

# Diseño de un sistema prototipo para transmisión de imágenes de estaciones meteorológicas a través de la red celular para brindar soporte al personal técnico de meteorología del INAMHI

Rodrigo Pucha, Ricardo Llugini, Christian Tipantuña, Escuela Politécnica Nacional (EPN), Quito - Ecuador

**Resumen** – En este artículo se presenta el desarrollo de un sistema prototipo para adquisición y transmisión de imágenes de la bóveda celeste en Ecuador. La adquisición de las imágenes se ha realizado mediante el uso de la plataforma de hardware libre Raspberry Pi, una cámara adaptada con un lente ojo de pez y un receptor GPS (Global Positioning System – Sistema de Posicionamiento Global). El sistema operativo usado es Raspbian, el cual es una versión ligera de la distribución Debian y es recomendado para procesadores de arquitectura ARM32 como la del Raspberry Pi. La transmisión de las imágenes adquiridas se ha realizado cada 5 minutos durante el día usando un módem 4G LTE (Long Term Evolution - Evolución a Largo Plazo). Finalmente esta información se ha almacenado en un servidor FTP (File Transfer Protocol - Protocolo de Transferencia de Archivos) localizado en el INAMHI para ser utilizada en el pronóstico climático.

**Índices** – Imágenes meteorológicas; bóveda celeste; LTE; Raspberry Pi; FTP.

## I. INTRODUCCIÓN

Hoy en día la predicción meteorológica, se realiza utilizando los datos obtenidos de sensores meteorológicos, imágenes de satélite y el uso de modelos matemáticos. Esta metodología debe complementarse con la observación del patrón de nubes durante un cierto periodo de tiempo. Este procedimiento no puede ser realizado por un meteorólogo en diferentes lugares al mismo tiempo, es por ello que es de gran utilidad si estas imágenes se recogen en una estación central con el fin de realizar la predicción del tiempo.

Las estaciones meteorológicas suelen instalarse en lugares remotos donde las redes de datos por cable no están disponibles, el problema de transmisión de información puede solucionarse mediante el uso de la red de telefonía celular.

---

Este trabajo fue apoyado por el Instituto Nacional de Meteorología e Hidrología (INAMHI) de Ecuador.

R. Pucha es graduado de la Carrera de Ingeniería en Electrónica y Telecomunicaciones de la Escuela Politécnica Nacional, Quito, Ecuador (e-mail: rodrigo.pucha@epn.edu.ec).

R. Llugini es docente de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, Quito, Ecuador (e-mail: ricardo.llugini@epn.edu.ec).

C. Tipantuña es docente de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, Quito, Ecuador (e-mail: christian.tipantuna@epn.edu.ec).

Las redes celulares que permiten la transmisión de datos a altas velocidades son 3G y 4G LTE. La última evolución de la tecnología 3G es HSPA + (Evolved High Speed Packet Access - Acceso a Paquetes a Alta Velocidad Evolucionado), que alcanza velocidades de transmisión de hasta 42 Mbps y 11,5 Mbps para el enlace descendente y enlace ascendente respectivamente, tal como se especifica en la versión 8 del 3GPP (Third Generation Partnership Project - Proyecto de Asociación para Tercera Generación). La tecnología LTE (Long Term Evolution - Evolución a Largo Plazo) es catalogada como tecnología 4G por la UIT (Unión Internacional de Telecomunicaciones) porque su rendimiento y capacidades son superiores a los actuales sistemas 3G, esta tecnología 4G LTE puede alcanzar velocidades de transmisión de 300 Mbps y 75 Mbps para el enlace descendente y enlace ascendente respectivamente [1].

Tomando en cuenta algunos aspectos como el envío de imágenes de mayor resolución, menor intervalo de transmisión, agregación de más cámaras, entre otros aspectos que podrían aparecer en el futuro, se ha considerado el uso de la tecnología 4G LTE para este proyecto.

Con el fin de utilizar una plataforma sólida para procesar la información primaria es necesario comparar diferentes tipos de hardware disponibles en el mercado y elegir el más fiable y eficiente, una de las opciones más populares hoy en día es el módulo Raspberry Pi. Este microcomputador permite al desarrollador implementar un amplio espectro de aplicaciones y proyectos tecnológicos con altos estándares de calidad, esta tarjeta electrónica de desarrollo también permite ejecutar un sistema operativo libre.

Además de las tecnologías mencionadas anteriormente, deben discutirse temas cruciales como el consumo de energía, la adaptabilidad de los módulos, la cámara y la memoria para obtener los mejores beneficios del sistema de adquisición y transmisión de imágenes propuesto.

## II. ESTADO DEL ARTE

La estimación de la cobertura nubosa la realiza un observador meteorológico de manera visual desde un punto de la superficie de la tierra, el cual divide la bóveda celeste en ocho partes conocidas como octas con el fin de estimar de

manera subjetiva que cantidad de estas están cubiertas por nubes y enviar la información al centro de comunicaciones [2]. Otra forma más práctica es transmitir fotografías de la cobertura nubosa de la bóveda celeste a través de una red celular con tecnología 3G o 4G.

Existen cámaras comerciales con objetivos ojos de pez que se utilizan para monitorear el cielo, principalmente con fines meteorológicos y astronómicos, un ejemplo es la cámara All Sky Cam del fabricante Moonglow Technologies [3] otra cámara es la Oculus all-sky camera del fabricante Starlight Xpress [4]. Este tipo de cámaras son profesionales por lo tanto su costo es elevado, además para su funcionamiento requieren de un computador con conexión a Internet conectado a la cámara, así como también un software privado. Estos sistemas están siendo utilizados en la actualidad por varias instituciones una de ellas es el Observatorio de la Pontificia Universidad Católica de Chile, el cual usa la cámara All Sky Cam y registra cada cinco minutos las fotografías de la bóveda celeste [5].

El aporte que se realiza mediante este proyecto es desarrollar e implementar un prototipo de bajo costo para transmitir imágenes de la bóveda celeste a través de un sistema integral de transmisión remota de imágenes es decir que incorpora no solo la cámara, sino también los dispositivos de transmisión y soporte como son un módem 4G LTE, receptor GPS y un sistema de alimentación eléctrico fotovoltaico, en resumen es una mini estación portátil de captura y transmisión de fotografías, adicionalmente la programación se realiza sobre una plataforma de hardware y software libre.

### III. METODOLOGÍA

#### A. Selección de la plataforma de hardware libre

La plataforma seleccionada debe tener excelentes capacidades gráficas, de tal forma que permita procesar imágenes sin mayor esfuerzo y brindar alto grado de compatibilidad en cuanto a los periféricos tales como cámara, GPS y módem, adicionalmente como el prototipo estaba pensado para lugares remotos el consumo de energía debe ser el adecuado.

En la Tabla I se hace un análisis comparativo entre varias placas de desarrollo que pudieron haberse utilizado para este proyecto.

De las placas mencionadas en la Tabla I, se escoge la tarjeta Raspberry Pi 2 B porque dispone principalmente de características tales como un conector serial exclusivo para cámaras fotográficas o de video, memoria RAM de 1 GB, puerto Ethernet y USB y su precio es inferior a todas las demás plataformas.

#### B. Direccionamiento de la información

Las imágenes adquiridas por la cámara se enrutan a través de Internet utilizando un módem 4G LTE hacia un servidor FTP. El sistema se ha elaborado con una conexión permanente al servidor, permitiendo el acceso a las imágenes almacenadas en el servidor FTP desde cualquier computador dentro o fuera de las oficinas del INAMHI.

TABLA I RESUMEN COMPARATIVO DE ALGUNAS PLACAS ELECTRÓNICAS

Placa Criterio	Arduino Yún	Beagle Bone Black	Intel Galileo	Raspberry Pi 2 B	Placa seleccionada
Consumo Potencia	1,5 W	2,3 W	3 W	4,5 W	Arduino Yún
Adaptabilidad de Módulos	No tiene conector de cámara. Dispone de transmisión serial y puerto USB para GPS.	No tiene conector de cámara. Dispone de transmisión serial y puerto USB para GPS.	No tiene conector de cámara. Dispone de transmisión serial y puerto USB para GPS.	Tiene conector de cámara. Dispone de transmisión serial y puerto USB para GPS.	Raspberry Pi 2 B
Memoria	RAM: 64 MB microSD	RAM: 512 MB microSD	RAM: 256 MB microSD	RAM: 1 GB microSD	Raspberry Pi 2 B
Conectividad	Puerto Ethernet. Wi-Fi	Puerto Ethernet.	Puerto Ethernet.	Puerto Ethernet.	Raspberry Pi 2 B
Precios	USD 55,01	USD 58,31	USD 62,31	USD 35	Raspberry Pi 2 B
<b>Selección final</b>					<b>Raspberry Pi 2 B</b>

#### 1) Direccionamiento IP

El transmisor consta de un host (Raspberry Pi) conectado a un módem 4G LTE a los que se les ha asignado una dirección IP privada Clase C, mientras que el receptor (servidor FTP) tiene asignado una dirección IP pública fija. La topología de red diseñada se muestra en la Fig. 1.

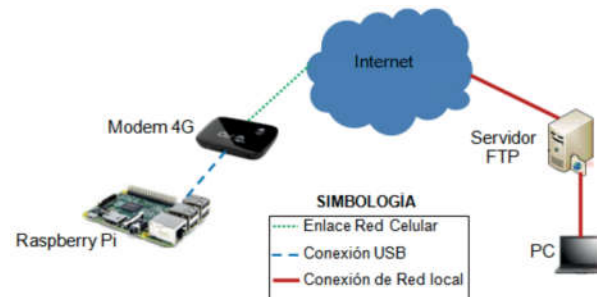


Fig. 1. Topología de la red

#### C. Dimensionamiento del servidor FTP

Esta etapa se ha implementado con el objetivo de determinar el espacio requerido en el disco duro del servidor FTP para la prueba del prototipo.

#### 1) Espacio en el disco duro

Tomando en cuenta una resolución de imagen de 1024 x 768 píxeles, una profundidad de color de 24 bits por píxel, una relación de compresión 4:1 y el formato JPEG, se estimó que el tamaño aproximado del archivo de una imagen estándar es de 576 KB [6].

Con el tamaño del archivo de cada imagen obtenido anteriormente y considerando el envío de imágenes cada cinco minutos todos los días durante 3 meses de pruebas, se calculó que el espacio necesario en el servidor es 7,11 GB.

Considerando un cierto margen de capacidad para futuras

mejoras y/o expansión del sistema, se ha sugerido un espacio libre mínimo de 10 GB en el disco duro del servidor FTP.

#### D. Capacidad del canal mínima necesaria

La capacidad de canal necesaria depende principalmente de parámetros tales como resolución de imagen, compresión, intervalo de transmisión e información de sobrecarga.

La información de sobrecarga corresponde a los bits generados por el encapsulamiento por parte de las capas del modelo TCP/IP (Protocolo de Control de Transmisión / Protocolo de Internet). Un ejemplo de la sobrecarga producida en TCP/IP es la sobrecarga de 78 bytes añadida al tráfico IPv4 (Protocolo de Internet versión 4) cuando se utiliza TCP a través de Ethernet [7].

El archivo también se fragmenta en segmentos TCP que se transmiten individualmente con su respectiva sobrecarga. El número de segmentos TCP y por lo tanto el número de tramas necesarias para enviar un archivo de imagen de tamaño 576 KB se ha estimado en 422 [7].

En consecuencia, si se conoce la sobrecarga para una trama, se puede calcular la sobrecarga total generada cuando se envía una imagen de la siguiente manera (1):

$$O_T = O_{TCP/IP} * F, \quad (1)$$

Donde  $O_T$  es la sobrecarga total,  $O_{TCP/IP}$  son los 78 bytes de sobrecarga cuando se transporta el tráfico IPv4 usando TCP sobre Ethernet para una trama y  $F$  son las 422 tramas. Usando (1) se calculó en 32,14 KB la sobrecarga total generada al enviar una imagen.

La capacidad de canal mínima requerida para garantizar la transmisión de la información completa (evitando congestionar la red) se calcula de acuerdo con (2):

$$v_{tx} = \frac{\#total\_bits\_a\_transmitir}{t_{tx}}, \quad (2)$$

Donde  $v_{tx}$  es la velocidad de transmisión en bps a calcular,  $\#total\_bits\_a\_transmitir$  son los bits de información más la sobrecarga total y  $t_{tx}$  es el tiempo de transmisión.

Sustituyendo la variable  $\#total\_bits\_a\_transmitir$  en (2), se obtuvo una expresión como la mostrada en (3).

$$v_{tx} = \frac{TAI + O_T}{t_{tx}}, \quad (3)$$

Donde TAI es el tamaño del archivo de imagen (576 KB),  $O_T$  son los 32,14 KB de la sobrecarga total y  $t_{tx}$  es el tiempo de transmisión, que en este caso debe ser inferior a 5 minutos.

Sustituyendo los valores anteriores en (3), se calculó que la velocidad de transmisión mínima que el canal debe soportar es de 16,6 Kbps.

#### E. Selección de la tecnología móvil para la transmisión

Se sabe que las tecnologías 3G o 4G proporcionan una velocidad de transmisión superior a los 16,6 Kbps calculados según se indicó en la introducción de este artículo.

Por lo tanto se pueden utilizar para este trabajo cualquiera de las dos tecnologías 3G o 4G LTE puesto que ambas cumplen ampliamente los requisitos de la velocidad de transmisión requerida.

#### F. Programación del sistema prototipo

En esta etapa se describen los principales scripts desarrollados en la Raspberry Pi a través de su consola.

##### 1) Programación de la cámara para capturar fotografías hemisféricas

La fotografía hemisférica es ampliamente utilizada en climatología. Este tipo de imágenes permite obtener un registro detallado de toda la cobertura nubosa de la bóveda celeste. Para capturar fotografías hemisféricas, se adaptó al módulo de cámara un lente ojo de pez para aumentar el campo de visión de la cámara a alrededor de 180 grados.

La Fig. 2 muestra cómo la imagen capturada por el lente se proyecta a un plano bidimensional para formar la fotografía. La porción de la escena capturada depende del campo de visión (FOV) del lente ojo de pez [8].

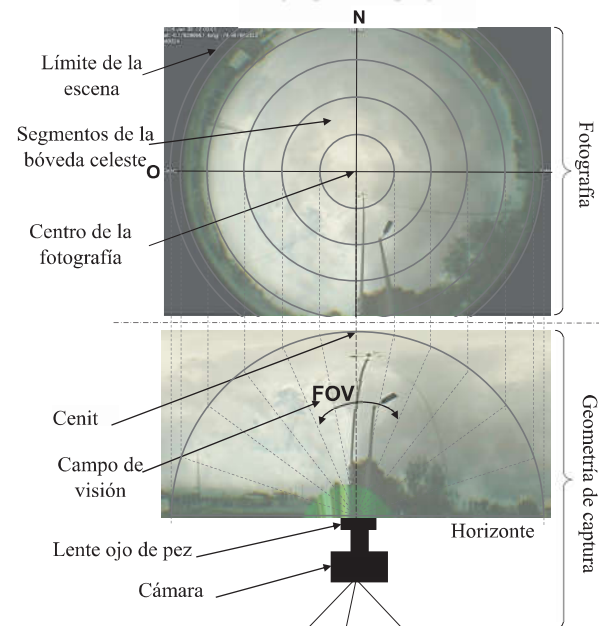


Fig. 2. Proceso de captura de una fotografía hemisférica [8].

Para calcular el campo de visión para grandes distancias de enfoque en comparación con la longitud focal se utiliza (4) [9].

$$FOV = \arctan \frac{d}{2f}, \quad (4)$$

Donde FOV es el ángulo que describe el campo de visión (en la dirección horizontal, vertical o diagonal),  $f$  es la distancia focal y  $d$  es la dimensión del sensor de la cámara (en la misma dirección).

La librería picamera para Python permite controlar el

módulo de cámara Raspberry Pi.

Como se muestra en el Fragmento de Código 1, mediante el método o función `resolution` se especifica la máxima resolución (5 Megapíxeles) de la cámara, para de esta manera obtener el máximo ángulo de visión permitido originalmente por la misma, luego mediante el método `capture` se captura la foto y se guarda en un directorio temporal, no sin antes con la opción `resize` redimensionar la imagen a un valor menor con el fin de reducir su peso de aproximadamente 3,6 MB a alrededor de 576 KB.

```
#!/usr/bin/python
import time
import picamera

with picamera.camera.PiCamera() as camara:
    # Máxima resolución -> máximo ángulo de visión.
    camara.resolution=(2592, 1944)
    time.sleep(3) # 3 s de espera para obtener la imagen
    #Se reescala la imagen para disminuir peso y se guarda.
    camara.capture('/home/inamhi/temporal/foto.jpg',resize=(1024,768))
    camara.close()
```

Fragmento de Código 1. Script en Python para capturar fotografías

## 2) Programación del módulo GPS

El módulo GPS utilizado es el GlobalSat BU-353S4 el cual cuenta con puerto USB, este se utiliza para obtener las coordenadas geográficas de la estación meteorológica y también para sincronizar la fecha y la hora de la Raspberry Pi ya que esta no cuenta con un reloj de tiempo real integrado.

Para controlar y mostrar los datos recibidos del GPS se utilizó el paquete `gpsd` (GPS Daemon) y para sincronizar el tiempo de la Raspberry Pi se utilizó el "demonio" `ntpd` el cual soporta el servicio NTP (Network Time Protocol - Protocolo de Hora en Red) que permite sincronizar el tiempo mediante un receptor GPS (primera opción programada) o Internet.

Para configurar NTP, se ha editado el archivo de configuración "ntp.conf", el cual se encuentra en el directorio `/etc`. En este archivo de configuración, se agregó información sobre el hardware del GPS, así como el intervalo de sondeo que `ntpd` utilizará para leer las marcas de tiempo enviadas por el receptor GPS. Se determinó que un intervalo de sondeo de 16 segundos era suficiente para mantener sincronizada la fecha y la hora de la Raspberry Pi.

La Fig. 3 resume el procedimiento seguido para establecer la fecha y la hora del sistema, así como para obtener las coordenadas GPS de la estación meteorológica. Este proceso se realiza en cada arranque del sistema operativo.

Para obtener la fecha y hora mediante el receptor GPS y ajustar el tiempo del sistema se empleó la herramienta `gpspipe` que permite conectarse a `gpsd`, obtener sentencias GPS y mostrarlas como salida estándar, lo que le permite trabajar como una tubería de `gpsd` a otro programa.

En la Fig. 4 se puede ver parte de la salida del comando `gpspipe`, en donde se resaltan los datos que son de interés.

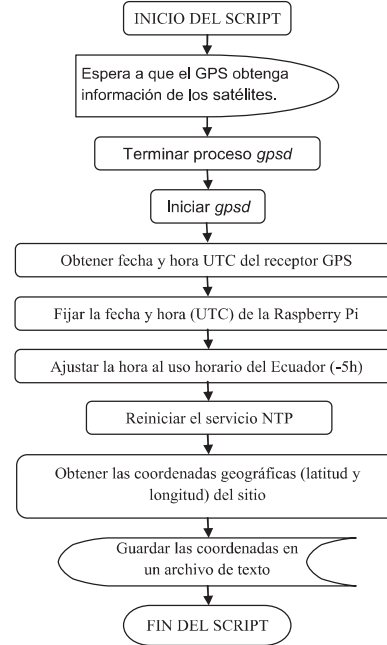


Fig. 3. Diagrama de flujo para auto configurar la fecha y hora

```

inamhi@raspberrypi ~ $ gpspipe -w -n 5
{"class":"IPV","tag":"RMC","device":"/dev/ttyUSB0","mode":3,"time":"2015-10-05T14:15:46.000Z","ept":0.005,"lat":-0.177723333,"lon":-78.487860000,"alt":2830.800,"epx":12.016,"epy":10.730,"epv":46.000,"track":84.3300,"speed":0.000,"climb":0.000,"eps":24.03}
  
```

Fig. 4. Salida del comando `gpspipe`

Tal como se puede ver en el Fragmento de Código 2, se almacena en variables (fecha, hora) parte de la salida del comando `gpspipe` mostrado en la Fig. 4. La opción `-w` indica que imprima las sentencias nativas o decodificadas de `gpsd` que contienen los diferentes datos de salida claramente delimitados (por `:`), se comprobó que con 5 sentencias es suficiente para mostrar el tiempo, latitud, longitud y altitud.

```

# Se obtiene la fecha y hora UTC del GPS
fecha=$( gpspipe -w -n 5 | grep -m 1 lat | cut -d ":" -f 6-12 | cut -b 2-11 )
sleep 4 # Espera 4 segundos hasta obtener la fecha
hora=$( gpspipe -w -n 5 | grep -m 1 lat | cut -d ":" -f 6-12 | cut -b 13-20 )
sleep 4 #espera 4 segundos hasta obtener la hora

date -s "$fecha $hora" # Se configura la fecha y hora UTC
tiempoec="date --date='-5 hour'" # Desfasa -5 horas ->Ecuador
date -s "$tiempoec" # Se configura la hora local
  
```

Fragmento de Código 2. Configuración de la fecha y hora

Mediante el uso del comando `gpspipe` y tuberías se obtiene la fecha, la hora, latitud y longitud que proporciona el receptor GPS.

Una vez obtenidas la fecha y hora (UTC) se fija estos datos al sistema con el comando "date -s" y finalmente se configura la hora local que tiene un desfase de -5 horas con respecto al tiempo UTC como se puede ver en el Fragmento de Código 2.

## 3) Programa para sobrescribir texto sobre una imagen

La mejor forma de presentar al meteorólogo los datos del tiempo y el lugar en el que fue capturada la fotografía es



mediante la superposición de dicha información en forma de texto sobre la imagen.

Este proceso se realizó mediante el software libre "ImageMagick". Como se puede ver en el Fragmento de Código 3 mediante el comando `identify` se obtuvo el ancho (%w) y el alto (%h) en píxeles de la imagen y así el tamaño del texto se definió como función del ancho de la imagen y se ha visto conveniente que este valor sea 1,4% del ancho [10].

```
# El caracter "\n" indica salto de línea
IMAGEN_ENT=/home/inamhi/temporal/foto.jpg
IMAGEN_SAL=$IMAGEN_ENT
ANCHO=`identify -format %w $IMAGEN_ENT`; \
ALTO=`identify -format %h $IMAGEN_ENT`; \
TAMANO_TEXTO=$(( $ANCHO)*14/1000); \
```

Fragmento de Código 3. Parámetros iniciales para editar la imagen

En el Fragmento de Código 4 se observa el uso del comando "convert" de "ImageMagick" para sobreponer los puntos cardinales, coordenadas GPS, código de la estación meteorológica, fecha y hora, la opción `-flop` gira la imagen horizontalmente ya que al enfocar la cámara hacia arriba se produce un efecto espejo que es corregido con esta opción.

```
/usr/bin/convert $IMAGEN_ENT -flop \
-undercolor '#00000040' \
-pointsize $TAMANO_TEXTO -fill white \
-gravity NorthWest -draw "text 0,0 '$FECHA_E'" \
-gravity NorthWest -draw "text 0,$[( $ALTO)*22/1000] 'lat: $LAT
long: $LON'" \
-gravity NorthWest -draw "text 0,$[( $ALTO)*42/1000]
'$CODIGO_ESTACION'" \
-gravity North -draw "text 0,0 'Norte'" \
-gravity South -draw "text 0,0 'Sur'" \
-gravity East -draw "text 0,0 'Este'" \
-gravity West -draw "text 0,0 'Oeste'" \
$IMAGEN_SAL
```

Fragmento de Código 4. Escritura y posicionamiento del texto en la fotografía.

Las coordenadas en donde se ubicará el texto, estas están medidas en píxeles ancho y alto (w, h) cuyo centro (0,0) es la esquina superior izquierda de la imagen (Noroeste). La posición del texto se definió como función del alto de la imagen y se vio conveniente que el texto de latitud y longitud estén a 2,2% de altura y el código de la estación 4,2% de altura, de esta manera el texto no se superpone a sí mismo, como se puede ver también en el Fragmento de Código 4.

#### 4) Programación del cliente FTP

El cliente FTP es un programa de código abierto para Linux llamado Ncftp el cual permite automatizar las tareas mediante línea de comandos. Para cargar los archivos al servidor se usó la herramienta ncfpput que es una utilidad de Ncftp [11].

Tal como se puede ver en el Fragmento de Código 5 la opción `-f` de ncfpput permite leer el archivo que contiene el la dirección IP, usuario y contraseña del servidor, la opción `-R` o modo recursivo permite copiar árboles de directorios completos, la opción `-DD` borra el archivo local si la subida se realizó exitosamente y la opción `-m` intentará crear el directorio de destino remoto antes de copiar los archivos. Posteriormente se especifica la ruta del directorio destino y origen, \*.jpg significa que se cargarán al servidor todos los archivos con la extensión jpg.

```
# La opción -o especifica la opcion avanzada que desactiva el envio del
# comando FEAT ya que esta no responde en el servidor FTP del INAMHI.
CREDENCIALES=/home/inamhi/datos/info_ftp.txt
DIR_REMOTO=/home/pasante/$NOMBRE_ESTACION/$(date +%Y_%b_%d)
DIR_ORIGEN=/home/inamhi/temporal/*.jpg
ncfpput -f $CREDENCIALES -o useFEAT=0 -R -DD -m $DIR_REMOTO $DIR_ORIGEN
```

Fragmento de Código 5. Uso del cliente FTP ncfpput para subir las imágenes

#### 5) Conexión y programación de del módem 4G

Para este proyecto se utilizó el módem Huawei E5377Bs-508. Este módem es reconocido por la Raspberry Pi de forma predeterminada, como una unidad de disco [12]. Este problema se solucionó cambiando el modo de unidad de almacenamiento a modo módem, es decir que el ID del módem en lugar de ser 12d1:1f02 debe ser 12d1:14dc que corresponde a un dispositivo tipo Ethernet [13] se utilizó el programa "usb\_modeswitch" para conmutar del modo "almacenamiento USB" al modo "usbserial" [12].

Se configuró la interfaz eth1 en la Raspberry Pi para permitir la comunicación con el módem.

En caso de falla de la conexión entre la Raspberry Pi y el módem se desarrolló un script que verifique de forma automática la conectividad mediante el comando PING e intente reconectar la interfaz eth1 de la Raspberry Pi, este proceso se puede ver en el Fragmento de Código 6.

```
#!/bin/bash
IPPRUEBA=192.168.0.1 # IP del modem.

ping -c4 ${IPPRUEBA} > /dev/null

if [ $? == 0 ] # Si la variable especial $?=0 -> el modem esta conectado.
then
    exit # Si el modem está conectado correctamente salir del script.
else
    ifdown --force eth1 # Se baja la interfaz eth1.
    ifup eth1 # Se levanta la interfaz eth1.
    exit
fi
```

Fragmento de Código 6. Script que verifica la conectividad y de ser necesario reconecta la interfaz entre el módem y la Raspberry Pi.

#### G. Control del prototipo

El diagrama de flujo de la Fig. 5 indica el proceso que sigue el sistema. Este script principal se ejecuta automáticamente cada 5 minutos, para ello se usó la herramienta "cron".

Desde este script principal se llama al script que realiza el proceso de captura de fotografías descrito anteriormente, también se añaden de manera secuencial las líneas de código desarrolladas anteriormente.

Para detectar si se ha producido un error en la subida de algún archivo al servidor FTP se monitorea el código de retorno de ncfpput almacenado en la variable especial `?`. Si el código de retorno es cero indica que se subió exitosamente el archivo y por lo tanto el script principal finaliza, caso contrario se llama al script descrito en el Fragmento de Código 6 el cual verifica la conectividad con el módem, y posteriormente se reintenta subir el archivo por última vez.

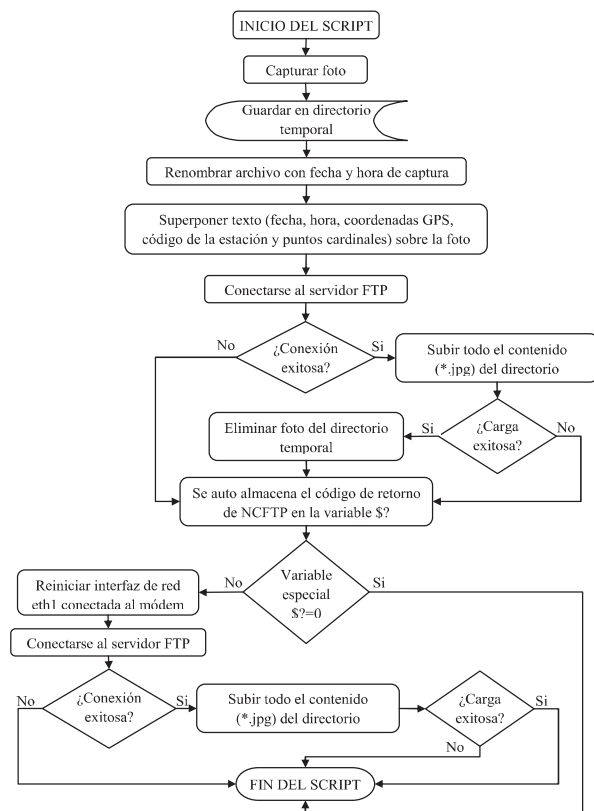


Fig. 5. Diagrama de flujo para controlar todo el prototipo

#### H. Automatización del prototipo

En esta etapa se utiliza el programa cron, este es un "demonio" que permite a los usuarios de Linux ejecutar comandos o scripts automáticamente en una fecha y hora específica.

Ejecutando el comando "crontab -e" se crea un archivo vacío en donde se debe ingresar las tareas programadas, tal como se observa en la Tabla II; en dónde; "m", indica los minutos; "h", la hora; "dom", día del mes; "mon", mes; y "dow" día de la semana en el que se ejecutara el script principal [14].

TABLA II TAREA PARA LA EJECUCIÓN AUTOMÁTICA DEL SCRIPT PRINCIPAL.

m	h	dom	mon	dow	Comando
*/5	6-18	*	*	*	"sudo /home/inamhi/scripts/principal_estacion.sh"

#### I. Alimentación eléctrica de los equipos

La alimentación eléctrica de los equipos se realiza mediante un sistema fotovoltaico, el cual se ha diseñado para una insolación global de 4 KWh/m<sup>2</sup>/día de acuerdo a la investigación del Conelec [15] y una carga DC de 90,37 Wh/día, correspondiente al prototipo. Considerando un factor de seguridad por pérdidas en el sistema de 1,2 (equivalente al 20%), se tuvo una carga diaria de corriente de 9,04 Ah/día.

La corriente de pico del sistema se calcula según (5) [15].

$$I_{\max} = \frac{I_D * E}{N}, \quad (5)$$

Donde  $I_{\max}$  es la corriente pico del sistema en amperios (A),  $I_D$  es la carga diaria de corriente de 9,04 Ah/día, E es la irradiancia solar (condiciones estándar de medida, correspondientes a 1 kW/m<sup>2</sup> sobre el plano del módulo solar) y N es el promedio de la insolación global para los peores meses de insolación (4 KWh/m<sup>2</sup>/día) según la investigación del Conelec [15].

Sustituyendo los valores anteriores en (5), se calculó la corriente de pico del sistema en 2,26 A. Considerando 2 días de reserva y una profundidad de descarga del 60%, se dimensionó un panel solar de 12 V/50 W, una batería de 12 V/35 Ah y un controlador de carga de 3 A.

Puesto que el prototipo funciona únicamente durante el día, se agregó un circuito que mantiene la alimentación eléctrica en el día y la desconecta automáticamente en la noche, para ello se usó un módulo fotorresistencia LDR (Light Dependent Resistor - Resistor Dependiente de la Luz) que a su vez activa o desactiva un relé controlado por un transistor.

#### J. Instalación del prototipo de prueba

El sistema prototipo desarrollado se ha probado en tres estaciones meteorológicas diferentes (Iñaquito, La Tola e Izobamba).

Como se puede ver en la Fig. 6 el prototipo es un sistema totalmente portátil.



Fig. 6. Partes externas del prototipo

#### IV. RESULTADOS

El trabajo de campo determinó que la cobertura celular 4G LTE de la operadora CNT estaba disponible sólo para las estaciones de Iñaquito y La Tola. Para la estación Izobamba estaba disponible en el momento de la prueba solo la cobertura celular 3G. Tomando en cuenta esto, se realizaron las pruebas concernientes a la captura y envío de imágenes de la bóveda celeste.

A pesar de inconvenientes como: congestión de la red, geografía del sitio, condiciones ambientales, se logró verificar la conectividad entre transmisor y receptor, a través de la realización de pruebas de ping, traceroute y netstat. Así también, las fotografías captadas se transmitieron sin

inconvenientes y con una adecuada calidad, una imagen de ejemplo capturada desde la estación Izobamba se ilustra en la Fig. 7.

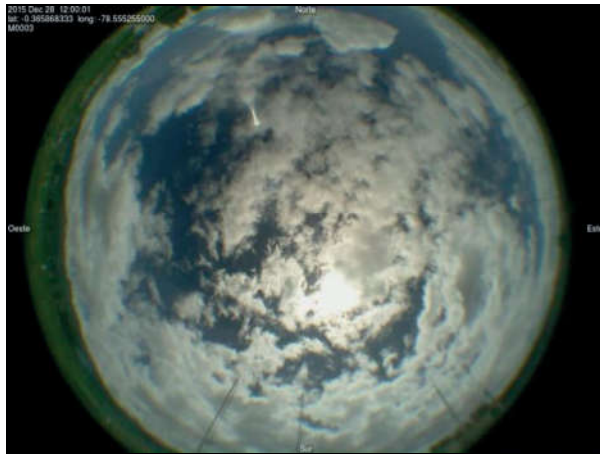


Fig. 7. Fotografía de la bóveda celeste vista desde la estación Izobamba

Como una forma alternativa para visualizar las imágenes se desarrolló una página web para poder ver las últimas fotografías que llegan al servidor FTP.

#### A. Análisis de resultados

Las imágenes mostraron información verdadera que permitieron identificar el clima en el lugar de captura y el campo de visión de la bóveda celeste fue lo suficientemente amplia como para analizar toda la cobertura nubosa. Las imágenes recibidas también mostraron suficiente información que permitió al meteorólogo dar un criterio relativo al clima basándose en la cobertura nubosa sobre la bóveda celeste.

Las imágenes obtenidas muestran la fecha y hora en la que fueron capturadas, lo que permite diferenciar una de otra y comparar entre diferentes instantes para ver la nubosidad a lo largo del tiempo. En la fotografía obtenida también se registran las coordenadas GPS expresadas como latitud y longitud. Adicionalmente se incluye el código de la estación meteorológica. También se muestra en las fotografías los puntos cardinales, gracias a ellos el observador puede conocer la posición de las nubes dentro de la bóveda celeste y también la dirección de las mismas mediante el análisis de varias fotografías. Lo anteriormente mencionado se ilustra en la Fig. 8.



Fig. 8. Visualización de la fecha y hora, coordenadas geográficas, código de la estación y puntos cardinales superpuestos en la fotografía

## V. DISCUSIÓN

Este sistema prototipo desempeña funciones similares en

relación con la cámara comercial All Sky Cam del fabricante Moonglow Technologies. La Tabla III muestra una comparación técnica del sistema prototipo diseñado versus la cámara comercial All Sky Cam.

TABLA III SISTEMA PROTOTIPO VERSUS ALL SKY CAM.

Parámetro	Sistema prototipo	All Sky Cam [3]
Campo de visión	Alrededor de 180 grados	190 grados
Proyección	Ojo de pez circular	Ojo de pez circular
Resolución	1024x768, máx 5 MP	546x457
Tecnología	Digital	Analógico
Sistema Operativo	Linux (Raspbian)	Windows
Conectividad a Internet	A través de un módem 4G LTE y la placa Raspberry Pi incluidos	Necesita una PC con conexión a Internet
Software para subir las imágenes al servidor	Software libre (Ncftp) incluido	Necesita de un software privado no incluido
Método de carga de las fotos	Automático a un servidor FTP, las imágenes se muestran en un sitio web diseñado	Automático a un servidor FTP o al sitio web
Localización del sitio de monitoreo	Automático con un receptor GPS	Manual
Alimentación eléctrica	DC 12 V, 2,26 A (Todo el prototipo)	DC 12 V, 120 mA (solo la cámara)
Estructura de soporte	Totalmente portátil de acero galvanizado	No incluida
Funcionamiento en exteriores	Si	Si
Operación nocturna	No	Si

Como se demostró este prototipo puede competir con otros productos comerciales, su principal ventaja es que es un sistema completo, portátil y listo para usar. Otra ventaja es que el prototipo se ha desarrollado sobre la base de hardware y software libre, lo que permite reducir los costos de licencias. El prototipo también puede trabajar en lugares aislados donde no hay redes cableadas debido a su conectividad inalámbrica a través de la red celular.

## VI. CONCLUSIONES Y RECOMENDACIONES

Mediante las pruebas del sistema, se comprobó que el prototipo es un sistema robusto, en cuanto a transmisión, ya que se comprobó que a pesar de pérdida de la conexión de datos entre el módem y la Raspberry Pi, no fue necesario reiniciar manualmente todo el sistema sino que gracias los scripts desarrollados el prototipo automáticamente detecta esta falla y se reconecta automáticamente a Internet.

En las pruebas de campo se verificó que el receptor GPS nunca perdió la conexión con los satélites incluso cuando estas se realizaron cerca de edificios, árboles y líneas de transmisión eléctricas. Para evitar problemas con la sincronización de tiempo por fallas del GPS se configuró al sistema para sincronizar el tiempo a través de Internet usando el módem.

Con la resolución escogida para las fotografías de 1024 x 768 pixeles los meteorólogos comprobaron que era suficiente para visualizar en detalle la cobertura nubosa de la bóveda celeste, de igual manera la lente ojo de pez de 235 grados de ángulo de visión fue idónea ya que cubría no solo las nubes del horizonte sino también parte de los edificios, vegetación y montañas cercanas a la estación meteorológica.

La velocidad de subida de los archivos al servidor FTP fue de aproximadamente 500 Kbps usando la red LTE, mientras que a través de la red 3G la velocidad aproximada fue de 140kbps.

Si el módem pierde conexión a Internet por ejemplo debido a problemas de cobertura, congestión de la red o daño del dispositivo, también se perderá la conexión con el servidor FTP y por tanto las imágenes no podrán cargarse, si esto ocurre el prototipo está preparado para guardar las fotografías en la tarjeta microSD y subirlas junto con las imágenes nuevas tan pronto se restablezca la conexión.

El sistema prototipo cuenta con las debidas protecciones del ambiente externo, sin embargo se recomienda realizar un mantenimiento por lo menos una vez al mes, especialmente del domo que protege a la cámara ya que podría acumularse polvo u otros objetos que obstruyan la visibilidad. También es recomendable limpiar el panel solar periódicamente.

Como trabajo futuro podría utilizarse el módulo de cámara Raspberry Pi sin filtro infrarrojo (NoIR) para fotografiar la cobertura nubosa en la noche.

Como una mejoras al prototipo, se podrían implementar un programa para rectificar el efecto ojo de pez producido por la lente y mostrar una foto panorámica, también es posible desarrollar un software para procesar las imágenes y calcular la cobertura nubosa en unidades llamadas octas, en base a la cantidad de pixeles que estén cubiertos de nubes.

#### VII. AGRADECIMIENTOS

Los autores manifiestan su agradecimiento al Instituto Nacional de Meteorología e Hidrología y al Ingeniero Marcelo Ayabaca por su constante apoyo brindado durante el inicio, desarrollo y culminación de este proyecto.

#### VIII. REFERENCIAS

- [1] L Alcatel. (2009, Nov.). Las Redes Móviles de Última Generación LTE: Long Term Evolution. Jornada en la Cátedra Alcatel-Lucent. pp. 14-15. [Online]. Disponible: [http://www.dit.upm.es/~catedra-alu/LTE\\_jornadas\\_catedra\\_UPM.pdf](http://www.dit.upm.es/~catedra-alu/LTE_jornadas_catedra_UPM.pdf)
- [2] O.M.M. (2010). Guía de Instrumentos y Métodos de Observación Meteorológicos. parte 1. cap. 15. pp. 6. [Online]. Disponible: [https://library.wmo.int/pmb\\_ged/wmo\\_8-2014\\_es.pdf](https://library.wmo.int/pmb_ged/wmo_8-2014_es.pdf)
- [3] Moonglow Technologies. (2017). Specifications (ASC-N1: NTSC version). All Sky Cam. [Online]. Disponible: <http://www.moonglowtech.com/products/AllSkyCam/Specifications.shtml>
- [4] Starlight Xpress. Oculus. 1/2" Format All-Sky CCD Camera. [Online]. Disponible: [https://www.sxccd.com/Oculus/Oculus\\_Product\\_Brochure.pdf](https://www.sxccd.com/Oculus/Oculus_Product_Brochure.pdf)
- [5] P.U.C. (2012). [Online]. Disponible: [http://www2.astro.puc.cl/ObsUC/index.php/AllSky\\_Cam](http://www2.astro.puc.cl/ObsUC/index.php/AllSky_Cam)
- [6] U. J. C. Rey. Compresión de Imágenes. Visión Artificial. pp. 6,12. [Online]. Disponible: <https://es.scribd.com/document/80190576/Modulacion-Digital-Compresion-de-Imagenes>
- [7] R. Weiss, JDSU. (2008). File Transfer Protocol (FTP) Throughput Testing. White Paper. pp. 4-5. [Online]. Disponible: [http://www.viavisolutions.com/sites/default/files/technical-library-files/FTP\\_WP\\_acc\\_TM\\_AE.PDF](http://www.viavisolutions.com/sites/default/files/technical-library-files/FTP_WP_acc_TM_AE.PDF)
- [8] G. Mauro-Díaz, J.D. Lencinas, y H. del Valle. (2014). Introducción a la fotografía hemisférica en ciencias forestales. Madera y Bosques, 20(1), 109-117. [Online]. Disponible: <http://www.scielo.org.mx/pdf/mb/v20n1/v20n1a10.pdf>

- [9] A. Douglas, P.E. Kerr. (2006, Feb.). Field of View in Photography. pp. 6. [Online]. Disponible: [http://dougkerr.net/Pumpkin/articles/Field\\_of\\_View.pdf](http://dougkerr.net/Pumpkin/articles/Field_of_View.pdf)
- [10] Image Magick Studio. (2017). Features and Capabilities. [Online]. Disponible: <http://www.imagemagick.org/script/index.php>
- [11] M. Gleason. NeFTP - Browser program for the File Transfer Protocol. [Online]. Disponible: <http://www.ncftp.com/ncftp/doc/ncftp.html>
- [12] D. Raboud. Manual del programa usb\_modeswitch. Obtenido al digitar el comando "man usb\_modeswitch" en la distribución Raspbian instalada en la Raspberry Pi.
- [13] DD-WRT. (2017, Jun.). Mobile Broadband. DD-WRT Wiki. [Online]. Disponible: [http://secure.dd-wrt.com/wiki/index.php/3G/\\_/3.5G](http://secure.dd-wrt.com/wiki/index.php/3G/_/3.5G)
- [14] S. González. (2017). Manual Básico de Cron. Linuxtotal. [Online]. Disponible: [http://www.linuxtotal.com.mx/?cont=info\\_admon\\_006](http://www.linuxtotal.com.mx/?cont=info_admon_006)
- [15] Conelec. (2008, Ago.). Atlas Solar Del Ecuador Con Fines De Generación Eléctrica. Quito. pp. 7-8, 49. [Online]. Disponible: [http://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2015/11/Atlas-Solar-Ecuador-CONELEC-CIE-2008-dic\\_2014.pdf](http://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2015/11/Atlas-Solar-Ecuador-CONELEC-CIE-2008-dic_2014.pdf)

#### IX. BIOGRAFÍAS



**Rodrigo Pucha**, nació en la parroquia Cornejo Astorga perteneciente al cantón Mejía de la provincia de Pichincha el 29 de Mayo de 1987. Se graduó en la Escuela Politécnica Nacional como Ingeniero en Electrónica y Telecomunicaciones en 2016. Áreas de interés: Linux, Plataformas de hardware libre, Televisión Digital y Domótica. (rodrigo.pucha@epn.edu.ec)



**Ricardo Llugsi**, Ingeniero en Electrónica y Telecomunicaciones (2008). Magister en Conectividad y Redes de Telecomunicaciones, Escuela Politécnica Nacional (2017). MSc. en Ciencias en Ingeniería de Comunicaciones. Universidad de Manchester (2013). Docente de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional. (ricardo.llugsi@epn.edu.ec)



**Christian Tipantuña**, Ingeniero en Electrónica y Telecomunicaciones. Escuela Politécnica Nacional. MSc en Wireless Systems and Related Technologies. Instituto Universitario Politécnico de Turín. Candidato a PhD en Telecomunicaciones. Universidad Politécnica de Cataluña. (christian.tipantuna@epn.edu.ec)