



MANUAL TÉCNICO



Contenido

[1. Requisitos del entorno 4](#_Toc53988308)

[2. Creación del entorno 4](#_Toc53988309)

[2.1 Instalación de Python3 4](#_Toc53988310)

[2.1.1 Python 4](#_Toc53988311)

[2.1.2 Descarga de paquetes 4](#_Toc53988312)

[2.1.3 Instalación *offline* de paquetes 5](#_Toc53988313)

[2.1.4 Ejecución y verificación 5](#_Toc53988314)

[2.2 Instalación de Apache Airflow 6](#_Toc53988315)

[2.2.1 Apache Airflow 6](#_Toc53988316)

[2.2.2 Esquema de funcionamiento 6](#_Toc53988317)

[2.2.3 Despliegue e Instalación 6](#_Toc53988318)

[2.2.4 Ejecución y Verificación 8](#_Toc53988319)

[2.2.5 Versiones 9](#_Toc53988320)

[3. Desarrollos realizados 10](#_Toc53988321)

[3.1 Ingesta de los datos 10](#_Toc53988322)

[3.1.1 Ingesta de los artículos del BOE 10](#_Toc53988323)

[3.1.2 Ingesta de los artículos de Aragón 10](#_Toc53988324)

[3.2 Conversión de los documentos 10](#_Toc53988325)

[3.2.1 De PDF a TXT 10](#_Toc53988326)

[3.2.2 De XML a TXT 10](#_Toc53988327)

[3.2.3 De HTML a TXT 10](#_Toc53988328)

[3.3 Extracción de la información 10](#_Toc53988329)

[3.3.1 Extracción mediante reglas 10](#_Toc53988330)

[3.3.2 Extracción mediante el modelo NER 10](#_Toc53988331)

[3.3.3 Modelo jerárquico 10](#_Toc53988332)

[3.4 Inserción en la base de datos 10](#_Toc53988333)

[4. Base de datos 11](#_Toc53988334)

[4.1 Estructura 11](#_Toc53988335)

[4.2 Tablas 11](#_Toc53988336)

[4.3 Campos 11](#_Toc53988337)

Control de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor | Descripción |
| 1.0 |  | Hiberus | Primera versión (incompleta) del documento |

# Requisitos del entorno

En esta sección se van a enumerar los diferentes requisitos que ha de cumplir el entorno para el correcto funcionamiento del programa.

# Creación del entorno

En esta sección se van a detallar las instalaciones realizadas para crear el entorno descrito en la sección 1.

## Instalación de Python 3

### Python

Python es un lenguaje de programación interpretado y multiparadigma que soporta programación imperativa, orientada a objetos y funcional. Es uno de los lenguajes más utilizados de la actualidad (Octubre de 2020), ya que se diseñó para aumentar la productividad y la legibilidad. Además, a lo largo de los años se han publicado diferentes módulos y *frameworks* que, unido a su sencillez y su velocidad de desarrollo, lo han convertido en una de las mejores opciones a la hora de desarrollar aplicaciones relacionadas con el campo del dato.

Las versiones principales de Python son Python2 y Python3. La versión 2 se quedó sin soporte cuando empezó el año 2020, mientras que la 3 sigue desarrollándose y mejorándose. De manera general, los sistemas operativos basados en Linux llevan preinstaladas ambas versiones por defecto, pero no tiene por qué ser así en distribuciones CentOS o Red Hat.

El objetivo de este capítulo es describir el proceso de instalación de Python 3.6.9 en un servidor sin conexión a Internet. Al estar aislados, estos servidores no pueden descargarse los archivos de instalación, por lo que estos deberán obtenerse previamente y copiarlos/disponibilizarlos sobre dichos servidores, para que estos puedan ejecutar una instalación *offline*.

### Descarga de paquetes

En este apartado se describen diferentes formas de obtener los paquetes requeridos para instalar python3.6 en sistemas con las siguientes características:

* Sistemas Linux basados en CentOS/Red Hat/similares.
* Sistemas con arquitectura x86\_64.
* Sistemas sin versiones ya instaladas de Python3.6.

El procedimiento, en todas sus opciones, acaba resultando en la obtención de los ficheros .rpm (Red Hat Package Manager) utilizados posteriormente para instalar Pyhton3.6.

#### Paquetes principales

Los programas generalmente poseen dependencias con otros paquetes/programas. Esto implica que en el proceso de instalación dichas dependencias deben ser cumplidas para que el procedimiento finalice correctamente.

El paquete principal necesario para instalar Pyhon3.6, así como sus dependencias directas se muestran en la tabla 2.1.

#### Descarga desde los repositorios

#### Descarga mediante gestor de paquetes yum

#### Descargar todas las dependencias

En apartados previos se ha explicado como instalar python3 junto con sus dependencias directas. Sin embargo, es posible descargar todas las dependencias simultáneamente, si así se desea:

$ cd ~

$ mkdir python

$ mkdir python/packages python/install-root

$ yum install --downloadonly \

--installroot=~/python/install-root/ \

--releasever=7 \

--downloaddir=~/python/packages/ \

rh-python36

En este caso la carpeta ~/python/packages contendrá todos los instalables asociados a las dependencias y sub-dependencias de rh-python36. El parámetro installroot indica al gestor de paquetes la ruta donde se encuentran todos los paquetes previamente instalados. Al asociarle una carpeta vacía, el gestor no detecta ninguna dependencia instalada previamente, y las descarga y guarda todas en ~/python/packages (~120 archivos .rpm).

Se recomienda utilizar esta opción con prudencia, ya que las dependencias de más bajo nivel suelen ser programas instalados por defecto en las distribuciones de Linux, y una reinstalación de estos puede suponer una actualización y afectar directamente al funcionamiento del servidor/SO.

### Instalación *offline* de paquetes

### Ejecución y verificación

## Instalación de Apache Airflow

### Apache Airflow

### Esquema de funcionamiento

### Despliegue e Instalación

#### Prerrequisitos

Airflow requiere de unos paquetes mínimos para poder instalarse de manera satisfactoria. Para descargarlos y poder instalarlos offline:

$ mkdir ~/

#### Obtención de paquetes con pip

Apache Airflow está disponible en PyPI, por lo que es fácilmente instalable con pip. Actualmente (Octubre de 2020) la versión estable es la 1.10.12. Debido a que existen diferente versiones de Python3 y Airflow requiere de más paquetes de PyPI para funcionar correctamente, es recomendable instalar las versiones específicas y adecuadas al intérprete que va a ejecutar Airflow. Para la versión 3.6 de python y la 1.10.12 de Airflow, se puede descargar el fichero de contraints de la siguiente forma:

$ wget -O constraint.txt \

https://raw.githubusercontent.com/apache/airflow/constraints-1.10.12/constraints-3.6.txt

$ pip install apache-airflow[<extras>]==1.10.12 --constraint “./constraint.txt”

También se recomienda hacer uso de entornos virtuales.

#### Configuración de PostgreSQL

En principio postgres ya está instalado, sino se puede instalar *online* a través de yum en un sistema Centos 7 o similares. Si se requiere offline la instalación es similar al procedimiento seguido con anterioridad en el punto 1.2.3.2.

$ sudo yum install postgresql-server postgresql-contrib

$ sudo postgresql-setup initdb

$ sudo systemctl start postgresql

$ sudo systemctl enable postgresql # ante reinicios

$ systemctl status postgresql # confirmar que está funcionando

Para configurar PostgreSQL y que Airflow lo utilice como backend o como metadata-db, es recomendable añadir una BBDD y un rol específicos para Airflow. Para ello hace falta tener permisos/rol de superusuario en postgres.

Por defecto, en la instalación se crea un usuario en Linux llamado postgres, que tiene dichos permisos de superusuario sobre el sistema PostgreSQL. Esta configuración puede cambiarse en el momento de iniciar el servicio.

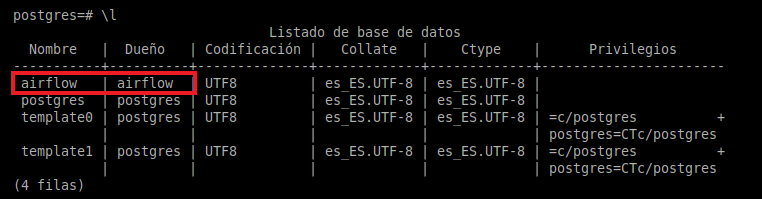
Para crear un rol y una base de datos para Airflow:

$ su – postgres

$ createuser --interactive --pwprompt

$ createdb -O airflow airflow

Se recomienda darle de nombre airflow, contraseña airflow, sin permisos de superusuario y sin permisos para crear BBDD o roles. También se recomienda que la base de datos tenga como nombre airflow. Si los pasos anteriores se han ejecutado bien, debería verse en la lista de BBDD:

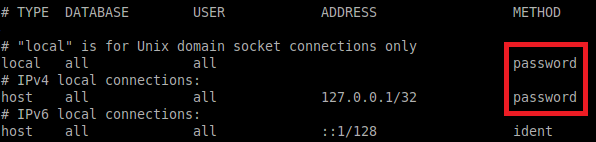


Además, es necesario cambiar el sistema de autenticación en PostgreSQL para las conexiones locales. Por defecto, cuando un usuario se conecta desde la terminal a psql, este interpreta que el usuario es el mismo que el que ha iniciado la sesión. Este comportamiento por defecto puede cambiarse para abstraer a Airflow del usuario que lo ha ejecutado, para ello hay que cambiar el archivo de configuración pg\_hba.conf.

$ sudo find / -iname pg\_hba.conf

$ sudo vi <path/to/pg\_hba.conf>

Y sustituir el método *peer* por *password* para conexiones locales y direcciones localhost:



#### Configuración de Airflow

Con esta configuración, el string de conexión requerido en la configuración de Airflow es (línea 75):

sql\_alchemy\_conn = postgresql+psycopg2://airflow:airflow@localhost/airflow

Y la conexión al result\_backend (línea 470):

result\_backend = db+postgresql://airflow:airflow@localhost/airflow

load\_Examples = False

sql\_alchemy\_conn = postgresql+psycopg2://airflow:airflow@localhost/airflow

result\_backend = db+postgresql://airflow:airflow@localhost/airflow

### Ejecución y Verificación

#### Entornos virtuales

<en este apartado se definen los paquetes a instalar con pip y se explican las dependencias de airflow>

--sistema con conexión a internet--

scl enable rh-python36 bash

python -m venv ~/dev-env

<instalar airflow con los extras y con los constraints>

<instalar paquetes específicos del >

--sistema sin conexión a internet—

scl enable rh-python36 bash

python -m venv ~/dev-env

Copiar la carpeta ~/dev-env/lib/python3.6/site-packages a su análogo en el sistema remoto destino offline

source ~/dev-env/bin/actívate

<lanzar airflow>

airflow initdb

airflow scheduler &

airflow webserver

<cambiar el archivo de configuración creado en ~/airflow>

Sustituir las líneas indicadas arriba para que se conecte a postgres

Relanzar airflow

#### Prueba de funcionamiento

<en este apartado se muestra un código muy sencillo que puede usarse para verificar el funcionamiento>

### Versiones

# Desarrollos realizados

En esta sección se van a detallar los desarrollos realizados para el presente proyecto.

## Ingesta de los datos

### Ingesta de los artículos del BOE

### Ingesta de los artículos de Aragón

## Conversión de los documentos

### De PDF a TXT

### De XML a TXT

### De HTML a TXT

## Extracción de la información

### Extracción mediante reglas

### Extracción mediante el modelo NER

### Modelo jerárquico

Este modelo es el que decide de cuál de los dos extractores se va a recoger la información de cada campo.

## Inserción en la base de datos

# Base de datos

## Estructura

## Tablas

## Campos