

Trabajo Fin de Grado

Jaime Viscarret Aranzadi

Tutor: Mikel Izal Azcárate

Pamplona, 7 de Septimebre de 2015

E.T.S. de Ingeniería Industrial, Informática y de Telecomunicación



Backend sistema de detección de meteoritos por radio

Grado en Ingeniería en Tecnologías de Telecomunicación

# RESUMEN

En este proyecto se pretende hacer conjuntamente con otras personas un sistema de detección de meteoritos por radio. Concretamente en esta parte se va realizar la creación de una base de datos y un backend para poder administrar y observar los datos obtenidos por las antenas.

Además se creara un script que se ejecute en la máquina cliente periódicamente con el fin de enviar los datos a la nube (servidor), la cual debe ser configurada también anteriormente.

En este proyecto se pretendía hacer conjuntamente con otras personas un sistema de detección de meteoritos por radio. La parte correspondiente a este trabajo consiste en la administración y observación de los datos obtenidos por antenas, las cuales detectan la aparición de un meteorito en la atmosfera debido a cambios en la frecuencia. Además se ha creado un script en la maquina cliente, con el fin de subir periódicamente los datos a la nube (servidor). Este script se ejecuta cuando existen datos nuevos, y establece una conexión con un php alojado en el servidor de modo seguro (HTTPS), y este php es el encargado de subirlos directamente a la base de datos.

ÍNDICE

[RESUMEN 2](#_Toc428286821)

[INTRODUCCIÓN 4](#_Toc428286822)

[OBJETIVOS 6](#_Toc428286823)

[ESTADO DEL ARTE 7](#_Toc428286824)

[METODOLOGÍA 10](#_Toc428286825)

**INTRODUCCIÓN**

Desde hace muchos años se ha estudiado el paso de meteoroides cercanos a la Tierra, debido a la preocupación de las personas por posibles catástrofes o simplemente por el hecho de conocer más a fondo todo que nos rodea y nos afecta. En muchos lugares del mundo se están estudiando técnicas para poder detectar cuando estos meteoroides consiguen traspasar la atmosfera terrestre, este trabajo se centra en la detección de meteoritos cuando han conseguido llegar a la atmosfera debido al calentamiento de esta y efectos que se producen y pueden ser captados.

[cita] Los meteoroides son restos pequeños de roca y metal que pueden proceder de asteroides, cometas y raramente de la luna de marte. Muchos de ellos son atraídos por la gravedad terrestre. La gran mayoría de ellos se vaporizan al llegar a nuestra atmosfera, dejando un rastro visible de polvo brillante al que popularmente se le atribuye el nombre de “estrella fugaz”. Entonces los meteoritos son meteoroides que han conseguida alcanzar la atmosfera terrestre y llegan a la Tierra.

Concretamente este trabajo se va a centrar en los meteoritos. Se explica brevemente a continuación el proceso que se produce al entrar en la atmosfera.

[meteor.fr]Cuando un meteoroide entra en la atmosfera, la estela luminosa que se forma se debe principalmente a la vaporización y ionización del cuerpo en el aire por la compresión de este. Al contacto con la capa de la atmosfera terrestre la energía cinética del cuerpo provoca el calentamiento y ionización (plasma), la cual se convierte en el haz de luz que logramos ver. Eso se produce a una altura entre 80 y 120 km de altitud a una velocidad entre 11 y 72 km/s. El tamaño de estos meteoritos no llega a superar 1g. Las propiedades de este plasma son las que condicionan su capacidad para reflejar una onda.

El método para poder detectar estos meteoritos consiste en enviar una onda desde una antena, y debido a este plasma que refleja la onda incidente, poder ver en el receptor la onda reflejada.

Al ser un proyecto conjunto, otro departamento es el encargado de diseñar el sistema de antenas y detección. Este proyecto se encarga en el envío y administración de estos datos que han sido obtenidos y enviados a una maquina cliente (un puerto serie). A partir de ahí son analizados por un software libre que se ha obtenido en radio.meteor.free, llamado Colorgramme lab. Este proporciona unos archivos de texto de salida, los cuales se leen y envían automáticamente a la base de datos mediante un script que se ha realizado.

Además se tiene acceso en modo administrador a todos los datos enviados, así como unas gráficas que aparecen de forma que puedan ser observadas por cualquiera que visite la web.

# OBJETIVOS

El objetivo de este trabajo es el de crear un backend para administrar los datos que son recogidos y enviados desde el cliente. En él se podrá acceder a toda la información, incluyendo lugar desde donde ha sido recogida la muestra, tipo de antena, polarización, días del mes y horas del día con sus correspondientes datos.

Para ello también es necesario crear un script que corra en una maquina cliente, cual debe estar conectada el tiempo que desee el que realice las muestras, podría ser una Raspberry en un futuro. También es necesario alojar en el servidor un php que se encargue de recoger los datos que han sido enviados a través del script, y desde ese mismo php son enviados a la base de datos de forma directa.

Para realizar el proceso de envío de datos de cliente – servidor es necesario que se haga de forma segura. Para ello se enviaran vía SSL para que los datos estén encriptados, y para mayor seguridad se pasara una clave que solo el servidor conozca, y de esta forma no sea posible que un usuario intruso no deseado envíe datos dañinos.

# ESTADO DEL ARTE

Los meteoritos siempre han tenido importancia a lo largo de los años, debido al interés por conocer todo lo relacionado con el exterior de la Tierra. También han sido tema de conversación de la mayoría de las personas sobre un posible meteorito gigante que pueda caer en la tierra como se supone que ocurrió con la extinción de los dinosaurios.

Existen aficionados a la astronomía y a la fotografía con ganas de poder observar las llamadas lluvias de estrellas. Existe una en especial de gran actividad, que se extiende desde el 16 de julio hasta el 24 de agosto cada año, dándose su máximo el día 10. Son meteoritos de alta velocidad (59km/s) que radian de la constelación de Perseo, y por ello reciben el nombre de perseidas. Estos meteoritos provienen del cometa 109P/Swift-Tuttle, el cual orbita alrededor del sol con un periodo de 135 años.

Debido al interés astrofísico o simplemente por el hecho de controlar más los acontecimientos que se producen en el firmamento, existen cada vez más una gran cantidad de sistemas de detección de meteoritos. Podemos encontrarnos con varios métodos para la detección, pero solo comentaremos los principales.

Algunos de ellos utilizan procesado de imágenes de cámaras, otros captan ondas de radiofrecuencia reflejadas, y otros simplemente se encargan de observar diariamente el firmamento en busca de posibles meteoroides.

Radio Detección de Meteoros.

http://www.astrosurf.com/blazar/meteoros/Radio%20Observacion%20de%20Meteoros.html

Estos sistemas utilizan las ondas que se reflejan en la estela que dejan los meteoroides al entrar en la atmosfera para detectar y contar el número de meteoros que llegan a la Tierra. Como ya se ha explicado anteriormente, cuando un meteoroide llega a las capas altas de la atmósfera, la fricción hace que la temperatura del cuerpo aumente rápidamente, pudiendo alcanzar los 2000 K. A esa temperatura es cuando comienza la sublimación, dejando un rastro de electrones tras de sí. Este rastro (aire ionizado) es el que interesa ya que es capaz de reflejar las ondas de radio desde 30 Mhz hasta 500 Mhz.

Las ionizaciones puedes ser de pocos segundos de duración (pings) o de hasta varios minutos (burst). Cuando se producen las lluvias de estrellas, es posible establecer conversaciones de radioaficionados a grandes distancias debido a la transmisión de las ondas por medio de la atmosfera.

Este método consiste en utilizar una antena transmisora, la cual emite en ese rango de frecuencias. Las ondas se desplazan hasta la atmosfera donde son reflejadas por el rastro ionizado que ha dejado el meteorito, y se recibe por medio de una antena receptora. A la salida de este receptor se conecta un dispositivo como puede ser un ordenador, que tenga una tarjeta de sonido.

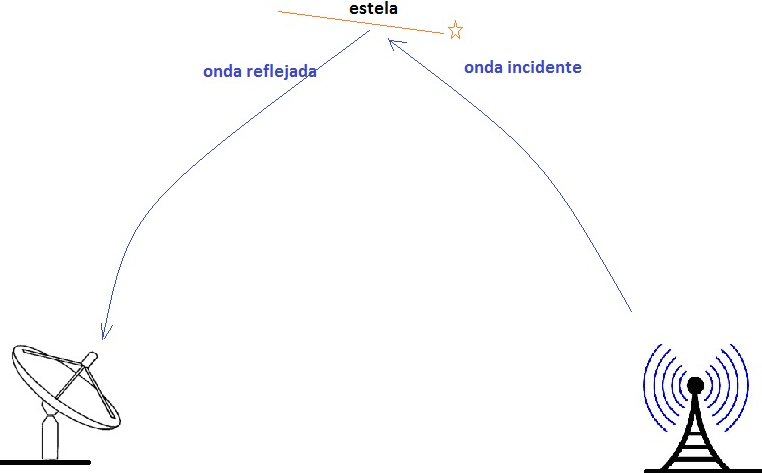


Figura 3.1 Funcionamiento de Radio detección.

Para poder analizar los ecos del sonido existen software libres como por ejemplo el HROFF, y pueden ser analizados y representados por medio de otros software como el Colorgramme Lab V 2.5, el cual se pretende usar para llevar a cabo este proyecto global.

Detección por cámaras CCD.

El dispositivo de carga acoplada (CCD) es in circuito integrado formado por un conjunto de condensadores acoplados. En la fotografía digital se trata de un sensor con pequeñas células fotoeléctricas que registran cada imagen. Estos sensores se basan en transformar la luz que captan en corriente eléctrica. La fotografía CCD ha ayudado a la astronomía sustituyéndola por la fotografía convencional a partir de 1980. Puede alcanzar hasta un 70% de sensibilidad, en comparación con la típica que se encuentra en torno al 2%.

La idea consiste en utilizar varías cámaras con gran angular u ojo de pez con detectores CCD. Estas cámaras monitorizan durante toda la noche el firmamento y permite automatizar la llegada a la atmosfera de meteoroides.



Figura 3.2 Camaras CCD

Estos equipos CCD pueden servir además para calcular la órbita que siguen, de manera que puede saberse de que cuerpo del Sistema Solar procede (de un asteroide, planeta, etc).

Se trata de un sistema de detección muy útil si se quiere estudiar la composición de una roca que haya caído en la superficie terrestre (si no llega a descomponerse) debido a que se conoce el lugar de impacto exacto al quedar recogido en las grabaciones.

# METODOLOGÍA

El proyecto consta de dos partes claramente diferenciadas. Por una parte se realiza la detección de los meteoritos por el método de reflexión de ondas de radio. Por otra parte, se ha realizado un script que se ejecuta periódicamente (cada día) para poder subir a la nube los datos que han sido obtenido y analizados por el software ColorGramme Lab 2.5.

Además se ha creado una aplicación web donde poder observar las detecciones más recientes con sus correspondientes gráficas. Esta aplicación web cuenta con un panel de administrador al cual solo tendrán acceso los usuarios que estén involucrados con el proyecto.

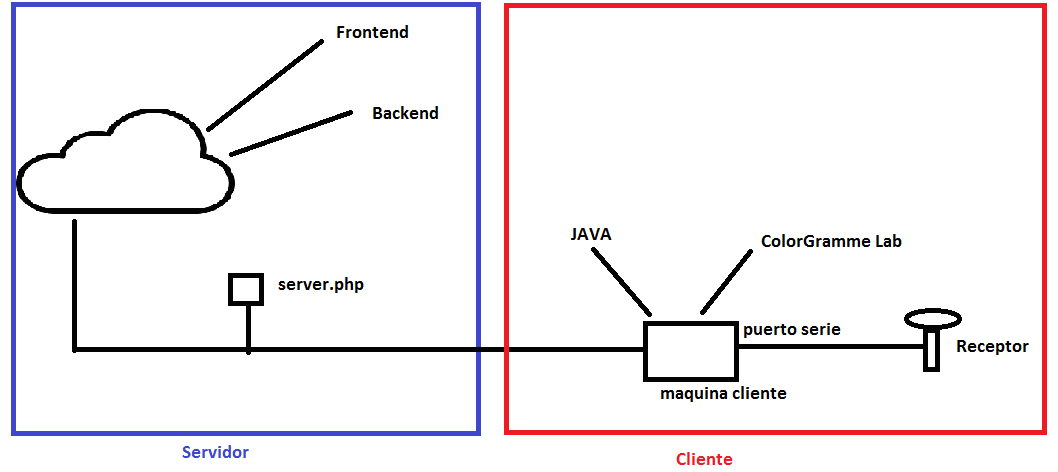


Figura 5.1 Esquema del Proyecto conjunto

Para realizar esta web se ha hecho uso de una plataforma llamada OpenShift. Se trata de un servicio de la empresa REDHAT cuyo objetivo es facilitar el trabajo del desarrollador sin tener que preocuparse de la infraestructura, con muchas opciones como PHP,JAVA,PHYTON,RUBY,etc.

En lugar de utilizar FTP para subir código como es lo habitual en Windows, este trabaja con una herramienta llamada GIT. Se trata de un software diseñado por Linus Torvalds pensado para controlar las versiones de los archivos que vamos añadiendo. Puede dar detalles de la persona que ha tocado por última vez ese archivo, cúando y qué. De esta manera es posible trabajar desde distintos dispositivos y lugares de forma conjunta.

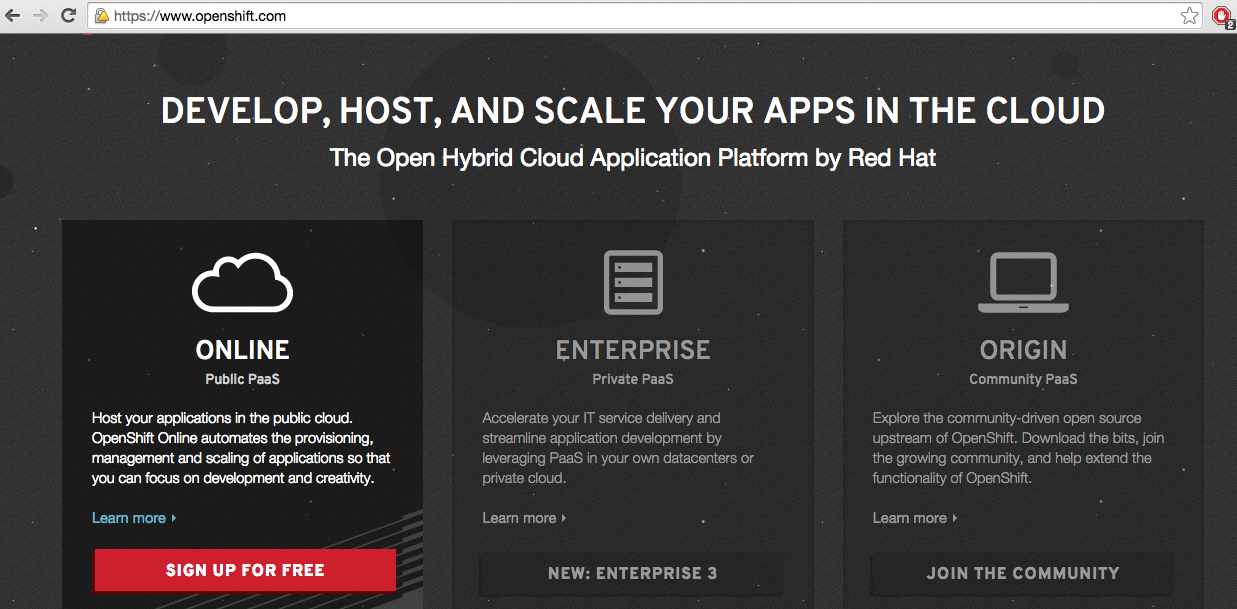


Figura 5.2 Openshift

OpenShift ofrece la posibilidad de crear 3 aplicaciones en cada cuenta de forma gratuita, con posibilidad de pagar para conseguir alguna ventajas (certificado propio para la página web, mayor capacidad de alojar datos, etc).

Una vez creada la aplicación, se ha seleccionado el entorno de programación en el que se va a trabajar, eligiendo en este caso PHP 5.4, MySQL 5.5 como programa gestor de base de datos, y phpMyAdmin 4.0 para poder administrar la información que ha sido almacenada en la base de datos.

Una de las cualidades de OpenShift es que utiliza una clave pública para encriptar la conexión que se realiza entre tu máquina y el servidor, autorizándote solo a ti a poder subir y actualizar el código. Utiliza SSH para realizar las operaciones de GIT para poder dar acceso remoto a la aplicación desde cualquier lugar.

DESCRIPCIÓN DE LA BASE DE DATOS

La pieza clave de esta parte del proyecto consiste en administrar y observar los datos que han sido recogidos por el receptor y el software Colorgramme. Para ello ha sido necesario crear una base de datos con MySQL. Se trata de un sistema multiusuario (con la idea de poder trabajar de manera conjunta), libre y relacional.

El modelo relacional fue creado por el matemático Edgar Frank en 1970, acaparando gran atención por su simplicidad y funcionalidad. Se trata de un conjunto de tablas (o relaciones) con unas determinadas características que se encuentras relacionadas unas con otras. Cuando hablamos de atributos de una tabla nos referimos a las columnas de esa tabla.

La base de datos ha sido creada con el nombre **meteor** mediante la herramienta phpMyAdmin y posee las siguientes tablas:

* administradores
* observación
* observador
* tipodispositivo

A continuación se representa el modelo entidad-relación de las tablas anteriores.

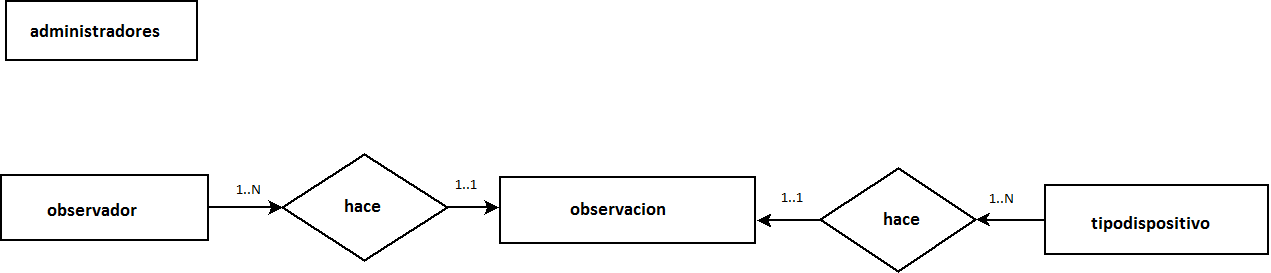


Figura 5.3 Modelo entidad-relación

En la relación existente tenemos que un observador realiza de 1 a N observaciones, y una observación solo es hecha por un único observador.

(Para que un observador este registrado en el sistema debe haber hecho una observación y por ello es 1..N y no 0..N).

Además una observación solo es hecha por un tipodispositivo(1..1), mientras que un dispositivo puede haber realizado de 1 a N observaciones.

(También debe haber hecho una observación para estar registrado como dispositivo, y por ello es 1..N).

Los administradores no tienen ninguna relación con las demás tablas, meramente se utiliza para poder hacer login en el sistema como administrador.

Se describen a continuación las tablas que han sido creadas con sus atributos.

Administradores.

Esta tabla ha sido creada para poder registrar y hacer login las personas que estén encargadas de observar y administrar los datos en el backend. Solo un administrador es capaz de crear un nuevo administrador, de borrar o de modificar.

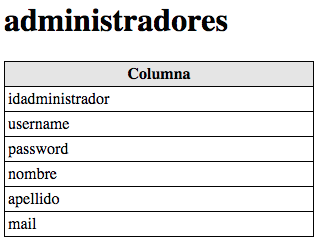


Figura 5.4 Tabla administradores

El idadministrador ha sido establecido como primary key (clave primaria) de la tabla. De esta manera podemos identificar cada tupla de la relación de manera inequívoca, ya que nadie puede tener el mismo idadministrador.

El password ha sido creado de manera que va encriptado en MD5, aunque existen otros métodos más seguros como Blowfish.

Observador.

Al utilizar el script que lee los datos y los sube a la nube aparece el nombre de la persona que ha realizado esa determinada observación. Para poder listar bien las observaciones se ha decidido crear una tabla de observador, con los siguientes atributos.

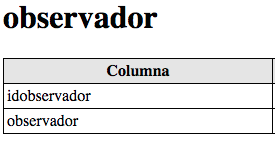


Figura 5.5 Tabla observador

En este caso el idobservador es la primary key, la cual nos ayudara a relacionar esta tabla con la de la observación

Observación.

En esta tabla es donde se guardan todos los datos que se recogen de cada hora del día de cada mes (si se ha realizado correctamente la observación). Además aparecen datos que pueden ser más relevantes

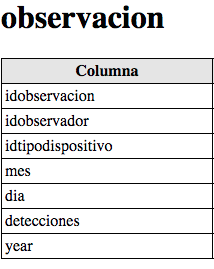


Figura 5.6 Tabla observación

En lugar de año se ha utilizado “year en ingles” debido a un problema que podía surgir por el uso de la ñ. De esta manera se puede saber cuántos meteoros se detectaron en una determinada hora del día.

Se ha optado por utilizar como primary key el idobservacion, y como foreing key (o clave extranjera) idobservador e idtipodispositivo (los dos atributos son enteros).

De esta manera ha sido posible establecer unas reglas en cuanto a esos dos atributos. Si por un casual el idobservador en la tabla de observador es actualizado (cosa que no debería pasar), automáticamente la base de datos cambiara ese idobservador de la tabla observación.

Como el idobservador y el idtipodispositivo son foreing key, no es posible eliminar una tupla de observador o tipodispositivo si existen datos relacionados con estos. Si por un casual esos datos son borrados, entonces si se podría llegar a eliminar las tuplas que quedaban ligadas a ésta.

Tipodispositivo.

Se trata del dispositivo receptor que se ha utilizado para recoger las ondas reflejadas. Todos estos datos son utilizados para el análisis de los datos por el Colorramme.

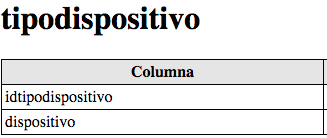


Figura 5.7 Tabla tipodispositivo

Como en los casos anteriores el idtipodispositivo es la primary key de la tabla e identifica a todas las tuplas como únicas.

PROGRAMA JAVA PARA LEER DATOS Y ENVIAR AL SERVIDOR

El programa se ha realizado en JAVA con la ayuda de un par de librerías de JSON. La función que realiza principalmente es la búsqueda de archivos .TXT dentro del directorio que se encuentra (así aparecen los datos guardados por el Colorgramme), y va leyendo uno a uno extrayendo toda la información relevante.

Una vez que se ha guardado todo, se establece una conexión HTTPS (Hyper Text Transfer Protocol) con un archivo PHP alojado en el servidor, y se envían por el método POST. De esta forma los datos se envían encriptados y de forma segura. Lo que ocurre es que la dirección en la que se encuentra el php en el servidor, posee un certificado de seguridad (SSL)

Además del protocolo HTTPS se ha implantado otra medida de seguridad para proporcionar al sistema. Se ha utilizado una clave que solo conoce el cliente y el servidor. Una vez que el servidor recibe los datos, comprueba si la contraseña es correcta, y de ser así, comienza a subir las medidas a la base de datos.

Para ejecutar el programa de forma automática cada día se ha utilizado el demonio cron que mira en el archivo de texto crontab la lista de comandos deseados a ejecutar. Crontab mira la fecha y hora además de los permisos de ejecución, y lo realiza en el background.

Una vez que los datos son enviados correctamente los archivos pasan a ser eliminados del directorio.

A continuación se explica con algo de detalle las partes principales del código implementado.

El archivo de texto que lee el programa tiene la siguiente forma. Se ha obtenido como ejemplo de la página: LINK

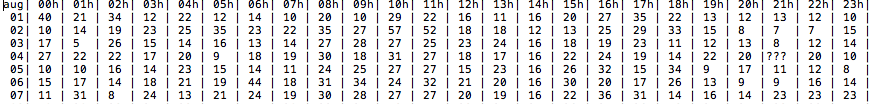


Figura 5.8 Archivo.TXT

En primer lugar, una vez que se ha seleccionado el archivo .txt, se va leyendo línea a línea obtener y guardar todos los datos. A continuación se crea un objeto JSON mediante la librería que ha sido importada previamente, y se le introducen los datos deseados con su clave correspondiente para poder ser obtenidos posteriormente en el servidor. Como por ejemplo la clave que se envía al servidor para comprobar que somos el cliente.



Los datos son concatenados y enviados en una sola cadena, para disminuir las conexiones que se realizan con el servidor. Una vez allí pasan a ser obtenidos de forma individual para poder ser subidos a la base de datos.





Se genera la URL a la que se va a enviar ,se establece la conexión HTTPS y se indica el método que utilizamos para enviar los datos, en este caso POST.

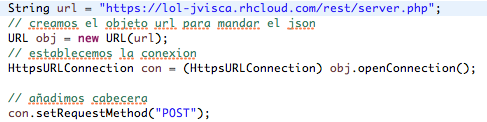


Figura 5.9

Por último se envían los datos por el método POST y se cierra la conexión.

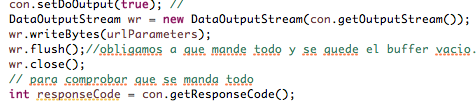


Figura 5.10

El proceso se repite de manera iterada tantas veces como archivos .TXT haya en el directorio en el que se encuentra. Finalmente los archivos son eliminados.

Existen una gran cantidad de lenguajes de programación con el que poder realizar este programa. Posiblemente sea más adecuado utilizar PHYTON, pero nos hemos decantado por JAVA.

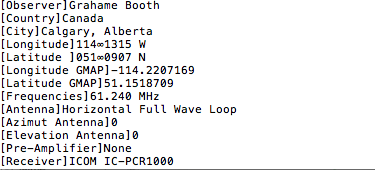


Figura 5.11 Final del archivo TXT

ARCHIVO PHP PARA RECOGER LOS DATOS

Se ha creado un archivo server.php el cual se encuentra alojado en el servidor (dentro de la aplicación que hemos creado previamente). Este se encarga de coger los datos que se envían por POST, darle el formato deseado para ser introducidos en las tablas y hacer la conexión con la base de datos.

Para poder subir los datos a la tabla observación, es necesario hacer varias consultas previas a esta. Como se ha explicado anteriormente, el observador y el tipodispositivo deben ser registrados en la base de datos, con el fin de establecer la relación entre las tablas (idobservador y idtipodispositivo).

Por ello en primer lugar se inserta en la tabla observador el nombre del observador. Al estar configurado el idobservador como autoincrement, esté ira aumentando en 1 automáticamente cada vez que se inserte uno nuevo. Se realiza la misma acción con el tipo de dispositivo.

Una vez hecho esto, se hacen dos consultas más, esta vez para saber el idobservador y el idtipodispositivo que han ocupado en sus respectivas tablas.

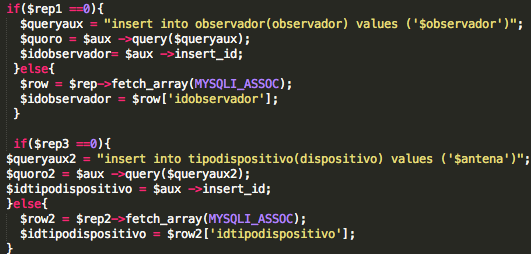


Figura 5.12 Obtención del idtipodispositivo y idobservador

Cuando ya se conocen todos los datos, se realiza un bucle que recorre todos los datos y se van enviando uno a uno en un bucle “for”.

Se muestra a continuación la parte del código donde se suben todos los datos.

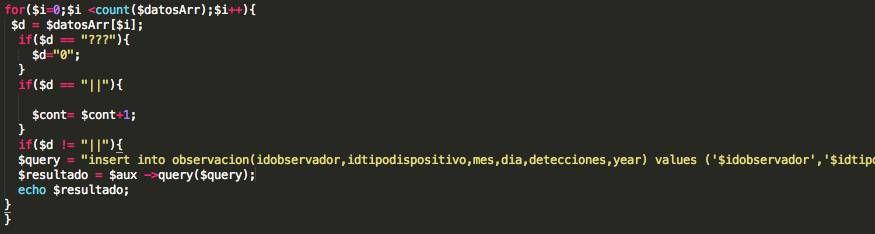


Figura 5.13 Parte del código de server.php

**BACKEND Y WEB PÚBLICA**

En principio la idea era solo realizar una zona de administrador donde poder tratar y manejar los datos de las detecciones. Nos pareció apropiado dejar una parte pública con gráficas y comentarios donde poder observar las últimas novedades de los datos captados. Toda la zona web se ha realizado con HTML, PHP, JAVASCRIPT y CSS para darle algo de estilo.

**https://lol-jvisca.rhcloud.com**

Utilizando herramientas como Bootstrap, se ha logrado conseguir dar estilo de forma sencilla utilizando los css que vienen predefinidos. Además se ha realizado para versión web y para móvil.



Figura 5.14 Imagen de la página versión web

En un principio la idea no estaba del todo clara y no se sabía exactamente si se iba a realizar una parte que fuera de cara al público, para poder curiosear algo sobre las últimas detecciones y sobre lo relacionado con el proyecto. La página web queda recogida en un subdominio de rhcloud, por lo que llegar a ella sin conocer el enlace es prácticamente imposible. Se ha pensado cambiar el nombre en un futuro si se continuara avanzando sobre este trabajo.



Figura 5.14 Captura de pantalla versión móvil

En el Frontend de la página nos encontramos con dos pestañas en el menu superior, una de inicio donde se muestra una breve introducción del proyecto, además de una gráfica que muestra las últimas detecciones que han sido guardadas en la base de datos durante el último mes, y otra pestaña de información donde se habla muy brevemente sobre la teoría de detección de meteoritos por radio.

Para su realización se ha utilizado la herramienta flot chart. Simplemente es una página web en la que están disponibles unas hojas de estilo y unos archivos js para poder realizar graficas de forma sencilla. El resultado es bueno y no tiene gran dificultad para ser utilizado.



Figura 5.15 Numero de detecciones a los largo del último mes

En esta zona pública prácticamente no se hacen llamadas a la base de datos, simplemente para recoger los últimos datos que se han obtenido para ser representados en la gráfica que aparece en la parte inferior del inicio.

Para representar todos los datos simplemente se utiliza una función de flotchart.com que ha sido descargada previamente. En ##1##,##2##, etc se introducen los datos deseados desde el vista.php. Cabe decir que se ha pretendido usar el método MVC (modelo vista controlador). Utilizando este modelo para realizar la página web, se puede dividir el trabajo de forma clara.

El controlador es el encargado de llamar a la vista y al modelo. El modelo.php se encarga de hacer las consultadas a la base de datos, y devolver en una variable los datos obtenidos a la vista. El vista.php se encarga de recoger esos datos, tratar con ellos e insertarlos en la plantilla (en este caso un archivo HTML).

Es un método muy eficaz para hacer código ya que si se desea cambiar el entorno de trabajo de web a móvil u otro lugar, el modelo nos serviría para cualquiera tipo. Simplemente sería necesario sustituir el vista con las plantillas deseadas e introducir los datos en ellas.

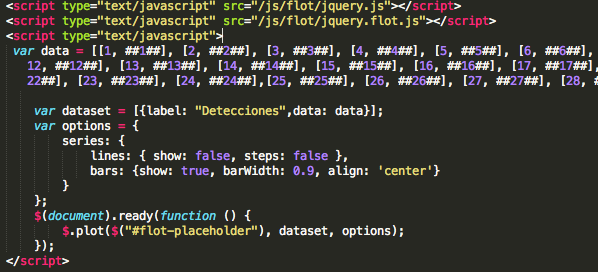


Figura 5.15 Codigo para dibujar la gráfica con flotchart

Este hace una llamada a la base de datos con el máximo idobservacion para ver cual ha sido el ultimo dato introducido. Una vez que se conoce, se mira cual es el observador el mes y el año para realizar otra consulta donde se recogen todos los datos de ese mes.

En cuanto a la zona de administrador, para acceder a ella es necesario hacer login, y por lo tanto estar registrado. Si el usuario y la contraseña son correctos, se guardara la sesión en una variable $\_SESSION() para poder navegar por la zona de administrador sin que expire. Se encuentra en

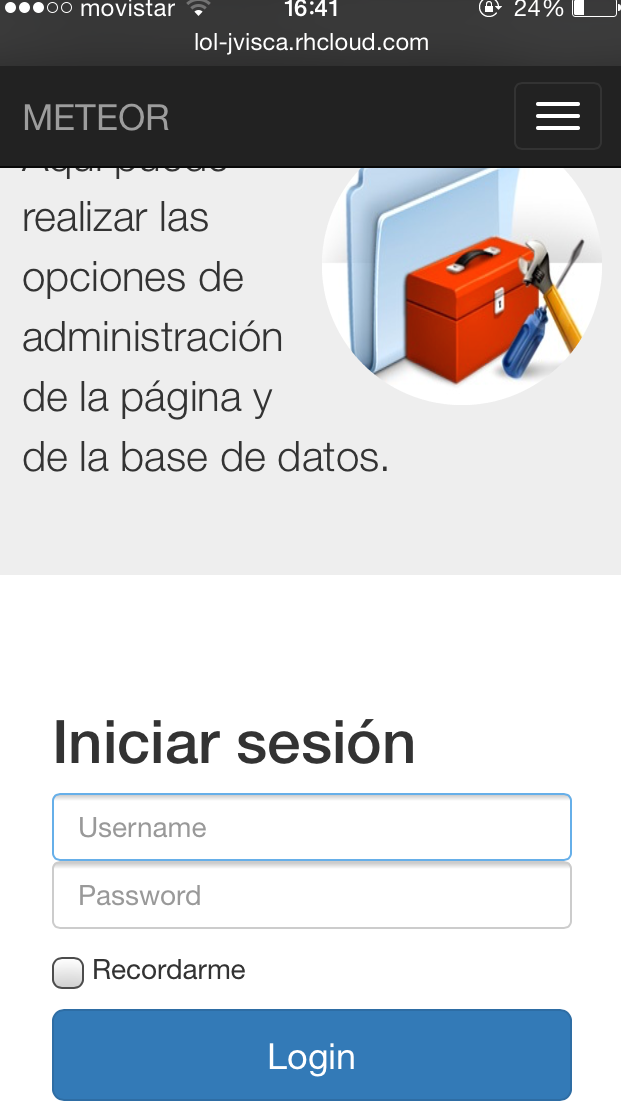


Figura 5.16 Login de la zona de administración

Hay que tener precaución a la hora de realizar los inputs del formulario HTML, debido a que los campos que se introducen van a realizar una llamada a la base de datos, y esto puede dar problemas si no se protege. Pueden aparecer los ataques de SQL INJECTION. Para evitar esto lo único que hay que hacer es filtrar bien las variables utilizadas en la consulta. Los atacantes pueden llegar a obtener cualquier información que se encuentre en la base de datos, así como usuarios, contraseñas, datos privados, etc. Puede utilizarse el comando mysql\_real\_scape\_string para proteger las consultas.

Una vez que se ha iniciado sesión tenemos el siguiente menú de opciones disponibles.



Figura 5.17 Menú de aminitración

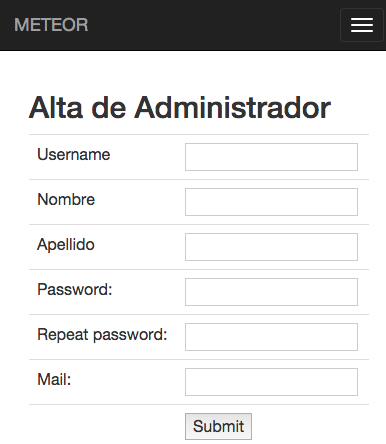


Figura 5.18 Alta de administrador

Es necesario rellenar los campos que aparecen en la figura ¿¿ para poder registrar al nuevo administrador ( nombre y apellido son opcionales). Se comprueba que el Username no este cogido, y que las 2 password sean la misma para poder insertar en la base de datos al administrador.

Si pinchamos en la pestaña Modificar / Eliminar administradores nos aparece una lista con los administradores que existen en el sistema, además de las dos opciones posibles ( figura 5.18)

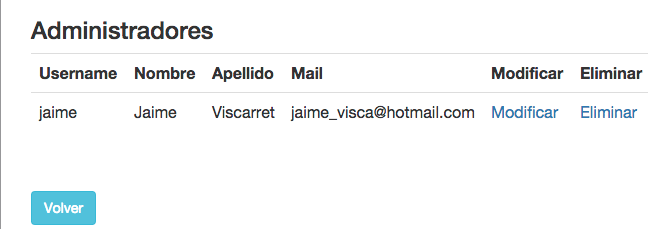


Figura 5.18. Modificar/Eliminar Administradores

En modificar nos aparecerá un formulario relleno con los datos del administrador seleccionado, para poder ser modificado y enviado.



Figura 5.19

Como puede verse en la Figura 5.19 si no introducimos ninguna contraseña se conservara la que ya existía.

En el caso de darle a eliminar simplemente nos aparecerán los datos del administrador y un botón para borrar, que una vez pulsado, será eliminado de la base de datos y ya no tendrá acceso al panel de administración.

En la parte de Observadores, tendremos una zona muy similar a la de administradores y a la de dispositivos. Una vez que se ha realizado una detección completa de todo el mes el archivo es subido a la nube, y es ahí cuando se registra un nuevo observador (si no existía) o no se realiza ninguna acción, ya que los observadores no están repetidos.

A diferencia del apartado anterior de administradores, se ha suprimido el botón de eliminar, ya que como se ha explicado anteriormente al ser el idobservador clave extranjera de la tabla de observaciones no puede ser borrado un observador que relaciona una observación con su idobservador.

Un ejemplo seria tener una tupla de la tabla observación cuyo atributo idobservador sea igual a 1. En ese caso si se eliminara de la tabla observador el que corresponde con el idobservador 1 se daria un error al no existir un idobservador 1 correspondiente a la tupla de observación.



Figura 5.20

Al hacer click en el listado de observaciones aparece una lista con los diferentes observadores que han subido datos.



Figura 5.21

Dando a mas detalles se obtiene una lista de los años en los que ha hecho observaciones el observador determinado. De nuevo mas detalles para poder ver los meses en los que se han recogido muestras de datos a lo largo de ese año y de ese observación concreto.



Figura 5.22

Ya por ultimo se selecciona un mes y se observa una tabla con todos los detalles de la recogida de datos. Se trata del numero de detecciones que se han detectado en cada hora de cada día del mes, en ese año y con ese observador. Esta es la consulta que se realiza a la base de datos para poder obtener los datos correspondientes. Se hablara de los resultados y de las tablas en el apartado siguiente.

Cabe decir que la zona de los dispositivo es muy similar a la de los observadores y no se va a explicar a fondo. Solamente dejar una imagen de algunos de los dispositivos registrados.



Figura 5.23

RESULTADOS

En este apartado se muestran los resultados más representativos de la subida de datos, de las tablas de observaciones y de la base de datos. En primer lugar se muestra como es el formato para enviar los datos al servidor. La idea era subir en un objeto JSON todos los campos que interesaban (dispositivo, observador, mes, año, y los datos de ese mes). Para ello la idea ha sido concatenar los datos de cada hora en una sola cadena, poniendo “||” para saber que se trata de un cambio de día. De esta forma solo se realiza una conexión al server.php por cada archivo .TXT.

Para comprobarlo se ha implementado una línea en el código para imprimir por pantalla los datos que van a ser enviados. El resultado es el de la Figura 6.1

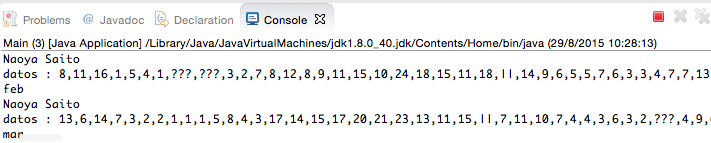


Figura 6.1

Se puede observar que el resultado es de 24 datos concatenados con comas, y que aparece “||” para poder diferenciar el cambio de día. Una vez en el servidor se realiza una separación por comas para formar un array con un dato en cada posición.



Figura 6.2

En la imagen 7.2 tenemos un fragmento del código de server.php, donde se guarda en la variable $datos la cadena mostrada en la figura 7.1, y en $datosArr el array con los datos individuales. Una vez que se tiene el array simplemente

Para demostrar que los datos han sido subidos al servidor correctamente tenemos dos opciones. Una es mirar simplemente en la base de datos, y otra en la zona de administración gracias al apartado de observaciones.

En la figura 7.3 se muestra el numero de observaciones que se han realizado en el mes de febrero del año 2014 por el observador Naoya Saito. Se puede comprobar que son los mismos datos que aparecían en la figura 7.1

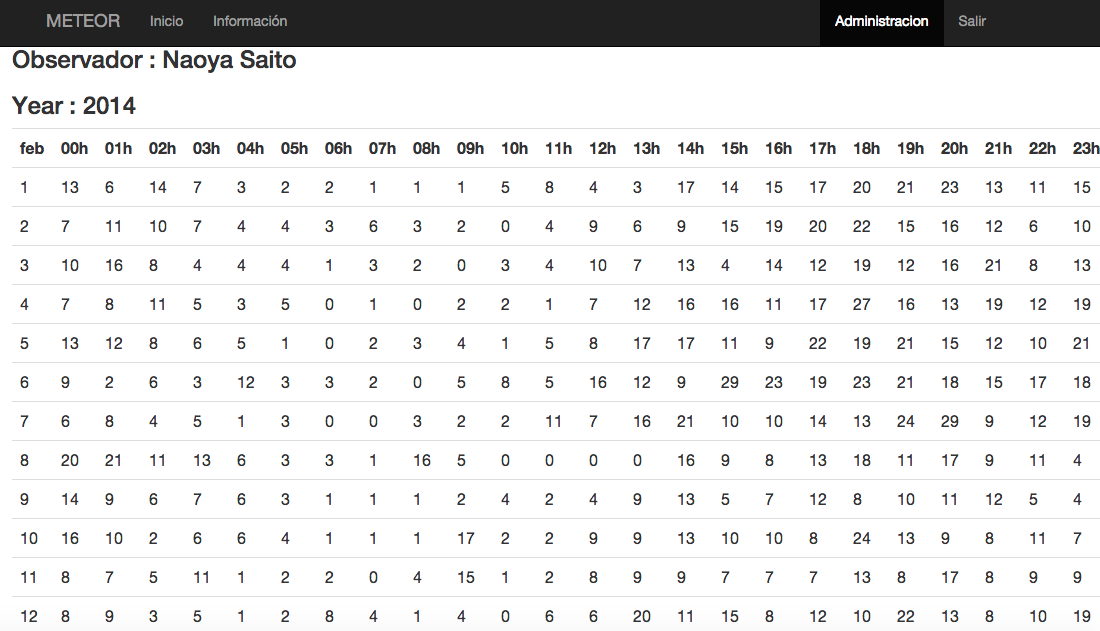


Figura 6.3 Tabla de datos

Otra forma para verlo como ya hemos explicado es realizar una consulta SQL en el phpMyAdmin.

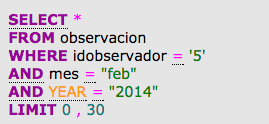


Figura 6.4 Consulta SQL

El resultado de la consulta se muestra en la Figura 7.5

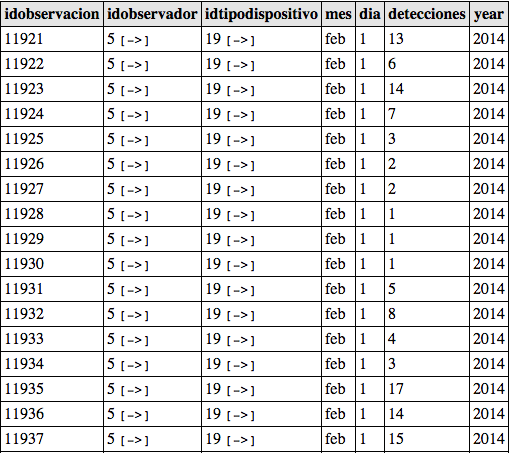


Figura 6.5 Fragmento de la consulta

También aparece el tiempo que ha costado realizarla y el numero total de valores que se han obtenido.



Figura 6.6 Tiempo de consulta

Con todo ello puede demostrarse que el método funciona y que queda claramente recogido en la base de datos para un buen tratamiento de los datos.

PRUEBA CON WIRESHARK

Otro de los objetivos que se planteaban era el envío de los datos al servidor de forma totalmente segura. Para ello utilizábamos el protocolo HTTPS para establecer la conexión con el server.php, ya que rhcloud dispone de certificados de seguridad.

Para comprobar que realmente los datos viajan encriptados se ha realizado capturas del trafico en red con la herramienta Wireshark. En un principio se utilizó el protocolo HTTP y se observó como los datos podían ser vistos si se conocía mínimamente el Wireshark.

En la figura 6.7 se muestra el trafico observado de la conexión entre cliente y servidor.

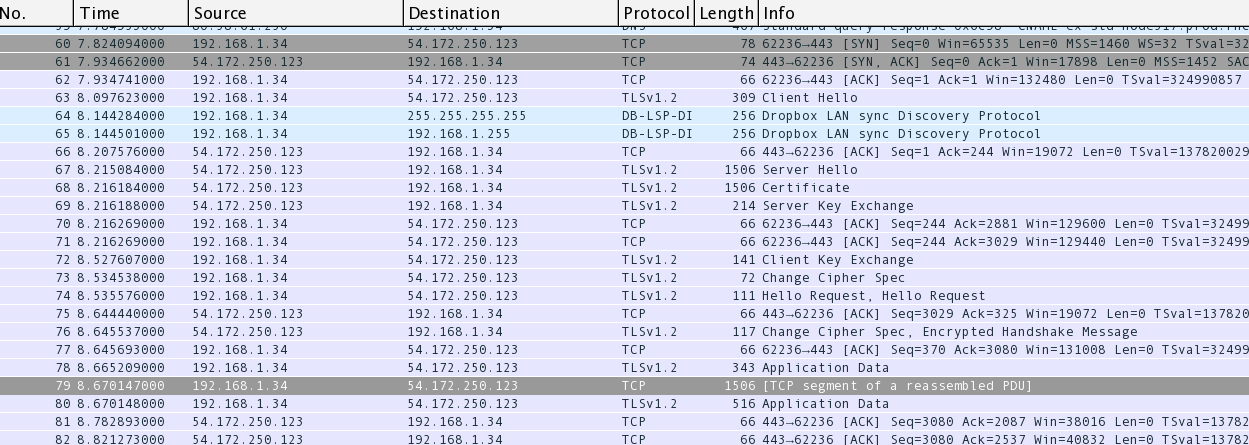


Figura 6.7 Captura de trafico cliente-servidor

Se puede observar el establecimiento de la conexión TCP en la línea numero 60, donde aparece el SYN, SYN +ACK y ACK propios de este protocolo. Así queda establecida la conexión entre cliente y servidor, y ya pueden comenzar a mandar datos. En la línea 63 aparece un tipo de protocolo llamado TLSv1.2 (Transport Layer Security). Se trata de un protocolo criptográfico que proporciona seguridad a la conexión y se basa en especificaciones previas de SSL. El servidor es el que tiene una autentificación y no el cliente.

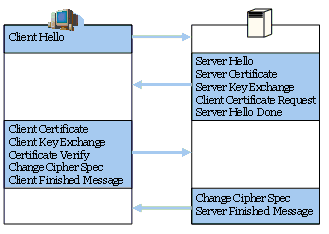


Figura 6.8 Esquema del protocolo TLS

Como puede verse el cliente envía un mensaje Client Hello al servidor donde se especifican los algoritmos, métodos de compresión y versión mas alta de SSL que puede darse durante la comunicación. A continuación se recibe un Server Hello por parte del servidor, donde han sido elegidos los parámetros de conexión en función de lo que ofrece el cliente. Puede verse en la siguiente Figura la forma de cifrar los datos que se establecen.

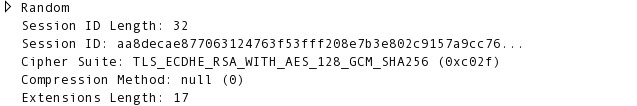


Figura 6.9

RSA es el método de encriptación más conocido, en el que se utilizan la clave publica y la clave privada para cifrar y descifrar los datos. Es aquí donde el servidor envía su certificado de autentificación (línea 68 de la Figura 7.6), además de envíar el Server Key Exchange. A continuación el cliente es el que envía el Client Key Exchange, intercambiándose ya las claves entre cliente y servidor, y estableciendo un canal seguro.

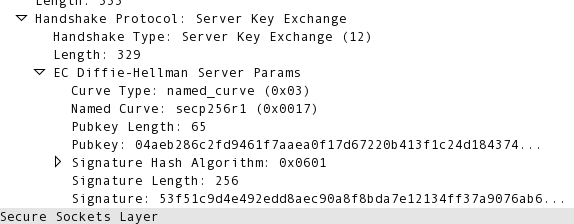


Figura 6.10 Server Key Exchange

Se muestra el contenido del paquete en el protocolo SSL del Server Key Exchange y del Client Key Exchange.

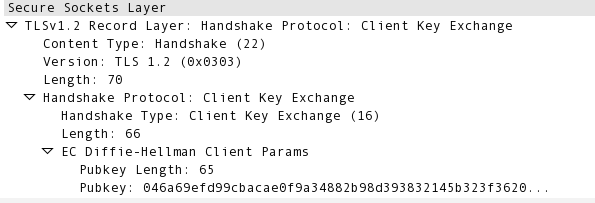


Figura 6.11 Client Key Exchange

En ambos aparece la clave publica que se utiliza para encriptar y desencriptar el mensaje por medio de la clave privada de cada uno (servidor y cliente). De esta forma queda comprobado que la conexión es segura entre cliente y servidor, y gracias a la clave que creamos en el programa JAVA solo el servidor aceptara la conexión con Nosotros (nuestro cliente).

En la Figura 6.12 se puede ver el certificado que tiene el servidor, perteneciente a rhcloud.com, mostrado en el paquete numero 68 de la traza de Wireshark.

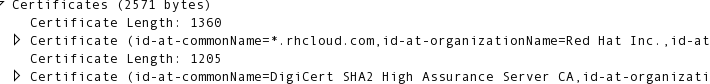


Figura 6.12

https://es.wikipedia.org/wiki/Transport\_Layer\_Security

CONCLUSIONES

En este trabajo se ha realizado una de las dos partes de las que consta este proyecto conjunto. Esta se trataba de recoger los datos que eran analizados por el ColorGramme, conseguir subirlos a la nube de forma segura con una clave de autentificación del cliente.

Ha quedado comprobado que la comunicación era totalmente cifrada y que los datos, si llegaran a ser interceptados por un agente malicioso (por ejemplo si alguien realiza un Man in the middle, en principio no serían entendibles por él)

Además de ello se pretendía realizar una zona de administración de los datos almacenados en la nube. En un futuro si se realizara la otra parte del proyecto ( la de la obtención y detección de los meteoritos con una antena receptora) se podría considerar la idea de aplicar las funciones de administración, que no estaban muy definidas.

La la parte publica de la web deja visibles algunos de los datos en forma de gráficas a cualquier usuario que desee curiosear sobre ello. Estas funciones del Frontend se podrían ampliar también en función del numero de personas que este interesada o que se vea que ha entrado en la web.

BiBLIOGRAFÍA