



GITUD
GRUPO DE INVESTIGACIÓN DE TELECOMUNICACIONES
DE LA UNIVERSIDAD DISTRITAL

LoRa-GTCP: LoRa- Georeferencing Terrestrial Control Points.

Versión: 1.0.2



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

MANUAL TÉCNICO

AUTORES:

**ANDRES CAMILO GONZALEZ
HERNANDEZ**
Ingeniero Electrónico
acgonzalezh@correo.udistrital.edu.co

ELVIS EDUARDO GAONA G.
Docente Facultad de Ingeniería
egaona@udistrital.edu.co

VICTOR DANIEL ANGULO.
Docente Facultad Ingeniería
vdangulom@correo.udistrital.edu.co

2020

TABLA DE CONTENIDIO

TABLA DE CONTENIDIO	2
INDICE DE FIGURAS	3
INDICE DE TABLAS	4
1. INTRODUCCIÓN	5
2. OBJETIVOS DEL SISTEMA.....	6
2.1 Objetivo general del sistema.....	6
2.1 Objetivos secundarios	6
3. REGLAS DE NEGOCIO	7
4. FUNDAMENTACIÓN DE LA TECNOLOGÍA UTILIZADA.....	8
5. CRITERIOS TÉCNICOS DEL SISTEMA	9
5.1 Robustez E Integridad	9
5.2 Extensibilidad	9
5.3 Desempeño	10
5.4 Portabilidad Y Usabilidad	10
6. DESPLIEGUE Y CONFIGURACIÓN DEL ENTORNO DE LA APLICACIÓN.....	12
6.1 Iniciar aplicación Django	12
7. USUARIOS	13
8. REQUISITOS.....	15
8.1 Requisitos Funcionales.....	15
8.1.1 Interfaces Externas	15
8.1.2 Funciones	15
8.2 Requisitos No Funcionales	15
8.3 Requisitos de software y Hardware.	16
8.3.1 Requisitos de software:	16
8.3.2 Requisitos de Hardware:	16
9. VISTA FUNCIONAL	17
10. VISTA LÓGICA DEL SISTEMA.....	19
10.1 Modelo lógico de datos.....	19
10.2 Diagrama de despliegue	20

INDICE DE FIGURAS

Figura 1. Tecnología utilizada para la generación de RED LORA GCP8

Figura 2. Diagrama Casos de uso del Sistema9

Figura 3. Distribución de la aplicación en archivos y carpetas 11

Figura 4. Vista principal de la aplicación posterior al proceso de login 11

Figura 5. Inicializando aplicación web 12

Figura 6. Creación de usuarios en la aplicación web 13

Figura 7. Creación de usuarios en la aplicación web - Admin 14

INDICE DE TABLAS

<i>Tabla 1. Requerimientos de software.</i>	<i>10</i>
<i>Tabla 2. Descripción de los distintos roles de usuario.</i>	<i>14</i>
<i>Tabla 3. Actualizar Datos.</i>	<i>17</i>
<i>Tabla 4. Detalles Datos.</i>	<i>17</i>
<i>Tabla 5. Detalles Datos</i>	<i>18</i>
<i>Tabla 6. Descargar Datos.</i>	<i>18</i>

1. INTRODUCCIÓN

LoRa-GTCP, se encarga de recopilar los datos entregados por el hardware del prototipo registrador de parámetros de posicionamiento de los nodos, organizarlos y presentarlos de una manera sencilla para el usuario, que son escritos por el módulo de comunicaciones en un archivo de texto plano. Esta aplicación no está en capacidad de generar la corrección de puntos sobre una imagen tomada con drones. Este sistema trae beneficios para el usuario, ya que presenta la basta información registrada de una manera sencilla y útil y facilita el análisis de la misma.

LoRa-GTCP está compuesto por dos estructuras fundamentales, el back-end: compuesto por el servidor de la herramienta, que es el encargado de realizar operaciones de almacenamiento, modificación, borrado y envío de los datos, y el front-end de la herramienta: una aplicación que implementa módulos estadísticos y de analítica visual, y operaciones de descarga de datos. Esta aplicación es parte de los resultados del proyecto de investigación Evaluación de la condición de vegetación en cultivos de papa a partir del análisis de imágenes obtenidas con drones, financiado por el Centro de Investigación y desarrollo científico (CIDC) de la Universidad Distrital Francisco José de Caldas.

En el repositorio de github (<https://github.com/ElvisGaona/LoRaGTCP>) está publicado los archivos necesarios para ejecutar el software, copia de los manuales técnicos y del programador y el registro en la Dirección Nacional de Derecho de Autor.

2. OBJETIVOS DE LA APLICACIÓN

2.1 Objetivo general

Almacenar, analizar y representar la información provista por el hardware de posicionamiento de puntos de control terrestre.

2.1 Objetivos secundarios

- Tener la capacidad de interactuar con el hardware a través de un Access Point
- Visualizar el estado general de la red y una posición aproximada.
- Visualizar los detalles de información sobre un nodo.
- Poder descargar la información de los nodos.

3. REGLAS DE NEGOCIO

- LoRa-GTCP s debe comunicarse con la base de datos para realizar operaciones de lectura y escritura.
- LoRa-GTCP funciona en todos los sistemas operativos, se recomienda el uso de explorador Google Chrome 51.0.1 o superior.

4. FUNDAMENTACIÓN DE LA TECNOLOGÍA UTILIZADA

El desarrollo de la aplicación LoRa-GTCP se estructuró sobre Python (Django, Flask), así como uso de html5 y Bootstrap para la visualización de los datos, así como para operaciones de lectura y escritura en la base de datos, como muestra la figura 1. Dicha tecnología se eligió debido a la necesidad de comunicación asíncrona entre la base de datos, el servidor y la aplicación web, así mismo la necesidad de visualización mediante diagramas estadísticos, visualización mediante tablas dinámicas y organización de la información.



Figura 1. Tecnología utilizada para la generación de RED LORA GCP

5. CRITERIOS TÉCNICOS DEL SISTEMA

5.1 Robustez E Integridad

El uso de tecnologías basadas en Python, permite que LoRa-GTCP realice múltiples operaciones tanto de consulta, escritura y representación estadística simultáneamente sin afectar el flujo de ejecución de la herramienta, presentar lentitudes o afectar la información compartida desde y hacia el sistema.

5.2 Extensibilidad

LoRa-GTCP se basa en programación modular, donde son 4 sus módulos, los cuales se pueden evidenciar en la Figura 2 que corresponde a los casos de uso de la aplicación. Al ser una aplicación desarrollada visualmente en django y en su parte back-end con django, si lo que se busca es agregar nuevas funcionalidades al sistema simplemente se debe agregar un nuevo módulo a la parte front-end o back-end de la aplicación.

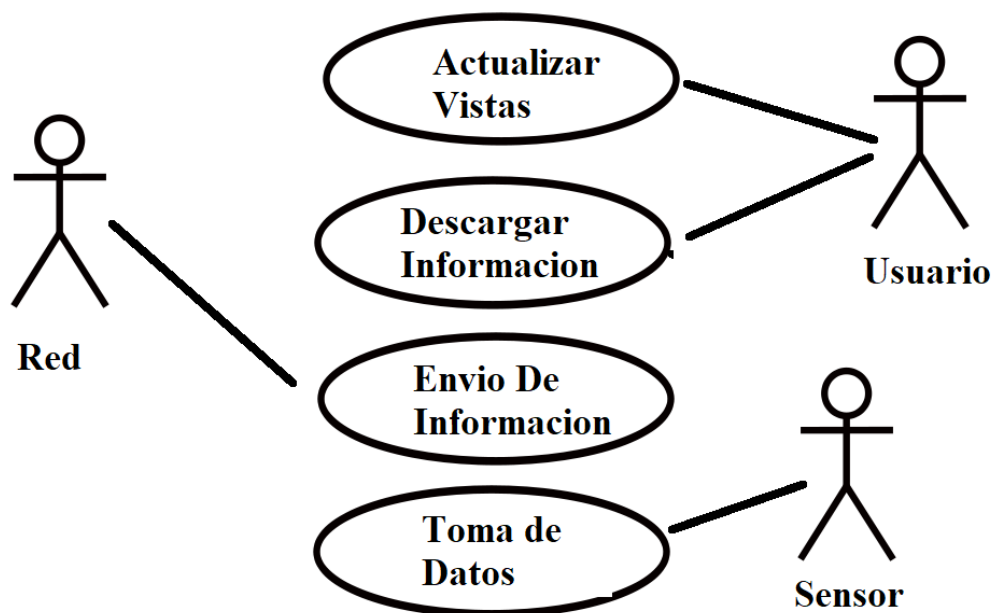


Figura 2. Diagrama Casos de uso del Sistema

5.3 Desempeño

La tecnología utilizada para LoRa-GTCP, utiliza comunicación asíncrona, que permite la ejecución de múltiples tareas al tiempo sin necesidad de que el sistema deba esperar el resultado de una de ellas para continuar con la otra, el sistema retorna la respuesta de una tarea específica. De igual forma los tiempos de respuesta de la aplicación no exceden los 2 segundos de tiempo y el consumo de memoria RAM al ser una aplicación web, es regulado por el navegador.

5.4 Portabilidad Y Usabilidad

LoRa-GTCP requiere de los siguientes requerimientos de software, adjuntos a un sistema operativo basado en Linux para el servidor que comprende la base de datos, Python back-end , así mismo para la base de datos SQL de django. Ver tabla 1.

Tabla 1. Requerimientos de software.

Tecnología	Versión
Python	>=3.5
Pip (python framework)	8.0.1
Django (python library)	2.2.4
Django-bootstrap4	Cualquiera
Django-tables2	Cualquiera
Pynmea2	Cualquiera

Al tener instaladas las librerías y los frameworks correspondientes, tanto para la parte de back-end como para la parte front-end se debe ejecutar el siguiente comando en una terminal ó consola del sistema operativo:

```
pip install -r requirements.txt
```

donde requirements.txt tiene las dependencias a instalar

La estructura del sistema para su uso se divide en 3 carpetas distintas como muestra la figura 3.

lora	1/10/2019 12:01 p....	Carpeta de archivos	
nodos	6/11/2019 8:17 a. m.	Carpeta de archivos	
static	17/12/2019 3:57 p....	Carpeta de archivos	
db.sqlite3	1/07/2020 11:06 a. ...	Archivo SQLITE3	140 KB
dotfile.dot	14/05/2020 12:17 ...	Microsoft Word 9...	19 KB
manage.py	1/10/2019 12:01 p....	Archivo PY	1 KB
requirements.txt	12/11/2019 9:02 p....	Documento de tex...	1 KB

Figura 3. Distribución de la aplicación en archivos y carpetas

De esta forma se puede hacer uso de la aplicación localmente.

En cuanto a la usabilidad del sistema, visualmente se utilizó una interfaz basada en grillas, con diseño adaptable al dispositivo, estructurada con un menú lateral y vertical y componentes gráficos que ayudan a hacer la mejor experiencia del usuario. Ver figura 4.

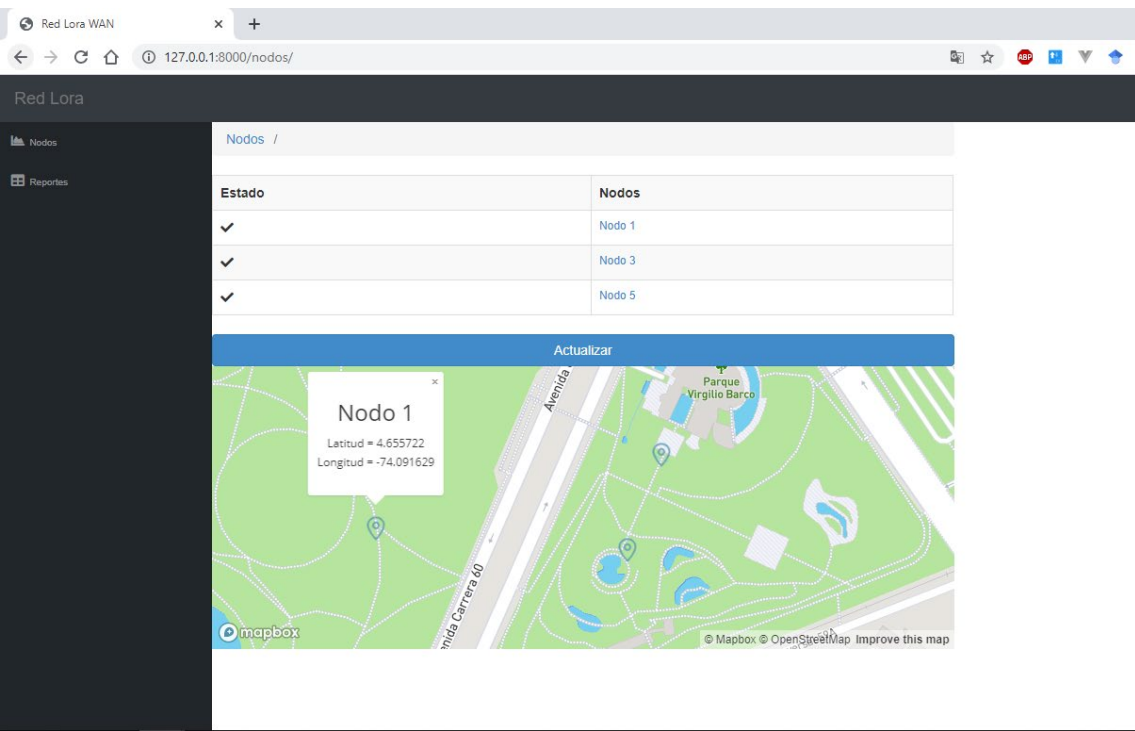


Figura 4. Vista principal de la aplicación posterior al proceso de login

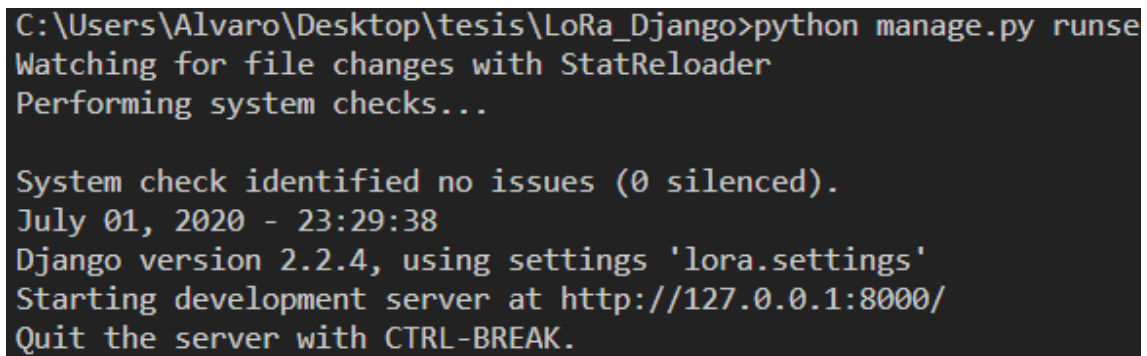
6. DESPLIEGUE Y CONFIGURACIÓN DEL ENTORNO DE LA APLICACIÓN.

6.1 Iniciar aplicación Django

Sobre la ubicación de nuestros archivos ejecutamos el comando

```
python manage.py runserver 192.168.0.20:8000
```

donde podemos especificar el puerto y la dirección IP sobre la que va a correré nuestro aplicativo, en la figura 5 la ubicación IP por defecto que es 127.0.0.1:8000 ya que no se especifica de dirección sobre la que vamos a correr el aplicativo.



```
C:\Users\Alvaro\Desktop\tesis\LoRa_Django>python manage.py runse
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 01, 2020 - 23:29:38
Django version 2.2.4, using settings 'lora.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura 5. Inicializando aplicación web

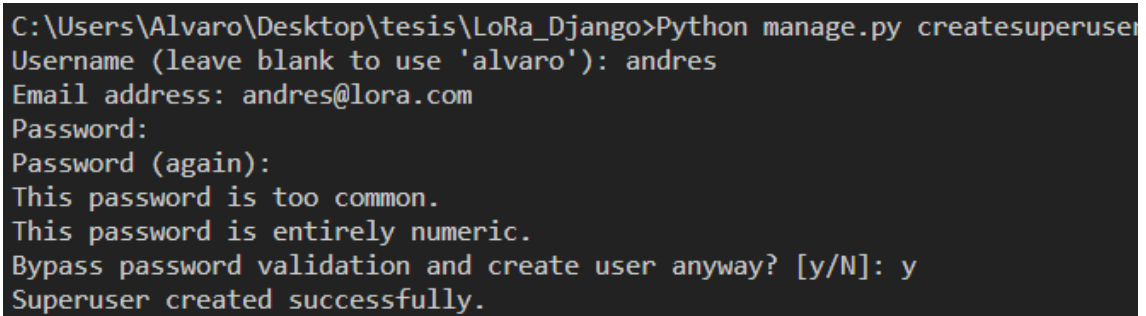
7. USUARIOS

Ya que se está llevando a cabo de manera local la implementación del aplicativo, para la creación de usuario se utiliza el siguiente comando:

```
Python manage.py createsuperuser
```

Donde especificaremos el nombre de usuario que queramos utilizar, así como su respectiva contraseña y un correo

En la figura 6 se muestra la sesión de inicio del aplicativo web.



```
C:\Users\Alvaro\Desktop\tesis\LoRa_Django>Python manage.py createsuperuser
Username (leave blank to use 'alvaro'): andres
Email address: andres@lora.com
Password:
Password (again):
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Figura 6. Creación de usuarios en la aplicación web

Como también se pueden generar usuario utilizando la web administrativa generada automáticamente al crear el proyecto de Django a la cual accederemos añadiendo /admin a la dirección IP que estemos utilizando para correré nuestro aplicativo web, para lo cual, si es necesario haber realizado el paso anterior, en la figura 7 se muestra la web administrativa de django donde podemos observar los usuarios creados y su información.

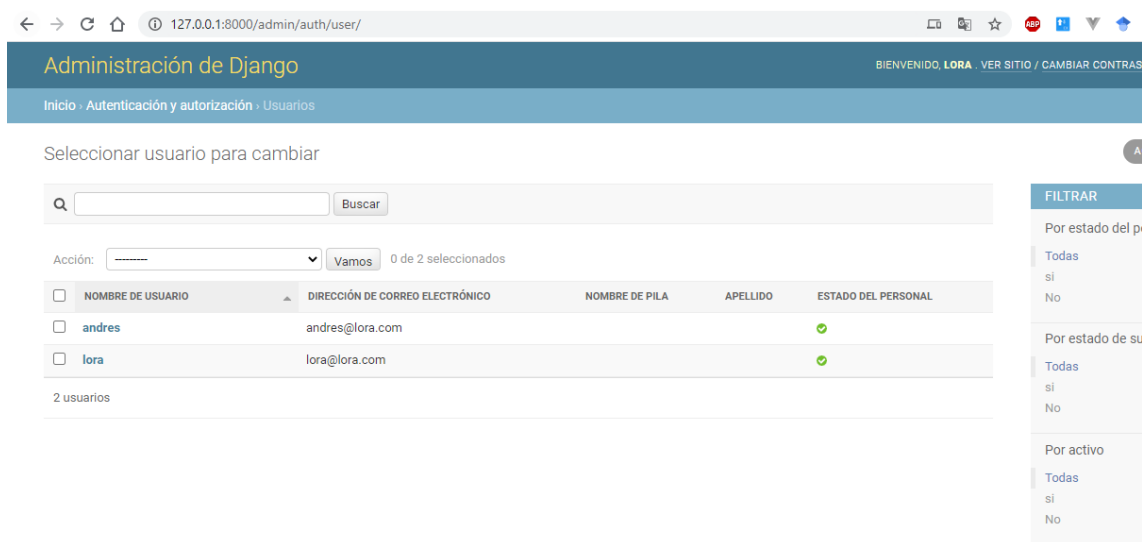


Figura 7. Creación de usuarios en la aplicación web - Admin

En la tabla 2 se describe los distintos roles de usuarios.

Tabla 2. Descripción de los distintos roles de usuario.

ACTOR	DESCRIPCIÓN
usuarios	Posee total acceso a las funcionalidades del sistema, puede ver, descargar y actualizar los datos mostrado en la interfaz

8. REQUISITOS

A continuación, se enlistan los requisitos y se detallan los requisitos.

8.1 Requisitos Funcionales

8.1.1 Interfaces Externas

- Módulo de comunicación WI-FI
- Módulo de comunicación LoRa
- Módulo de posicionamiento GNSS

8.1.2 Funciones

- Tomar datos generados por los módulos de comunicaciones LoRa y GNSS sobre un directorio.
- Visualizar los datos generales sobre el ultimo estado de la red.
- Visualizar datos específicos de un nodo de la red, así como la información de posición del mismo.
- Generar archivos para la descarga de la información de un nodo.

8.2 Requisitos No Funcionales

- Visualización a través de ambiente WEB, un mapa de la ubicación de los nodos
- Almacenamiento de datos de la posición en la base de datos ya que la información principal queda guardada en archivos de texto plano.
- Soporta sesiones múltiples de usuarios consumiendo la aplicación.

8.3 Requisitos de software y Hardware.

8.3.1 Requisitos de software:

- Navegador chrome browser versión 51.0.1 o superior.
- Python 3.3 o superior.
- Django
- Django tables
- Pynmea2

8.3.2 Requisitos de Hardware:

- Memoria RAM de al menos 2GB
- 1GB disponible de espacio para almacenamiento en disco duro.
- Unidad de WI-FI

9. VISTA FUNCIONAL

En las tablas 3 a 6, se muestra los diferentes modos del aplicativo según el usuario.

Tabla 3. Actualizar Datos

Título:	Actualizar Datos
Actor:	Usuario
Escenario 1:	<ol style="list-style-type: none">1. El usuario en la ventana principal selecciona el botón que dice “Actualizar”2. El aplicativo web busca sobre el directorio donde se guardan por defecto la información de la red.3. Lee cada archivo en el directorio obteniendo los datos de los diversos nodos.4. Actualiza o Crea un nodo en la base de datos SQL dependiendo de la información encontrada en los archivos de texto plano.

Tabla 4. Detalles Datos.

Título:	Detalles Datos
Actor:	Usuario
Escenario 1:	<ol style="list-style-type: none">1. El usuario en la ventana principal selecciona un nodo2. El aplicativo web busca en la base de datos en base al número de nodo3. El aplicativo Renderiza una nueva vista donde se muestran las especificaciones de una nodo en la red , ya sean sus parámetros de posición o de transmisión de datos.

Tabla 5. Detalles Datos

Título:	Ver Datos
Actor:	Usuario
Escenario 1:	<ol style="list-style-type: none">1. El usuario en la ventana de detalles de nodo selecciona el botón “Ver Datos”2. El aplicativo web carga sobre un cuadro de texto editable el archivo de texto plano que genera la red LoRa

Tabla 6. Descargar Datos.

Título:	Ver Datos
Actor:	Usuario
Escenario 1:	<ol style="list-style-type: none">1. El usuario en la ventana de detalles de nodo selecciona el botón “Descargar”2. Sobre el dispositivo que se esté realizando la visualización del aplicativo se generar un archivo con el nombre “nodo” mas numero el numero de nodos en formato txt

10. VISTA LÓGICA DEL SISTEMA

10.1 Modelo lógico de datos

En la figura 24 se muestra el modelo lógico de datos, implementados por medio de modelos en django, donde la tabla nodos representa la información de los diversos nodos disponibles en la red con su información, la tabla maestroES representa si el nodo actual es el nodo maestro de la red, y lista se refiere a la ubicación de los archivos de texto plano que se han generado.

lista	
	AutoField
ivo	FileField
bre	TextField
mo	IntegerField

maestroES	
id	AutoField
index	IntegerField
maestro	BooleanField

nodos	
id	AutoField
AnchoBanda	IntegerField
CargaUtil	TextField
Corriente	IntegerField
EstadoLora	BooleanField
FuerzaSenal	IntegerField
NumeroNodo	IntegerField
NumeroSatelites	IntegerField
PaquetesEnviados	IntegerField
PaquetesRecibidos	IntegerField
TiempoEnvio	FloatField
altitud	FloatField
dilucion	FloatField
estadoGnss	BooleanField
fixQuality	IntegerField
latitud	FloatField
longitud	FloatField

9: Modelo lógico de los datos

Figura

10.2 Diagrama de despliegue

En la figura se muestra el diagrama de despliegue del aplicativo

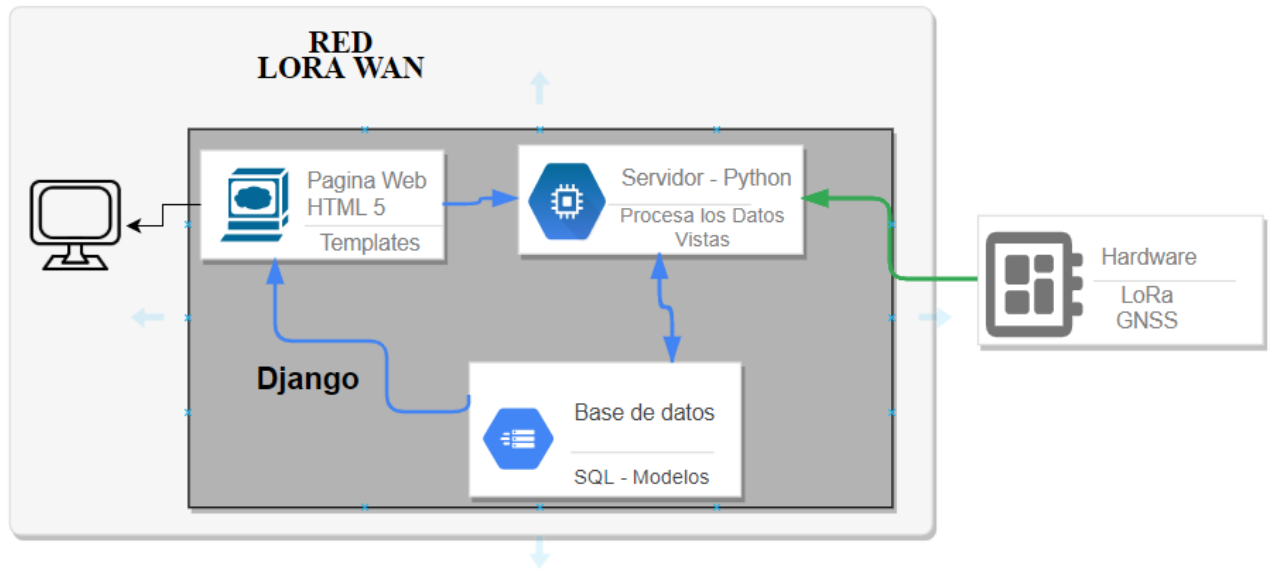


Figura 10: Diagrama de despliegue