT).

## Casos de pruebas. (Solo para certificación).

## Evidencias de pruebas.

1. Componentes**.**

|  |  |  |
| --- | --- | --- |
| **Nombre Modulo.** | **Tipo** (java, bat, tabla, etc.) | **Destino en producción** (path, ruta, etc.) |
| **/Servlet** | **War** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Plan de implantación**.** (instrucciones para instalación en cert. o prod.)

## Preparación Ambiente

Estos son los pasos básicos para empezar con un proyecto “Laravel” en Ubuntu. (Este documento se basa en Laravel 4.2 y Ubuntu 14.04, que han sido actualizadas en este escrito.)

En primer lugar, para realizar este proceso usted necesita crear un ambiente LAMP (Abreviatura de "Linux-Apache-MySQL-PHP”), incluyendo los [enrollamiento](http://curl.haxx.se/) utilidades, [Git](http://git-scm.com/) control de versiones, y [OpenSSL](http://www.openssl.org/) , instalados en Ubuntu. Para ello, ejecute este comando en la terminal, que instalará automáticamente todas las dependencias.

sudo apt-get -y instalar php5 mysql-server phpmyadmin mysql-client php5-rizo openssl git rizo

Durante el proceso de instalación, se le pedirá que especifique una contraseña de la cuenta **ROOT** de MySQL. Después de que el proceso de instalación, abra el navegador web e introduzca <http://localhost/> en la barra de direcciones. Usted debe ver "It’s Work!" como página por defecto.

El [Composer](https://getcomposer.org/) es uno herramienta de gestión de la dependencia que se basa en “Laravel” requiere que se habilite el módulo de extensión de [PHP Mcrypt](http://php.net/manual/en/book.mcrypt.php) . El proceso de instalación en el paso 1 anterior será instalarlo (el paquete llamado php5-mcrypt ). Sin embargo, no se puede activar. Para habilitarlo, ejecute este comando en el terminal.

sudo php5enmod mcrypt

Ahora [reinicie el servidor web Apache](http://www.cyberciti.biz/faq/ubuntu-linux-start-restart-stop-apache-web-server/) para cargar el mcrypt módulo:

sudo service apache2 restart

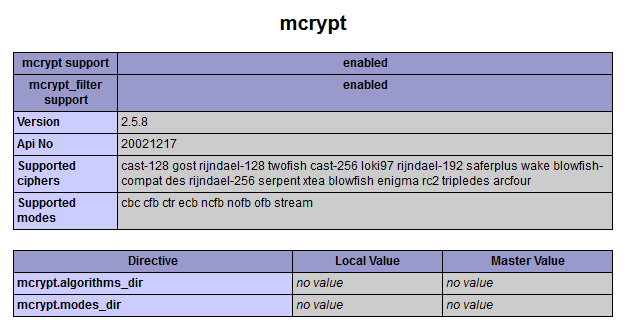
Para confirmar que el mcrypt módulo cargado, crear un pequeño script PHP en un editor de texto que contenga:

<?php

echo phpinfo();

?>

Guarde este archivo en la carpeta raíz para el servidor Web Apache /var/www/html y con el nombre de phpinfo.php. Abra su navegador Web y vaya a <http://localhost/phpinfo.php> . El navegador mostrará un largo informe que muestra los detalles de la configuración de PHP en su sistema, incluyendo una sección *parecida* a esta:



**Nota:** Como alternativa, puede utilizar la línea de comandos PHP -m [opción](http://www.php.net/manual/en/features.commandline.options.php) para comprobar si mcrypt está habilitado o no:

php -m | grep mcrypt

Si mcrypt está habilitado, este comando debe mostrar una línea con mcrypt en él.

Es posible que haya observado que usted necesita raíz permisos para guardar el phpinfo.php archivo en el directorio /var/www. Esto es un poco incómodo y relativamente puede arreglar esto fácilmente. En lugar de utilizar el directorio del servidor web Apache por defecto de /var/www, crearemos un llamado [Host virtual basado en nombre](https://httpd.apache.org/docs/2.2/vhosts/name-based.html) que hace referencia a nuestro proyecto.

Ahora, tenemos que crear un nuevo archivo de Apache "sitio" para nuestro host virtual.

sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/{proyecto}.conf

**Nota**: Si nuestro proyecto se llama “LANDING”, colocaremos en cada parte donde dice {proyecto} “landing.local” para saber que se encuentra en nuestra máquina.

Por lo general, tendrá que dar a su archivo "sitio", un nombre similar al nombre de dominio que va a utilizar. En este caso, vamos a crear un nuevo nombre de dominio local de {proyecto}. Abra la nueva {proyecto}.conf con un editor de texto. Recuerde que al tratarse de una modificación, debe usar sudo para iniciar el editor de texto. Por ejemplo:

sudo [vim/nano/pico/gedit/...] /etc/apache2/sites-available/{proyecto}.conf

Reemplazar el contenido del archivo con este.

<VirtualHost {proyecto}>

ServerName {proyecto}

ServerAlias {proyecto}

DocumentRoot /var/www/{proyecto}/public/

<Directory /var/www/{proyecto}/public/>

Options Indexes FollowSymLinks MultiViews

AllowOverride All

Allow from all

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</ VirtualHost>

Asegúrese de que reemplaza nombre del proyecto en /var/www/{proyecto} con su nombre de proyecto como se menciona anteriormente. Además, asegúrese de que el nombre de la ruta en la Directiva tiene un '/' al final. Guarde el archivo y cierre el editor de texto.

Tenemos que permitir que esta nueva configuración "sitio" y reiniciar el servidor web Apache para activarlo.

sudo a2ensite {proyecto}.conf

sudo service apache2 restart

Edita tu /etc/hosts y agregue una línea que contiene lo siguiente:

127.0.0.1  {proyecto}

Esto le permite utilizar el host (dominio) nombre de alias {proyecto} en sus URL del navegador web. Para probar esto, abra el navegador web e introduzca [http: //{proyecto}/](http://projects.local/) en la barra de direcciones. Usted debe ver el estándar Apache "It’s Work!" o su similitud.

Luego de todo esto creamos o movemos nuestro proyecto a la carpeta que

Ahora estamos listos para instalar la herramienta [Composer](https://getcomposer.org/) para la gestión de la dependencia. En el terminal, escriba los siguientes comandos:

cd /var/www/{proyecto}

mkdir temp

cd temp

sudo curl -sS https://getcomposer.org/installer | sudo php

Para simplificar el uso de Compositor, podemos hacer en disponible sin importar el directorio que estamos en copiándolo a un directorio en el entorno $PATH. Para ello, ejecute:

sudo mv composer.phar /usr/local/bin/composer

Se dará cuenta de que hemos eliminado el phar extensión de archivo para mayor comodidad. Ahora, para ejecutar Composer, sólo tenemos que entrar ingresando el comando “composer”. En muchos casos, usted verá ejemplos en línea que utilizan la sintaxis **php composer.phar;** simplemente puede reemplazar eso con el **composer**. (También puede eliminar la temp directorio que ha creado, ya que sólo se necesitaba como un lugar temporal para instalar Compositor.)

Para confirmar que todo está funcionando, basta con ejecutar el composer en la terminal por sí mismo. Debería ver algo como esto:

   \_\_\_\_\_\_

  / \_\_\_\_/\_\_\_  \_\_\_\_ \_\_\_  \_\_\_\_  \_\_\_\_  \_\_\_\_\_\_\_\_  \_\_\_\_\_

 / /   / \_\_ \/ \_\_ `\_\_ \/ \_\_ \/ \_\_ \/ \_\_\_/ \_ \/ \_\_\_/

/ /\_\_\_/ /\_/ / / / / / / /\_/ / /\_/ (\_\_  )  \_\_/ /

\\_\_\_\_/\\_\_\_\_/\_/ /\_/ /\_/ .\_\_\_/\\_\_\_\_/\_\_\_\_/\\_\_\_/\_/

                    /\_/

Composer version \*\*\*\*\*

Usage:

  [options] command [arguments]

Luego de esto debemos dar permisos a nuestra aplicación. Lo que debemos hacer es configurar los permisos del {proyecto}:

cd /var/www

chmod 775 -R {proyecto}

cd {proyecto}/app

chmod 777 -R storage

Estamos listos para probar la instalación! Abra su navegador Web y escriba http://{proyecto}/ en la barra de direcciones. Si todo está configurado correctamente, debería ver la página “Amicar Cotizante 2015.

## Base de datos (BBDD)

Cargar el script que se encuentra alojado en el repositorio SVN dentro de la carpeta BBDD.

## Configuración Aplicativo

Dentro de la carpeta principal del proyecto en SVN existe una carpeta de nombre “Configuration”, en la cual se encuentran dos archivos de configuración uno para java (.properties) y otro para php (.ini), ambos se deben copiar a la ruta “/apps/Amicar/config/” o ruta de configuración predeterminada por QA, para luego ser modificados dependiendo de los parámetros que se necesiten.

## Configurar cotizanteAmicar.properties

1. Se debe crear la siguiente ruta: “C:\apps\Amicar\config\” ***debes respetar mayúsculas y minúsculas***
2. En la carpeta config se debe almacenar el siguiente archivo, cotizanteAmicar.properties, este tiene la siguiente información:

* javax.persistence.jdbc.url=jdbc:mysql://localhost1:3306/amicarcotizante
* javax.persistence.jdbc.user=USER
* javax.persistence.jdbc.password=PASSWORD
* salidaemmesaging=C:/clients/Amicar/DATA/
* dominioclick=http://localhostSolicitudCotizacion:8080/AmicarSvl/Clicks
* dominiolectura=http://localhostServler:8080/ AmicarSvl / Lecturas dominiodesinscrito=http:// localhostDesinscritos:8080/AmicarSvl/Desinscritos
* dominiolanding=http://localhostLanding:8000/Amicar
* logs.path=C:/Logs/

1. Aquí deben cambiar los datos resaltados en rojo, tal como se indica a continuación:

* *Localhost*: IP del servidor donde se encuentra alojada la base de datos.
* *amicarcotizante:* nombre de la base de datos correspondiente
* *user:* nombre del usuario de la base de datos
* *pass:* contraseña del usuario que se conecta a la base de datos
* *C:/clients/Amicar/DATA/* ruta donde de salida para emessaging.
* *localhostSolicitudCotizacion:* Servidor donde se encuentra alojada la aplicación Solicitud Cotización Amicar.
* *localhostServler:* Servido donde se encuentra alojada el servlet de lecturas de Amicar
* *localhostDesinscritos:* Servido donde se encuentra alojada el servlet que registra los usuarios que no desean recibir más correos de Amicar
* *localhostLanding:* Servido donde se encuentra alojada la aplicación web de Amicar realizada en PHP.
* *logs.*path: Directorio donde se requiere dejar los archivos logs generados por el aplicativo.

## Servlets “Amicar Cotizante v2”

El proyecto se compone de 3 servlets realizados en Java y compilado en formato WAR alojados dentro de un mismo componente, estos son:

### Lecturas

Servlet para realizar el registro de la fecha y hora de lectura del correo recibido por el cliente.

### Clicks

Servlet para capturar el click dentro de correo, generar archivo ejecutivo para envió de solicitud y redirección hacia la landing.

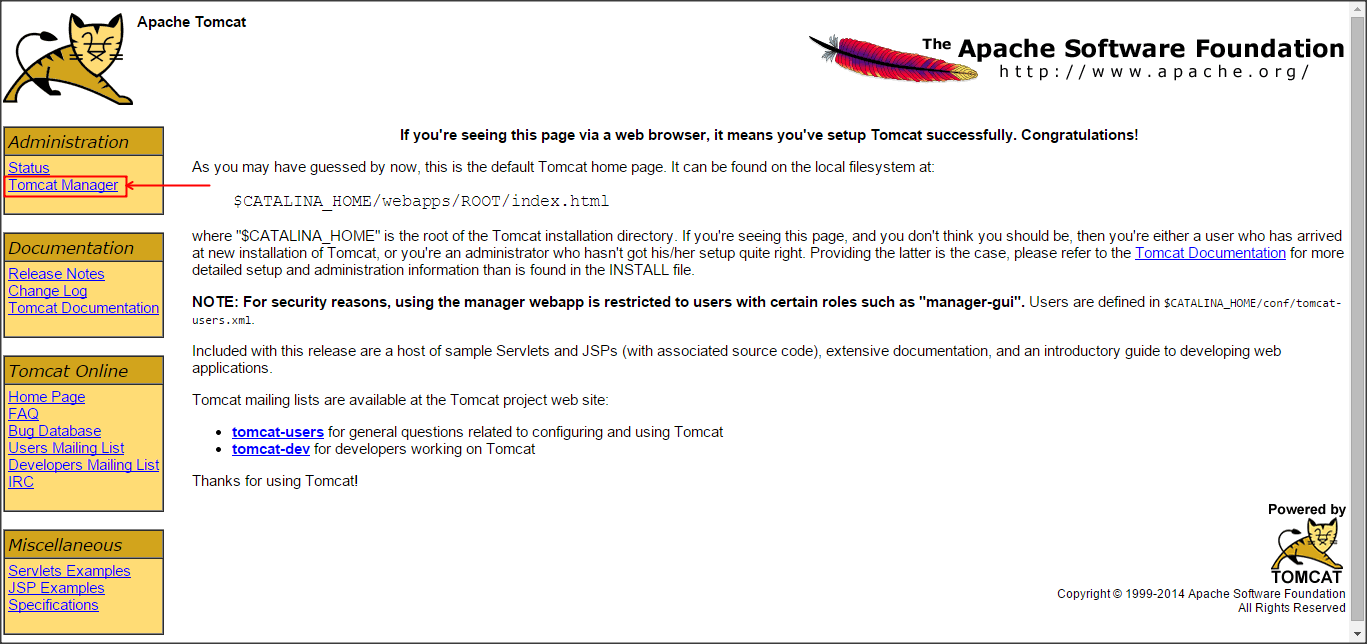
### Desinscritos

Servlet para que modifica un registro del cliente para que este no reciba correos a futuro y redirección hacia la landing.

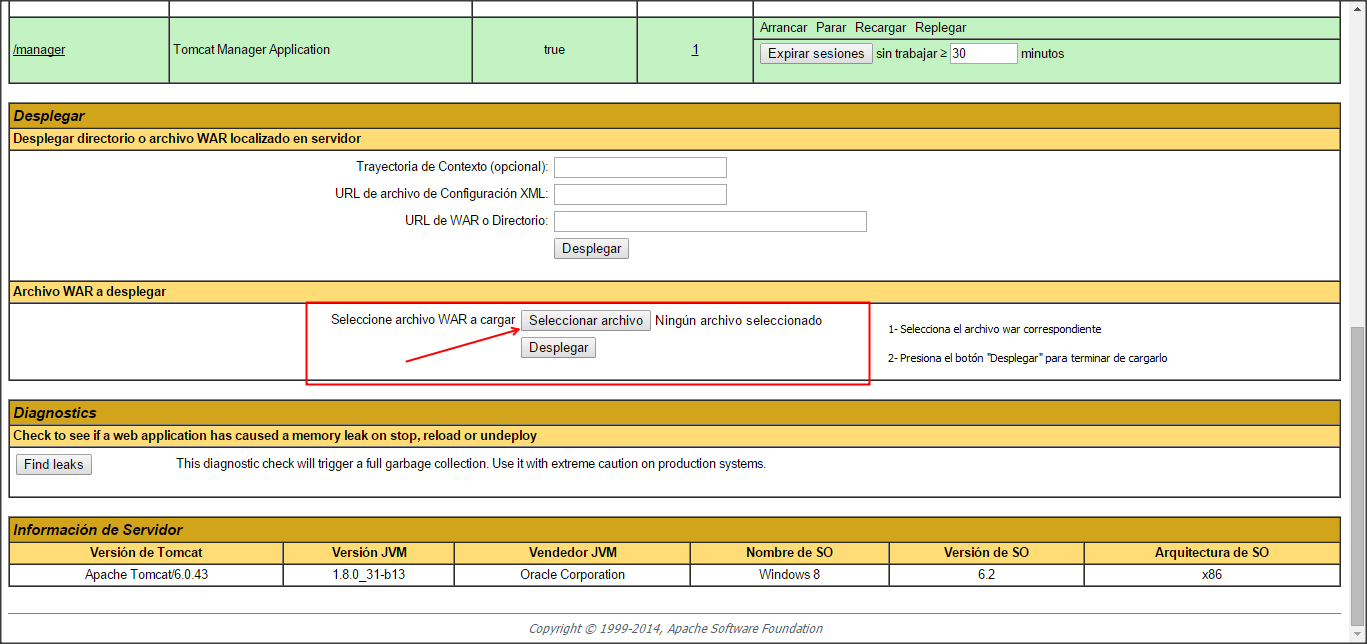
Estos tres compontes deben ser incluidos dentro de Tomcat con la ayuda del administrador. Para llevar a cabo esto debemos seguir los siguientes pasos.

Lo primero que debemos realizar es ingresar a Tomcat a través de un navegador colocando como url **“http://localhost:8080”,** la dirección ip puede variar según donde esté instalado.

Una vez cargada la página nos aparecerá la siguiente pantalla en donde haremos click en **“Tomcat Manager”** tal como lo indica la imagen.



Al ingresar al manger, apreciaremos que casi al final de la página el sector de **“Archivo WAR a desplegar”**, es aquí donde se realiza la carga seleccionando uno de los compontes war descritos al inicio y luego el otro sucesivamente hasta cargarlos todos.



Luego de todo esto estamos listos para probar el ambiente.

## Aplicativo Web “Amicar Cotizante 2.0”

### Aplicativo [app]

Para visualizar una barra de ayuda y los errores del aplicativo, debug=false

Url donde quedará el aplicativo, url=http://localhost/

### Base de datos [database]

Dirección de la BBDD, host=192.168.1.99

Nombre de la BBDD, dbname=amicarcotizante

Usuario BBDD, username=root

Contraseña usuario, password=inteladmin

### Archivos LOGS [logs]

Dirección donde se desean dejar los archivos “log”, path=C:/apps/Amicar/logs/

## Automatización

Este aplicativo cuenta con un sistema integrado de envió de correos y descarga de archivos, que deben ser ejecutados mensual y diariamente, cada uno de ellos se encarga de procesar un Reporte mensual y uno diario alojados en un FTP del cliente Amicar.

Para la configuración de estos procesos que corren bajo un ambiente Windows deben seguir los pasos descritos a continuación.

### Reporte Mensual y Reporte Diario

Es una aplicación de procesamiento por lotes (Batch o bat) de Windows, que se conecta a un ftp, descarga una lista de los archivos disponibles, ejecuta un aplicativo java que se encarga de buscar el ultimo archivo vigente según lo solicitado por el cliente, para que este archivo sea descargado, finalmente con el archivo en el equipo, ejecuta otro aplicativo java que se encarga de leer, validar y cargar los datos en una base de datos.

A continuación se desglosa y se explica en detalle los componentes importantes del bat y cuáles deben ser modificados por el explotador.

#### Aplicativos y estructura de trabajo

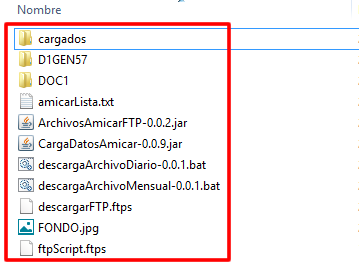
La estructura de trabajo y los archivos necesarios para ejecutar ambos procesos (descargaArchivoDiario-0.0.2.bat y descargaArchivoMensual-0.0.2.bat), se compone de la siguiente manera:

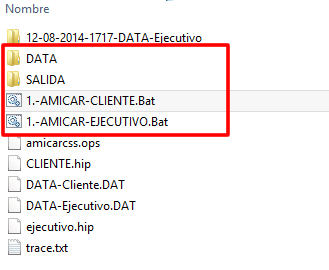
***Contenido Carpeta Raíz (App):***

* **ArchivosAmicarFTP-0.0.2.jar:** Contiene Tres aplicaciones integradas siendo estas Fecha, ArchivoDiario y FixHtmlAmicar, que devuelven la fecha del último archivo vigente, el nombre del archivo a descargar si este existe, y parcha los HTML generados, respectivamente (para mayor información revisar la sección anexo-> aplicaciones.).
* **CargaDatosAmicar2-0.0.1**.jar**:** Toma un archivo entregado por parámetros, lo lee, procesa y carga sus datos en la base de datos *“amicarcotizante”*, esta se encarga de procesar los archivos mensuales y diarios.
* **descargaArchivoDiario-0.0.2.bat:** Proceso que se encarga de descargar y coordinar el procesamiento del archivo diario.
* **descargaArchivoMensual-0.0.2.bat:** Proceso que se encarga de descargar y coordinar el procesamiento del archivo mensual.
* **ftpScript.ftps:** archivo que contiene el script necesario para conectarse al servidor ftp del cliente y descargar la lista de los archivos contenidos en él.
* **amicarLista.txt:** archivo generado por ambos procesos al momento de ejecutar el script del ftp, este contiene la lista de los archivos contenidos en el servidor ftp.
* **descargarFTP.ftps:** archivo generado por ambos procesos, este contiene un script dinámico que indica que archivo debe descargarse del servidor ftp.
* **Cargados:** Carpeta que contiene los archivos que se han descargado y procesado.
* **FONDO.jpg:** es una imagen del sistema necesaria para los envíos de correo
* **DOC1:** Contiene los bat necesarios para leer la data de salida con los clientes y genera los archivos html a procesar.
* **D1GEN57:** Contiene los aplicativos de DOC1.

***Contenido Carpeta DOC1:***

* **DATA:** Carpeta de salida donde se aloja la data generada para emitir correos de clientes.
* **SALIDA:** Contiene las carpetas y archivos con la data procesada.
* **1.-AMICAR-CLIENTE.BAT:** Toma la data de los clientes y genera los html para el envío de correos.
* **1.-AMICAR-EJECUTIVO.BAT:** Toma la data de los ejecutivos y genera los html para el envío de correos.
* **TaskEjecutivo.bat:** Proceso encargado de ejecutar cada 5 minutos durante las 24 horas del día el proceso 1.-AMICAR-EJECUTIVO.BAT.



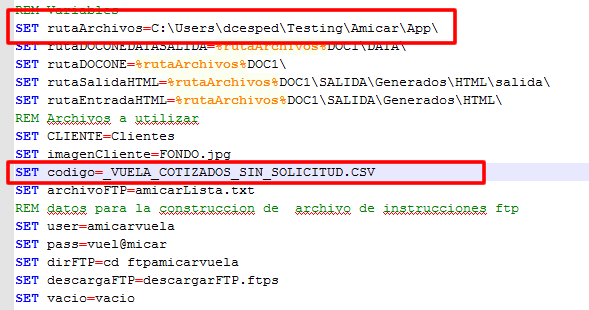


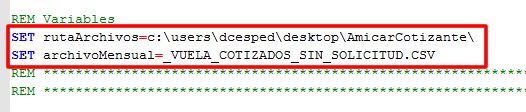
#### Configuración del batch reporte mensual y reporte diario.

La variable “rutaArchivos” contiene la ruta de la raíz (ruta donde se encuentra la carpeta app) que implementan las aplicaciones, para las aplicaciones java.

La variable “código” indica sufijo del nombre del archivo a descargar, a este se le agregara al inicio la fecha del archivo a descargar, este nombre debe ser cambiado según corresponda, se debe agregar el nombre del archivo mensual a leer si es el aplicativo mensual o se debe agregar el nombre del archivo diario a leer si es el aplicativo diario.

Para el archivo reporte mensual la variable “archivoMensual” debe llevar el nombre del archivo mensual a procesar.





Con todo esto solo resta ubicar las aplicaciones en una ruta específica y agregar a las tareas programadas de Windows los procesos ***descargaArchivoDiario-0.0.2.bat*** y ***descargaArchivoMensual-0.0.2.bat***, para su ejecución diaria y mensual respectivamente.

### Configuración del batch Amicar Cliente y Amicar Ejecutivo

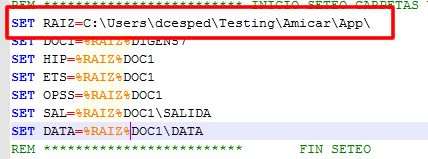
Para generar los html que serán enviados a los clientes y ejecutivos, deben ejecutarse dos procesos diferentes, pero en contenido, ambos bastante similares.

Cada uno de los procesos tiene una configuración propia, esta se describe de la siguiente manera:

#### 1.-AMICAR-CLIENTE.BAT

Este proceso se encarga de procesar la data de los clientes y generar en base a esta los archivos HTML, este debe ejecutarse una vez al día entre las 12:00 y las 15:00 horas de cada día. Los parámetros de configuración a modificar son:

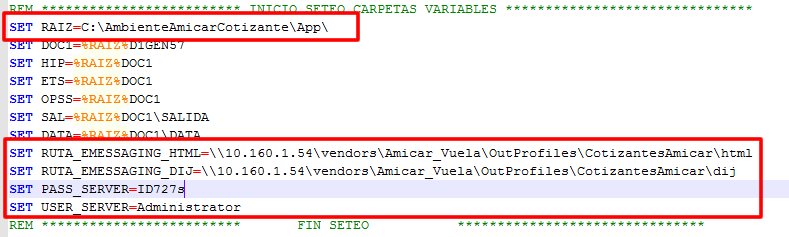
* **RAIZ:** Indica la ruta hasta el directorio App, definido como la raíz de estos procesos o ambiente.



#### 1.-AMICAR-EJECUTIVO.BAT

Este proceso se encarga de tomar la data de ejecutivos generada por el aplicativo SolicitudCotizacionAmicar.war, y generar los HTML correspondientes y los mueve a las dependencias de emessaging, para luego ser enviados por email. Los parámetros a configurar son:

* **RAIZ:** Indica la ruta hasta el directorio App, definido como la raíz de estos procesos o ambiente.
* **RUTA\_EMESSAGING\_HTML**: indica la ruta de la máquina y el profile donde copiara los archivos HTML generados.
* **RUTA\_EMESSAGING\_DIJ:** Indica la ruta de la máquina y el profile donde copiara los DIJ generados.
* **PASS\_SERVER:** Contraseña correspondiente al servidor donde se aloja el profile de emessaging.
* **USER\_SERVER:** Nombre de usuario correspondiente al servidor donde se aloja el profile de emessaging.



#### TaskEjecutivo.BAT

Proceso encargado de llamar a “1.-AMICAR-EJECUTIVO.BAT”, este debe ser programado para ejecutarse cada 5 minutos, dentro de las 24 horas de cada día, a fin de automatizar él envió de correos de los ejecutivos.

# Anexos

## Aplicaciones JAVA

Con el fin de informar y documentar se explican en detalle cada una de las aplicaciones.

### ArchivosAmicarFTP.jar

En su versión 0.0.2 este jar contiene tres aplicativos siendo estos FixHtmlAmicar, Fecha y ArchivoDiario, para acceder y ejecutar cualquiera de ellas debes tipear la siguiente sentencia en el CMD de Windows o Terminal de Unix/Linux.

java -cp ArchivosAmicarFTP-0.0.1.jar cl.intelidata.amicar.main.Aplicación Parámetros

Dónde:

* Aplicación: indica el nombre de cualquiera de las aplicaciones mencionadas anteriormente (Fecha o ArchivoDiario).
* Parámetros: Parámetros de la aplicación según sean requeridos o necesarios.

#### Fecha

Debe ser llamada desde línea de comandos de la siguiente manera:

java -cp ArchivosAmicarFTP-0.0.1.jar cl.intelidata.amicar.main.Fecha

Este imprime en pantalla una fecha en formato *“AAMMDD”* ejemplo *“140819”* la fechaque retorna corresponde al día hábil anterior al actual, por ejemplo:

* Si la fecha actual fuese Martes 19-08-2014 la aplicación imprimirá el día hábil anterior, siendo este el Lunes 18-08-2014, pero en formato de salida “AAMMDD” siendo “140818”.
* Si la fecha actual fuese Lunes 11-08-2014 la aplicación imprimirá el día hábil anterior siendo este el Viernes 08-08-2014, pero en formato de salida “AAMMDD” siendo el resultado “140808”.

#### ArchivoDiario

Debe ser llamada desde línea de comandos de la siguiente manera:

java -cp ArchivosAmicarFTP-0.0.1.jar cl.intelidata.amicar.main.ArchivoDiario <RutaDelArchivoaLeer> <NombreDelArchivoALeer> <NombreDeLaLineaABuscar>

Recibe tres parámetros por referencia siendo:

* RutaDelArchivoaLeer: Define la ruta de directorios que contienen el archivo a leer. Ejemplo “C:/usuarios/admin/carpetaContenedora/”.
* NombreDelArchivoALeer: Nombre del archivo que debe ser leído. Ejemplo “lista.txt”
* NombreDeLaLineaABuscar: Línea o valor a buscar dentro del archivo.

De manera simple este aplicativo se encarga de verificar la existencia de una línea o cadena de caracteres (String) dentro del archivo indicado, si encuentra alguna coincidencia, imprime en pantalla la línea hallada, en caso contrario imprime la palabra *“vacio”*.

### AmicarAddFixBody

Debe ser llamada desde línea de comandos de la siguiente manera:

java -cp AmicarAddFixBody-1.0.jar cl.intelidata.amicar.main.FixHtmlAmicar <RutaEntradaDeLosArchivos> <RutaSalidaDeLosArchivos> <NombreImagenAInsertar>

Recibe tres parámetros por referencia siendo:

* RutaEntradaDeLosArchivos: Define la ruta de directorios que contienen los HTML a leer. Ejemplo “C:/usuarios/admin/carpetaContenedoraHTML/”.
* RutaSalidaDeLosArchivos: Ruta donde se almacenaran los archivos HTML procesados.
* NombreImagenAInsertar: Nombre de la imagen que se insertara en el HTML, ej: “Imagen.jpg”.

Básicamente lo que realiza este aplicativo es modificar e insertar una imagen en un html dado, esta toma un link de la salida HTML de doc1 le inserta la imagen dejándola como botón del link, luego agrega una imagen servida desde un servlet especifico.

Al momento de leer, toma todos los archivos HTML existentes en la ruta de entrada, los modifica y los inserta en la ruta de salida indicada, la imagen debe tener el nombre de la imagen a insertar, esta también debe estar en la carpeta de entrada.

**CargaDatosAmicar2-0.0.1.jar**

En su versión más reciente 0.0.4 cuenta con una clase principal llamada “*CargaDatosAmicar*” para la llamada al aplicativo se debe abrir un Terminal de línea de comandos (CMD en Windows y Terminal en Unix/Linux). Y tipear la siguiente sentencia:

java -cp CargaDatosAmicar2-0.0.1.jar cl.intelidata.amicar.CargaDatosAmicar Parámetros

### CargaDatosAmicar2

Debe ser llamada desde línea de comandos de la siguiente manera:

java -cp CargaDatosAmicar2-0.0.1.jar cl.intelidata.amicar.CargaDatosAmicar <RutaDeEntradaDeArchivos> <RutaDeSalidaDeArchivos> <NombreDelArchivo>

Recibe tres parámetros por referencia siendo:

* RutaDeEntradaDeArchivos: Define la ruta de directorios que contienen el archivo a leer. Ejemplo “C:/usuarios/admin/carpetaContenedora/”.
* RutaDeSalidaDeArchivos: Define la ruta de directorios donde se alojaran los archivos que la aplicación generara. Ejemplo “C:/usuarios/admin/carpetaContenedora/”.
* NombreDelArchivo: Nombre del archivo a leer.

El fin de la aplicación es leer y procesar dos tipos de archivos diferentes, uno el archivo diario de Amicar y el otro el archivo mensual de Amicar, al momento de procesar estos archivos selecciona los datos válidos y los guarda en la base de datos “*amicarcotizante*”.

Al procesar el archivo mensual, obtiene los datos de los ejecutivos y sus respectivos locales vigentes durante el mes que se realice el proceso.

Al procesar el archivo diario, obtiene los datos de los vendedores y los clientes ligados a un respectivo local, estos datos son almacenado en la base de datos y agregados a un proceso, en base a este proceso se ejecuta la campaña de envió de correos, este aplicativo en base a dicho proceso genera un archivo de salida que contiene la data necesaria para ser procesada por Doc1 y luego para ser enviada a los clientes por email mediante la herramienta emessaging.

Los formatos de los archivos son los siguientes:

* **Archivo mensual**: “LOCAL|EJECUTIVO|CORREO”
* **Archivo diario:** “RUT\_CLIENTE|EMAIL\_CLIENTE|NOMBRE\_LOCAL|NOMBRE\_VENDEDOR| RUT\_VENDEDOR”

### SvlAmicarClicks.war

Aplicativo web que debe ser montado en un servidor tomcat7 (detallado en las secciones anteriores), contiene un Servlet que recibe por parámetros mediante su URL <http://d34dp0o1:8080/SvlAmicarClicks/Clicks?cotizacion=26311asd1sd13&cliente=1asd9121dctjm01>, la información correspondiente la cotización y el cliente ligado a ella, una vez procesado se genera un archivo que contiene la data necesaria para ser procesada por Doc1 y luego ser enviada por email a los ejecutivos de cada sucursal.

### SvlAmicarLecturas.war

Es un servlet que se encarga de registrar las lecturas de los correos, la manera de utilización es agregando la ruta del servlet con sus respectivos parámetros, como una imagen dentro del código HTML mediante el tag <img src=”RutaServletDeLectura”>, a su vez este sirve una imagen en blanco.

### SvlAmicarDesinscritos.war

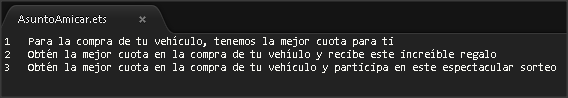
Es un servlet que se encarga de registrar los usuarios que desean no recibir más cotizaciones, ofertas, etc, desde Amicar, por lo que deja un “flag” dentro del registro de cada cliente, el que a su vez servirá para validar la carga de datos y por lo tanto el envío de correos.

### Asuntos Correos Cliente

Este proyecto consiste en tres campañas que se enviaran via correo electrónico a los clientes, cada uno de estos mensajes tiene un asunto o subject asociado, los cuales se configuran en el archivo **“AsuntoAmicar.ets”**, ubicado en la carpeta App del repositorio correspondiente a este proyecto.+

El archivo se visualiza de la siguiente manera.

# 



# Bitácora. (Exclusivo operaciones)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ambiente Certificación** | | | | | | | | | | |
| **Responsable Soporte.** | **Hr. Inicio.** | | **Hr. Termino.** | | **Actividad.** | | | | | |
|  |  | |  | |  | | | | | |
| **Ambiente Producción** | | | | | | | | | | |
| **Responsable Soporte.** | | **Hr. Inicio.** | | **Hr. Termino.** | **Actividad** | **101** | **102** | **205** | **232** | **202** |
|  | |  | |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |
|  | |  | |  |  |  |  |  |  |  |