

# Vitality Goes para Linux y Windows

## Creador: Jamie Vital

### (Traducción al español por XQ6DLW, Demys)

Vitality Goes es una aplicación web para mostrar datos satelitales meteorológicos geoestacionarios y otros. El software está diseñado para mostrar texto e imágenes recibidas de satélites GOES a través de goestools en Linux.

### ¿Qué hace Vitality GOES?

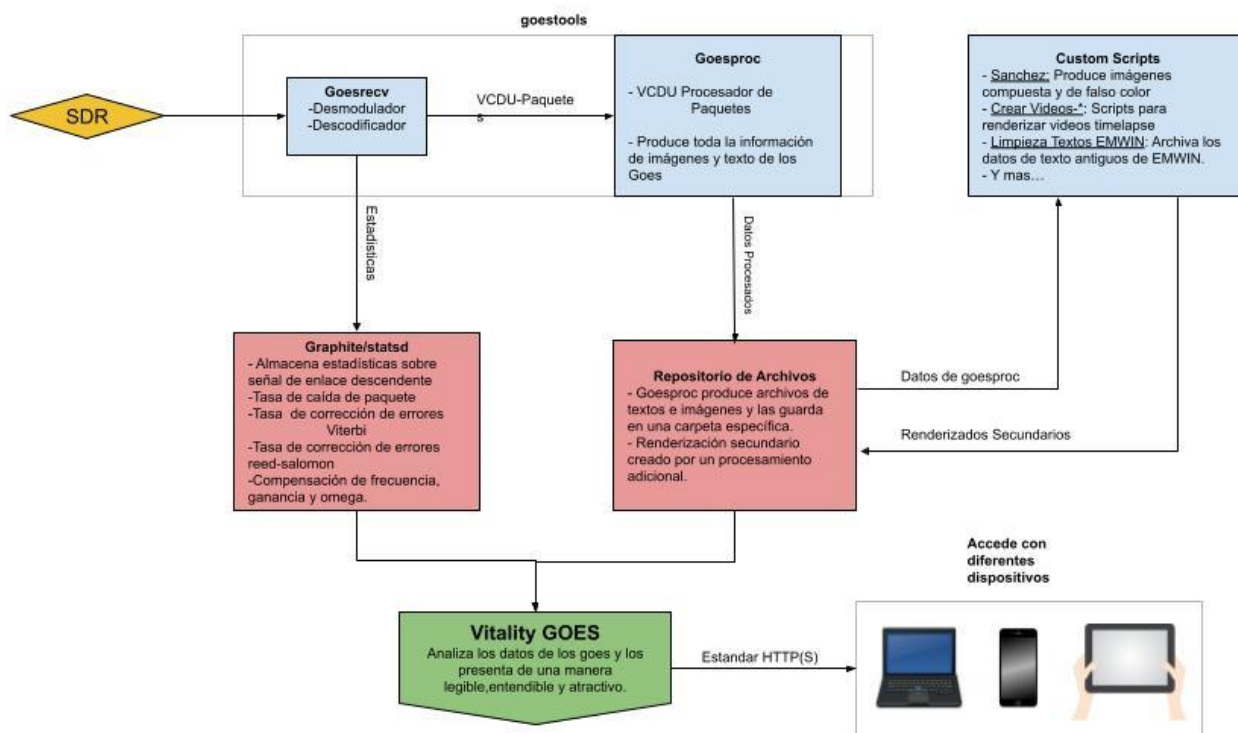
Vitality GOES hace que los datos de la fuente GOES HRIT/EMWIN sean fácilmente accesibles, a través de un navegador web, desde cualquier lugar de su red local. Incluso si Internet se cae, las personas en su LAN local aún pueden acceder a información meteorológica en tiempo real.

Vitality GOES tiene las siguientes características:

- Una vez configurado por el técnico de la estación terrestre (¡justed!), Vitality GOES es fácilmente utilizable por cualquier persona sin conocimientos de radio, satélites o programación.
- Presenta todas las imágenes de disco completo, productos de nivel 2 y mesoescala de una manera fácil de usar y fácilmente navegable.
- Los datos pertinentes de EMWIN (que incluyen las condiciones meteorológicas actuales, pronósticos, relojes y advertencias) se presentan al usuario de una manera atractiva y fácil de leer. No es necesario analizar los datos de otras ubicaciones: los datos de su ubicación configurada es lo único que se le muestra.
- Es capaz de monitorear el estado de la pila de goestools subyacente, incluidas las temperaturas de los sistemas, las tasas de corrección de errores y las tasas de caída de paquetes.

### ¿Cómo funciona?

El siguiente diagrama muestra cómo fluyen los datos desde un satélite GOES, a través de Vitality GOES, y a sus usuarios finales:



Se recomienda alojar Vitality GOES en su propia estación terrestre junto con Goestools para obtener la información más actualizada y simplificar la configuración / mantenimiento. Si lo desea, se puede alojar en otra máquina si tiene un proceso de sincronización configurado entre la estación terrestre y el servidor Vitality GOES. *La sincronización de las imágenes recibidas de otra máquina está fuera del alcance de Vitality GOES.*

Se recomienda utilizar una distribución Linux basada en Debian para alojar el servidor Vitality GOES. Se prefiere algo como Raspberry Pi OS, Ubuntu o Debian.

Vitality GOES alojado en Windows se ejecuta más lento que cuando se aloja en Linux, y su almacén de datos debe mantenerse en una partición NTFS si desea que se cargue la información meteorológica. Es un problema conocido en PHP que las operaciones de archivos son más lentas en Windows.

Si habilita los scripts de representación de video, tenga en cuenta que estos scripts pueden requerir más energía de la que puede proporcionar una máquina de gama baja,

como una Raspberry Pi. Es posible que deba descargar estas tareas a otra máquina o actualizar su servidor a algo más robusto. Estoy usando una computadora portátil con un procesador Core i5 de 4ª generación, y tiene potencia más que suficiente para ejecutar goestools, Vitality GOES y todos los scripts secundarios.

Una vez configurado, cualquier navegador web moderno puede conectarse a Vitality GOES y ver los datos.

**Nota: El programa goestools con Vitality Goes no es compatible con Raspberry 2 , 3 y 3B, producto a un elevado consumo por proceso ya que cuentan con solo con solo 1Gb de ram y también degrada la tensión de la fuente original de 5 v 2ª. Esto produce una afectación directa en el SDR y puede ocasionar perdidas de paquetes en la recepción del Goes. Se encarece más si usamos vías T del rtl.sdr para alimentar el LNA para Goes. Raspberry 3 y 3B se desempeña positivamente solo con Gooestols.**

## Preparación del sistema para Vitality GOES

### Grafito/statsd

Goesrecv admite el registro de información sobre la tasa de corrección de errores, las tasas de caída de paquetes, etc. en un servidor statsd. Esta información es invaluable para los operadores de estaciones terrestres, por lo que debe ser fácilmente accesible. Este proyecto logra esto al organizar la información en una base de datos de grafite, que Vitality GOES puede consultar y presentar al usuario.

La configuración de Graphite no es necesaria para usar Vitality GOES, pero no habrá gráficos disponibles si no lo configura. Si Vitality GOES está en una máquina diferente de goestools, se puede instalar grafite/statsd en cualquiera de las máquinas. Para configurar graphits/statsd:

Instale Docker en el equipo de destino. Esto varía según la distribución, pero puede encontrar instrucciones, pero en esta ocasión lo aremos sobre Debían y Raspbian.

**Abriremos un terminal en nuestra pc o raspberry con debían o raspbian y entramos como super usuario o root con el siguiente comando**

sudo su

En nuestro terminal quedaremos reflejado de la siguiente manera

```
root@raspberrypi:/home/pi#
```

**Actualizamos con los siguientes comandos.**

```
sudo apt update
```

```
sudo apt upgrade
```

**Instale los paquetes necesarios para ejecutar apt to sobre Https**

```
sudo apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

**Agregue la clave GPG**

Ahora necesitamos agregar la clave GPG de Docker a nuestro sistema Debian. Esto asegura que las descargas sean válidas.

Para agregar la clave GPG para el repositorio oficial de Docker, ejecute siguiente comando.

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

**Agregue el repositorio de Docker a las fuentes APT**

En este paso, agregará el repositorio de Docker a las fuentes de apt con el siguiente comando.

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

**Actualizar la base de datos del paquete**

```
sudo apt update
```

**Verificar la fuente de instalación**

```
apt-cache policy docker-ce
```

Esto devolverá una salida similar a la siguiente. Sin embargo, el número de versión de Docker puede variar.

```
Installed: (none)
```

```
Candidate: 5:19.03.5~3-0~debian-stretch
```

```
Version table:
```

```
5:19.03.5~3-0~debian-stretch 500
```

```
500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```

```
5:19.03.4~3-0~debian-stretch 500
```

```
500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```

```
5:19.03.3~3-0~debian-stretch 500
```

```
500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```

```
5:19.03.2~3-0~debian-stretch 500
```

```
500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```

```
17.03.1~ce-0~debian-stretch 500
```

```
500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```

```
17.03.0~ce-0~debian-stretch 500
```

```
500 https://download.docker.com/linux/debian stretch/stable amd64 Packages
```

## Ahora a instalar Docker

```
sudo apt install docker-ce
```

Esto instalará Docker, arrancará el motor y lo configurará para que siempre se inicie al arrancar. Una vez completada la instalación, verifique que se esté ejecutando.

```
sudo systemctl status docker
```

Si se instala correctamente, la salida indicará que el servicio está activo y en ejecución.

Saldrá esto

```
● docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
Active: active (running) since Mon 2019-12-09 11:48:32 UTC; 41s ago
Docs: https://docs.docker.com
Main PID: 17486 (dockerd)
CGroup: /system.slice/docker.service
└─17486 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
```

Para salir Control+C

Docker estaría listo para usarse en su raspberry o computador.

## Preparando Grafito/statsd

Como raíz, ejecute los siguientes comandos para crear un área de almacenamiento para el grafito.

```
Sudo mkdir -p /var/lib/graphite/config
Sudo mkdir -p /var/lib/graphite/data
Sudo mkdir -p /var/lib/graphite/statsd_config
Sudo mkdir -p /var/lib/graphite/log
```

Descargar grafito/statsd corriendo el siguiente comando.

```
docker pull graphiteapp/graphite-statsd
```

Ejecute el siguiente comando para configurar graphite/statsd, inícielo y configúrelo para que se inicie en el arranque del sistema copie y pegue todo junto en el terminal de comando lo siguiente.

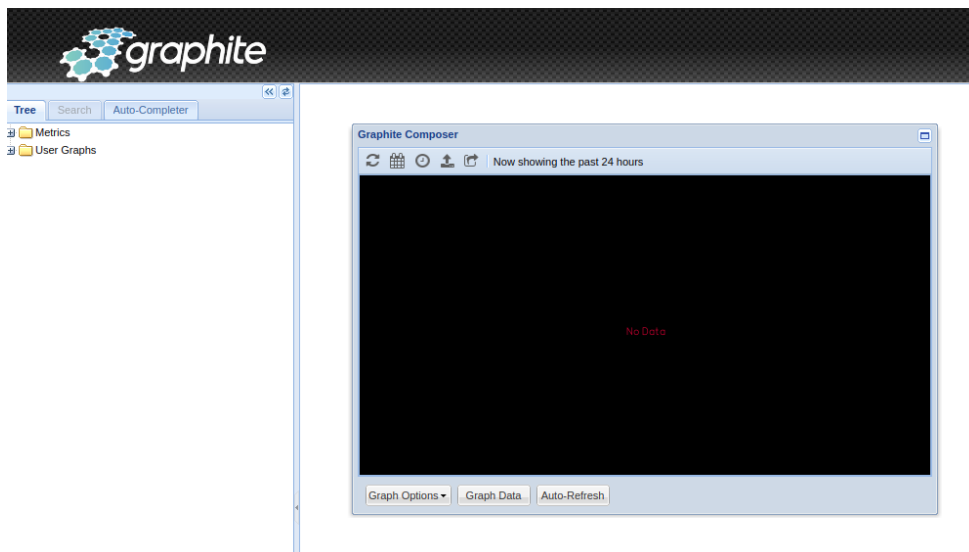
```
docker run -d\
```

```
--name graphite\  
--restart=always\  
-p 8080:80\  
-p 8125:8125/udp\  
-p 8126:8126\  
-v /var/lib/graphite/config:/opt/graphite/conf\  
-v /var/lib/graphite/data:/opt/graphite/storage\  
-v /var/lib/graphite/statsd_config:/opt/statsd/config\  
-v /var/lib/graphite/log:/var/log\  
graphiteapp/graphite-statsd
```

Para verificar su funcionamiento busque la ip local de su computador o raspberry y colóquelo en su navegador seguido el puerto 8080 como ejemplo la siguiente imagen.



Si todo salio bien al poner la ip 192.168.xxx.xxx: 8080 en el navegador debería salir lo siguiente.



## Goestools

Para ayudarle a configurar goestools para Vitality GOES, se han incluido archivos y ejemplos en la carpeta goestools-conf de este repositorio. Estos archivos están bastante cerca de los archivos sugeridos de "stock". No es necesario utilizar estas configuraciones exactas. Es posible que desee eliminar las secciones que no va a usar, y deberá hacer un "Buscar y reemplazar" para actualizar el directorio a donde desea que se almacenen sus productos GOES. Al final, su configuración debe configurarse de la siguiente manera: (goesrecv.conf) (goesproc-goesr.conf)

1 Si planea rastrear las estadísticas de decodificación satelital, asegúrese de que su archivo tenga un lugar definido donde está alojando Graphite / statsd

Modificar goesrecv.conf la siguiente línea statsd\_address con el siguiente comando

```
sudo nano goesrecv.conf
```

```
# Used by graphite/statsd
```

```
[monitor]
```

```
statsd_address = "udp4://localhost:8125"
```

2 Si va a habilitar la información de EMWIN, asegúrese de tener habilitado el controlador de emwin y que no ignore los archivos de texto. Busque el archivo en /usr/share/goestools/ goesproc-goesr.conf y modifique con el siguiente comando.

```
sudo nano /usr/share/goestools/ goesproc-goesr.conf
```

Ponga las siguientes líneas si no la tiene o modifíquelas en caso de estar de esta forma.

```
# EMWIN
```

```
[[handler]]
```

```
type = "emwin"
```

```
directory = "/home/pi/goes16/emwin"
```

3 En goesproc-goesr.conf, los controladores de imágenes deben tener el nombre de archivo terminado y el formato de archivo debe ser jpg.{time:%Y%m%dT%H%M%SZ}

4 Si bien toda la información de EMWIN estará en la misma carpeta, otros tipos de productos deben tener cada uno su propia carpeta. Por ejemplo, las imágenes del Canal 2 deben estar en su propia carpeta y no mezclarse con imágenes en falso color. No es



estándar mezclar tipos de productos en la misma carpeta, por lo que esto no debería ser un problema.

Además, goestools no guarda "Admin Text" debido a un cambio en la forma en que los satélites GOES envían el archivo hacia abajo. Para que se muestre este texto, vuelva a compilar goestools con este parche, coloque el siguiente comando:

```
sudo nano src/goesproc/handler_text.cc
```

Introduzca las siguientes líneas al final de todo el texto

```
// Parse time from file name.  
  
if (!goesrParseTextTime(text, time)) {  
  
    // Unable to extract timestamp from file name  
  
    // Unable to extract timestamp from file name. Use current system time instead.  
  
    auto rv = clock_gettime(CLOCK_REALTIME, &time);  
  
    ASSERT(rv >= 0);  
  
}  
  
}
```

## Instalación de Apache en Linux – Ubuntu y Debian

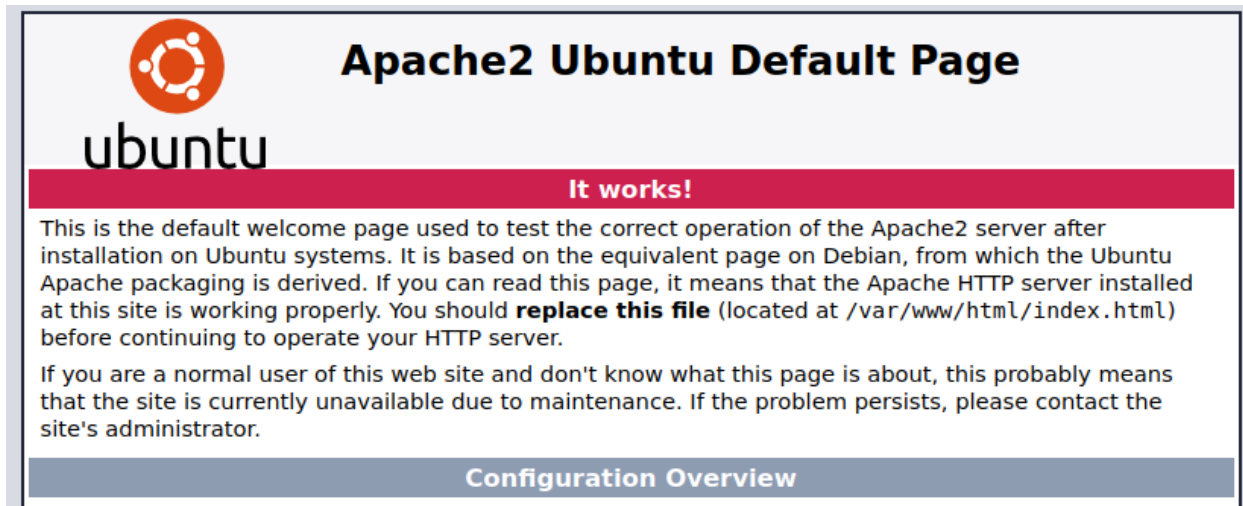
Suponiendo que esté en un servidor basado en Debian / Ubuntu, el siguiente comando debería instalar todas las dependencias que necesita, no es necesario actualizar Linux ya que se realizó al inicio de todas las configuraciones antes expuestas. Colocar el siguiente comando

```
sudo apt install apache2 php libapache2-mod-php
```

## Instalación de Apache en Windows

La forma más fácil de hospedar Vitality GOES en una caja de Windows es usar XAMPP (<https://www.apachefriends.org/>). Descargue e instale este software. Cuando se le solicita, las únicas partes que se necesitan son Apache y PHP. No olvide iniciar el servicio Apache en el panel de control de XAMPP antes de continuar.

Después de ver instalado Apache o Xampp en Linux o Windows coloque su ip local de su computador en el navegador y compruebe la página de prueba del servidor instalado en su pc como en la siguiente imagen.



## Instalación de Vitality GOES

### Linux

Para comenzar a instalar asegure que este como súper usuario o root con el siguiente comando.

```
sudo su
```

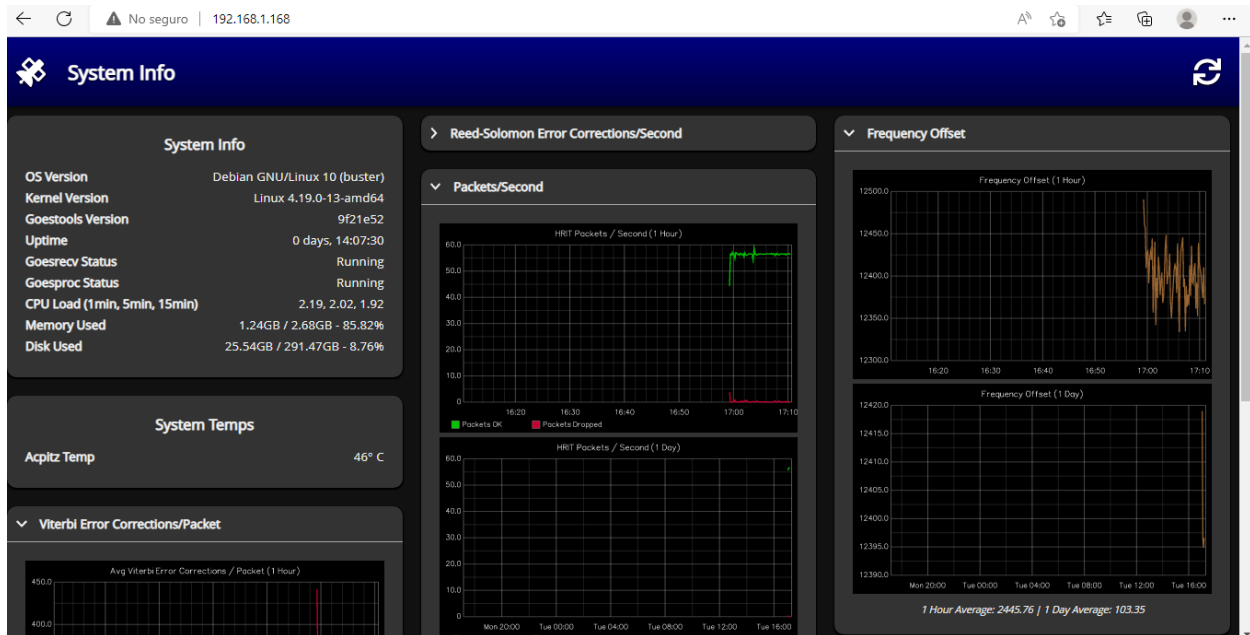
Luego coloque las siguientes líneas o comandos uno por uno.

- `sudo rm -rf /var/www/html`
- `git clone https://github.com/JVital2013/vitality-goes`
- `cd vitality-goes`
- `cp -r html /var/www/html`

Realice un reinicio con el siguiente comando a su pc

```
sudo reboot
```

Luego verifique que este correcto Vitality en su navegador poniendo su ip local de su pc o localhost como la siguiente imagen.



## Windows

Para empezar a usar Vitality GOES en Windows:

1. Descargue un zip del repositorio git de Vitality GOES ([enlace para los perezosos](#))
2. Extraer el zip
3. Eliminar el contenido de `C:\xampp\htdocs\`
4. Copie el contenido de `vitality-goes\html` en `C:\xampp\htdocs\`

**Goestools trabaja únicamente con Linux, por lo que tendría que buscar que goestools deposite las imágenes y datos en alguna carpeta compartida con Windows y direccionar a Vitality a buscar la información en dicha ubicación, cosa en la cual no redundaremos.**

Ejemplo en la siguiente imagen.

Este equipo > Disco local (C:) > xampp > htdocs				
Nombre	Fecha de modificación	Tipo	Tamaño	
config	18-06-2022 8:16	Carpeta de archivos		
lightgallery	18-06-2022 8:16	Carpeta de archivos		
opensans	18-06-2022 8:16	Carpeta de archivos		
splashscreens	18-06-2022 8:16	Carpeta de archivos		
videos	18-06-2022 8:16	Carpeta de archivos		
all.js	18-06-2022 8:16	Archivo JavaScript	1.234 KB	
dataHandler.php	18-06-2022 8:16	Archivo PHP	45 KB	
favicon.ico	18-06-2022 8:16	Icono	1 KB	
functions.php	18-06-2022 8:16	Archivo PHP	7 KB	
icon-192x192.png	18-06-2022 8:16	Archivo PNG	18 KB	
icon-256x256.png	18-06-2022 8:16	Archivo PNG	27 KB	
icon-384x384.png	18-06-2022 8:16	Archivo PNG	57 KB	
icon-512x512.png	18-06-2022 8:16	Archivo PNG	79 KB	
index.php	18-06-2022 8:16	Archivo PHP	3 KB	
manifest.json	18-06-2022 8:16	Archivo JSON	1 KB	
opensans.css	18-06-2022 8:16	Documento de ho...	3 KB	
script.js	18-06-2022 8:16	Archivo JavaScript	47 KB	
styles.css	18-06-2022 8:16	Documento de ho...	6 KB	

## Configuración de Vitality GOES

La configuración de Vitality GOES se almacena en /var/www/html/config. Puede comenzar con las configuraciones proporcionadas y a continuación, cambiarlas según sea necesario. La configuración se divide en los siguientes (archivos:html/config)

- **config.ini:** El archivo de configuración principal
- **emwin.ini:** almacena información sobre las imágenes emwin que desea mostrar. Este archivo no tiene ningún efecto en el texto emwin que se muestra
- **abi.ini:** Contiene información sobre las imágenes de disco completo. Si estás haciendo algún render de Sánchez, también lo pondría en este archivo.
- **meso.ini:** Contiene información sobre sus imágenes de mesoescala.

- **l2.ini:** Contiene información sobre sus productos ABI Nivel 2. Estas imágenes contienen información sobre la precipitación estimada, la temperatura de la superficie terrestre, la temperatura de la superficie del mar y más. Tenga en cuenta que goestools no los recibe a menos que la configuración de goesproc esté configurada para hacerlo. La configuración de ejemplo en este repositorio está configurada correctamente, pero si no está guardando estos archivos, simplemente deje l2.ini vacío.

Estos archivos ini se analizan con la función [php parse\\_ini\\_file](#), por lo que cualquier comentario debe comenzar con un punto y coma (;).

## config.ini

Este es el archivo de configuración principal y el mas importante. Es probable que deba configurarse cuando implemente Vitality GOES por primera vez. Se divide en las siguientes secciones:

- **graphiteAPI:** Apunta a su host de grafito. Debe incluir la ruta al final para que funcione correctamente. Si no estás usando Graphite, comenta esta línea/render/la ip es la local de su computador.

Ejemplo: graphiteAPI = <http://192.168.1.168:8080/render/>

- **emwinPath:** Apunte al repositorio emwin de su elección. Si está recogiendo tanto GOES West como East, puede usar cualquiera de los locaiton EMWIN. Comenta esta línea para desactivar completamente los datos de emwin (texto e imágenes)

Ejemplo: emwinPath = /home/pi/goes16/emwin

- **adminPath:** El directorio con el texto de administración que desea mostrar. Goestools debe ser parcheado, dejarlo como esta.

Ejemplo: adminPath = /path/to/goestoolsrepo/text

- **showSysInfo:** True si está en el mismo sistema que goestools. De lo contrario, configúrelo en false.

Ejemplo: showSysInfo = true

- **debug:** Se establece en true para habilitar los errores de PHP. Esto interrumpe las solicitudes AJAX dentro de Vitality GOES si hay algún error, así que solo

establezca esto en true si está depurando datos devueltos por DataHandler (solo usuarios avanzados).

Ejemplo: debug = false

## Rutas (Importante)

La sección Rutas es innecesaria, pero se recomienda que configure una ruta para cada satélite que esté recibiendo. Cada ruta definida en esta sección crea una variable que se puede utilizar en los archivos abi, meso y l2 ini.

Permite ser utilizado en el de cualquier imagen en los archivos abi, meso y l2 ini

Ejemplo de cómo debe estar configurado.

[paths]

GOES16 = /home/pi/

Si se aloja en Windows, esto sería algo así como GOES16 = C:\carpeta compartida/gooestols

## Ubicación (Valido solo para EE,UU)

Esta sección contiene información sobre su ubicación física. Si no está mostrando datos emwin, lo único que necesita configurar es lo anterior expuesto.

Puede encontrar una lista de las zonas horarias compatibles [aquí](#). timezone

Si está mostrando datos meteorológicos locales / EMWIN, esto es lo que significa cada una de las otras opciones. Tenga en cuenta que estas opciones también se pueden cambiar por navegador web cliente utilizando la pantalla "configurar ubicación" en Vitality GOES.

- **radarCode**: Las últimas 5 letras del archivo de radar para su región. En el directorio emwin, todos los archivos de radar terminan con RAD{radarCode}.GIF.

La imagen de radar que desea mostrar debe configurarse como una imagen disponible en emwin.ini

En el momento de escribir este artículo, los radarCodes válidos son:

- ALLAK (Alaska)
  - ALLGU (Guam)
  - ALLHU (Hawái)
  - ALLPR (Puerto Rico)
  - ALLUS (Todos nosotros)
  - GRTLK (Región de los Grandes Lagos)
  - NTHES (Región Noreste)
  - PACNW (Región Pacífico Noroeste)
  - PACSW (Región Pacífico Suroeste)
  - RCKNT (Región de las Montañas Rocosas del Norte)
  - RCKST (Región de las Montañas Rocosas del Sur)
  - REFUS (Estados Unidos Continentales)
  - SMSVY (Región del Valle del Río Mississippi sur)
  - STHES (Región Sureste)
  - STHPL (Región de las Llanuras del Sur)
  - UMSVY (Valle del Alto Misisipi)
- 
- **stateAbbr**: La abreviatura de la oficina de correos de su estado. También incluye cosas como relaciones públicas para Puerto Rico, AS para Samoa Americana, etc.
  - **wxZone**: La zona meteorológica de su ubicación. Por lo general, esta es la abreviatura de su estado, una Z y un número de 3 dígitos. Ejemplo: PAZ066. Puede usar la sección "Configurar ubicación" de Vitality GOES para resolver esto, o usar [este sitio](#) para buscar su ciudad. "Zona meteorológica" aparece en la parte superior derecha de esa página.
  - **orig**: La Oficina de Pronóstico del Servicio Meteorológico Nacional para su información meteorológica local. El código debe ser el indicativo de llamada de la oficina, más la abreviatura de estado de 2 letras.

Puede usar la sección "Configurar ubicación" de Vitality GOES para resolver esto o mirar [https://en.wikipedia.org/wiki/List\\_of\\_National\\_Weather\\_Service\\_Weather\\_Forecast\\_Offices](https://en.wikipedia.org/wiki/List_of_National_Weather_Service_Weather_Forecast_Offices). Por ejemplo, State College PA es "CTP", por lo que orig debe establecerse en CTPPA

- **rwrOrig**: Acepta el mismo tipo de código que , pero específicamente para la información de Resumen meteorológico regional (RWR; condiciones actuales en la interfaz Vitality GOES). Parece que el resumen meteorológico a veces es emitido por una oficina diferente al resto de su pronóstico. Use la sección "Configurar ubicación" dentro de Vitality GOES para resolver esto.orig
- **city**: El nombre de su ciudad/pueblo, exactamente como aparece en el Resumen Meteorológico Regional (RWR). La pantalla "Configurar ubicación" en Vitality GOES puede ayudarlo a resolver esto.
- **lat y lon**: Su latitud y longitud exactas. Esto solo se usa para determinar si se encuentra dentro de un área de alerta emitida por el NWS. Debe contener 2 puntos decimales para funcionar correctamentelon

Puede encontrar que la sección de ubicación es la parte más difícil de configurar de la configuración. Recomendaría dejarlo en sus valores predeterminados, luego usar la pantalla "Configurar ubicación" en Vitality GOES para determinar en qué se debe establecer cada valor para su ubicación. Una vez que lo tenga funcionando en el lado del cliente, configure los ajustes como appropriate en este archivo de configuración.

## Archivos abi.ini, meso.ini y l2.ini

**Nota:** recomendado no cambiar su configuración.

abi.ini, meso.ini y l2.ini funcionan igual. Estos archivos especifican qué imágenes desea mostrar en las secciones "Disco completo", "Imágenes de mesoescala" y "Gráficos de nivel 2" de Vitality GOES, respectivamente. Si un archivo no contiene imágenes configuradas, su sección se ocultará en Vitality GOES. *Si está recibiendo GOES-16, probablemente pueda usar este archivo tal cual.*

Para entender cómo funcionan estos archivos de configuración, veamos una sección de ejemplo:



```
[fdfc_16]
path = {GOES16}/goes16/fd/fc/
title = "GOES 16 - Color"
videoPath = GOES16FalseColor.mp4
```

- [fdfc\_16]: Un identificador único para la imagen. Esto puede ser cualquier cosa, pero debe ser único y no contener espacios.
- path: la carpeta que contiene todas las imágenes de un producto GOES en particular. En este ejemplo, utiliza la variable definida en la sección de config.ini{GOES16}Paths
- title: Cómo se etiquetará la imagen en Vitality GOES
- videoPath: el nombre del archivo de vídeo que contiene el timelapse de este producto. Los videos deben renderizarse por separado (por el [guión proporcionado](#) o cualquier otro medio), y deben mantenerse en la carpeta de Vitality GOES. Si no estás renderizando vídeos timelapse, comenta o elimina esta línea.html/videos

## emwin.ini

emwin.ini contiene información sobre los productos de imagen EMWIN que desea mostrar. Este archivo de configuración funciona igual que los archivos abi.ini, meso.ini y l2.ini con una excepción: es solo el nombre de archivo (incluida la extensión) del producto que desea mostrar. No se incluyen rutas de acceso en este archivo de configuración. *Es probable que no necesite configurar este archivo, a menos que esté personalizando lo que se muestrapath*

Una lista completa de los productos de imagen EMWIN se puede encontrar en [https://www.weather.gov/media/emwin/EMWIN\\_Image\\_and\\_Text\\_Data\\_Capture\\_Catalog\\_v1.3h.pdf](https://www.weather.gov/media/emwin/EMWIN_Image_and_Text_Data_Capture_Catalog_v1.3h.pdf). Utilice el GOES-N FN, pero debe estar en mayúsculas en esta configuración.

## Actualización de Vitality GOES

### Linux

Aceda como root con el comando:

```
sudo su
```

Ejecute los siguientes comandos en un terminal:

```
git clone https://github.com/JVital2013/vitality-goes
cd vitality-goes
rsync -av --exclude 'config' --exclude 'videos' html/ /var/www/html/
```

Si todavía tiene su repositorio de vitality-goes clonado de la última vez, también puede ejecutar antes de ejecutar rsync.

```
git pull
```

## Windows

1. Descargue un zip del repositorio vitality GOES más reciente ([enlace para los perezosos](#))
2. Extraer el zip
3. Abra una línea de comandos dentro del directorio vitality-goes-main del zip extraído
4. Ejecute el siguiente comando: `robocopy html C:\xampp\htdocs /MIR /R:0 /W:0 /XD videos config`

## Scripts adicionales

Vitality GOES viene con una serie de scripts para mejorar y ampliar su funcionalidad.

Es opcional implementar cualquiera de estos scripts, pero algunos como Cleanup-EmwinText son muy recomendables. Otros, como los guiones de creación de video, son un poco hackeados y podrían usar algunos retoques. Pero bueno, ¡funcionan muy bien para mí!

Todos los scripts están en la carpeta de este repositorio.scripts/

## Cleanup-EmwinText.sh

- *Paquetes de sistema adicionales requeridos: zip*
- *Modificar la línea 4 antes de ejecutarla: `cd /path/to/goestoolsrepo/emwin`*

Cuando goesproc está configurado para guardar información de texto EMWIN, guarda *muchos* archivos de texto, ¡aproximadamente 30,000 al día! Si bien estos

archivos probablemente no llenarán su disco duro, ralentizarán todo debido a la cantidad de archivos que deben analizarse.

Cleanup-EmwinText.sh resuelve el problema comprimiendo todos los archivos de texto EMWIN de ayer en una carpeta ZIP. Este script debe configurarse para ejecutarse todos los días entre las 1600 y las 2330 UTC.

## CreateVideos-ABI.sh

- *Paquetes de sistema adicionales requeridos: ffmpeg imagemagick*
- *Modifique las líneas 2 a 6 antes de ejecutarlas, que establecen las variables srcDir, videoDir, imgSource, vidName y resizeMode*

CreateVideos-ABI.sh crea videos timelapse de productos de imagen ABI. De forma predeterminada, representan lapsos de tiempo de la última semana a 15 cuadros por segundo. Los vídeos se almacenan en la carpeta de Vitality GOES para que puedan ser vistos en el cliente web.html/videos

Para configurar este script, es importante comprender cómo interactúan entre sí el , y las variables. Estas variables son matrices, y cada una de las matrices está "alineada" entre sí. Por ejemplo, el primer elemento en , , y son las configuraciones para el primer vídeo. El segundo elemento de cada matriz es la configuración para el siguiente video, el tercer elemento de cada matriz es la configuración para el tercer video, y así sucesivamente. imgSource vidName resizeMode imgSource vidName resizeMode

- imgSrc: especifica el origen de los fotogramas de cada vídeo. Esto debería ser similar a [la ruta en los archivos de configuración abi.ini, meso.ini y l2.ini](#)
- vidName: Especifica el nombre del MP4 que desea crear, sin la extensión MP4. Aparte de la extensión que falta, esto debería coincidir con [videoPath en sus archivos de configuración abi.ini, meso.ini y l2.ini](#)
- resizeMode: Dado que los satélites GOES pueden enviar imágenes de muy alta resolución, desea reducir la escala de algunas de ellas. El modo de cambio de tamaño especifica cómo cambiar el tamaño de las imágenes antes de convertirlas en vídeos:
  - 0: Cambie el tamaño de las imágenes a 1356x1356. Bueno para imágenes de disco completo
  - 1: Cambie el tamaño de las imágenes a 1000x1000. Bueno para algunas imágenes no CMIP de nivel II e imágenes de mesoescala
  - 2: Cambie el tamaño de las imágenes a 1402x954. Bueno para los compuestos de Sánchez
  - 3: No cambie el tamaño de la imagen.

## CreateVideos-EMWIN.sh

- *Paquetes de sistema adicionales requeridos: ffmpeg imagemagick rename*
- *Modifique las líneas 2 a 6 antes de ejecutarlas, que establecen las variables srcDir, videoDir, codeName, videoName e imgconvert*

CreateVideos-EMWIN.sh crea videos timelapse de productos de imagen EMWIN. De forma predeterminada, representan lapsos de tiempo de la última semana a 15 cuadros por segundo. Los vídeos se almacenan en la carpeta de Vitality GOES para que puedan ser vistos en el cliente [web.html/videos](http://web.html/videos)

Para configurar este script, es importante comprender cómo interactúan entre sí el , y las variables. Estas variables son matrices, y cada una de las matrices está "alineada" entre sí. Por ejemplo, el primer elemento en , , y son las configuraciones para el primer vídeo. El segundo elemento de cada matriz es la configuración para el siguiente video, el tercer elemento de cada matriz es la configuración para el tercer video, y así sucesivamente. codeName videoName imgconvert codeName videoName imgconvert

- codeName: especifica el nombre de archivo EMWIN de los fotogramas de cada vídeo. Esto debería ser similar a [la ruta en su archivo de configuración emwin.ini](#), solo que sin la extensión del archivo
- videoName: Especifica el nombre del MP4 que desea crear, sin la extensión MP4. Aparte de la extensión que falta, esto debería coincidir con [videoPath en sus archivos de configuración emwin.ini](#).
- imgconvert: especifica el formato de archivo de las tramas de origen. Esto debería coincidir con la extensión de las tramas de origen

## Sanchez.sh

- *Paquetes de sistema adicionales requeridos: xplanet*
- *Software adicional requerido (no del sistema): [sanchez](#)*
- *Modifique las líneas 2-7 antes de ejecutarlas, que establecen las variables srcPath16, srcPath17, dstPath16, dstPath17, dstPathComposite y sanchezPath*

Sanchez.sh es un script que automatiza los renders sánchez de tus capturas geoestacionarias. Para usarlo, xplanet primero debe estar configurado. Instale xplanet como lo haría normalmente para su distribución y descargue / extraiga Sanchez. A continuación, edite el archivo de configuración predeterminado de xplanet (at en la

mayoría de las distribuciones). Cambie la sección para que solo diga

```
esto:/var/share/xplanet/config/config[earth]
```

```
[earth]
```

```
"Earth"
```

```
map=/path/to/sanchez/Resources/world.200411.3x10848x5424.jpg
```

```
night_map=/path/to/sanchez/Resources/world.lights.3x10848x5424.jpg
```

El script actualmente crea 3 cosas: imágenes de color falso GOES-16, imágenes de color falso GOES-17 y compuestos de GOES-16 y 17. El script también realizará cualquier renderizado "back" que pueda haber perdido debido a que el script está deshabilitado, no se puede ejecutar u otros problemas. Este script deberá actualizarse una vez que GOES-18 tome el lugar de GOES West.

Cuando haya terminado, habilite las secciones Sánchez en su [archivo de configuración abi.ini](#) para mostrar sus nuevos renders elegantes.

## Delete-Old.sh

*Modifique la línea 2 antes de ejecutarla, que establece la ubicación de los archivos GOES.*

Delete-Old.sh elimina todos los archivos de texto ABI, EMWIN, NWS y admin que tienen más de 2 semanas de antigüedad. Ejecuto esto en mi estación terrestre manualmente después de verificar que mis archivos fuera de línea estén actualizados.

## Monitor-Recordings.sh

- *Paquete de sistema adicional requerido: inotify-tools*
- *Modifique la línea 2 para establecer la ubicación de los archivos GOES antes de ejecutarlos*

Monitor-Recordings.sh archivo apenas constituye un script, pero se puede usar para monitorear archivos a medida que los guarda goestools. Goesproc genera esta información, pero si está ejecutando goesproc como servicio, la información está oculta. Encuentro que este script hace un buen trabajo al verificar que goesproc realmente está procesando datos.

Ejecute manualmente según sea necesario.

## Archivo cron.d de ejemplo

Los sistemas Linux le permiten programar tareas mediante la creación de un archivo con una lista de tareas a ejecutar. Así es como tengo mi archivo cron configurado, en `:/etc/cron.d//etc/cron.d/goes`

```
0 0 * * * youruser /path/to/vitality-goes/scripts/CreateVideos-ABI.sh
0 5 * * * youruser /path/to/vitality-goes/scripts/CreateVideos-EMWIN.sh
55 11 * * * youruser /path/to/vitality-goes/scripts/Cleanup-EmwinText.sh
25,55 5-23 * * * youruser /path/to/vitality-goes/scripts/Sanchez.sh
```

## Compartir Vitality GOES a través de Internet (Importante)

Técnicamente hablando, no debería haber ningún problema con abrir un puerto en su enrutador para que se pueda acceder a Vitality GOES a través de Internet. En la práctica, **no** lo recomiendo. Por un lado, Vitality GOES no ha sido examinado por profesionales de la seguridad. Tuve en cuenta la seguridad al codificarlo, pero no estoy haciendo promesas. Si lo comparte a través de Internet, asegúrese de que su servidor web esté respetando los archivos. No todos los servidores web lo hacen de forma predeterminada..`htaccess`

Además, algunos archivos producidos por goestools pueden ser extremadamente grandes. Asegúrese de que su Internet tenga suficiente velocidad de carga para manejarlo y que las reglas de su firewall estén configuradas correctamente. Lo último que quieres es que alguien te haga un DDoS consultando 1.000 imágenes de disco completas a la vez...

---

## Nota de traducción.

Este gran material fue traducido al español con el consentimiento de su creador. No se modificaron todos los aspectos, solo aquellos que fueron más necesario para llegar al funcionamiento más directo del programa vitality para los Goes. Se irán corrigiendo cualquier información cuando el autor determine o considere oportuno.

Traducido en Valdivia, Chile por XQ6DLW. Demys

