



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**IG-Study
Documentación Técnica**



Presentado por Miguel Pérez López de
Echazarreta
en Universidad de Burgos — 3 de julio
de 2019

Tutor: Dr. Jose Manuel Galán Ordax
y Dr. José Ignacio Santos Martín

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	4
Apéndice B Especificación de Requisitos	7
B.1. Introducción	7
B.2. Objetivos generales	7
B.3. Catalogo de requisitos	7
Apéndice C Especificación de diseño	13
C.1. Introducción	13
C.2. Diseño de datos	13
C.3. Diseño procedimental	14
C.4. Diseño arquitectónico	15
Apéndice D Documentación técnica de programación	19
D.1. Introducción	19
D.2. Estructura de directorios	19
D.3. Manual del programador	21
D.4. Compilación, instalación y ejecución del proyecto	23

Apéndice E Documentación de usuario	25
E.1. Introducción	25
E.2. Requisitos de usuarios	25
E.3. Instalación	25
E.4. Manual del usuario	26
Bibliografía	29

Índice de figuras

C.1. Estructura de la Base de Datos	14
C.2. Diseño procedimental del diagrama de Secuencia	15
C.3. Estructura de los Scripts de Python	16
C.4. Estructura del directorio donde almacenar imagenes, csv, fuentes	17
C.5. Estructura de los templates de la web	18
D.1. Estructura de los directorios del proyecto	20
D.2. Modelo que vamos a elegir de PyCharm	21
E.1. Login requerido	26
E.2. Conexión a la base de datos	26
E.3. Información mostrada al iniciar la web con Flask	28

Índice de tablas

A.1. Coste del personal	5
A.2. Coste del Hardware	5
A.3. Coste Total del Proyecto	6
A.4. Tabla librerías	6
A.5. Tabla Licencias	6

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En esta sección vamos a tratar la planificación seguida durante el proyecto. La planificación es una parte fundamental a la hora de embarcarse en un proyecto, y es ahí cuando se toman se fijan temas como:

- Los objetivos
- El tiempo
- La financiación

Por este motivo la planificación se va a dividir en:

- Planificación temporal
- Estudio de viabilidad

A.2. Planificación temporal

Para la planificación temporal, haremos uso de una metodología ágil: SCRUM. Se sigue una estrategia con el desarrollo incremental. La duración de los Sprints era de una o dos semanas, dependiendo de las tutorías futuras posibles y factores externos que pudieran entorpecer el ritmo del proyecto. Se describirán ahora los Sprints realizados:

Sprint 0 (6/3/19 – 12/3/19):

En la primera reunión de planificación del proyecto se hablo de las ideas que iban a llevarse a cabo. Se mandaron las primeras tareas a realizar. Las tareas que realizar fueron:

- Descarga de ciertos Softwares a usar durante 1el proyecto, tales como Balsamic o ZenHub.
- Estudio de la políticas y licencias de Instagram.
- Investigación del funcionamiento de la API.

Sprint 1 (13/3/19 – 20/3/19):

Las tareas a realizar en este Sprint fueron:

- Crear una cuenta de Instagram
- Conectarme a la API

Sprint 2 (21/3/19 – 27/3/19):

Las tareas a realizar en este Sprint fueron:

- Investigación sobre como hacer el reconocimiento de la gente de Burgos
- Hacer una búsqueda de palabras que se usen en las descripciones de los perfiles.

Sprint 3 (28/3/19 – 10/4/19):

Las tareas a realizar en este Sprint fueron:

- Espera a los permisos de la conexión de la base de datos
- Búsqueda de Bases de Datos
- Conexión VPN

Sprint 5 (11/4/19 – 17/4/19):

Las tareas a realizar en este Sprint fueron:

- Preparar servidor en ordenador personal
- Descarga de Screen para Terminal
- Instalación de bibliotecas

Sprint 6 (18/4/19 – 1/5/19):

Las tareas a realizar en este Sprint fueron:

- Instalación del conector de la base de datos
- Recopilación de datos
- Extracción y maquetación de los datos

Sprint 7 (2/5/19 – 15/5/19):

Las tareas a realizar en este Sprint fueron:

- Instalación de la biblioteca de Louvain
- División por comunidades de los datos
- Investigar si hay alguna comunidad clave

Sprint 8 (9/5/19 – 22/5/19):

Las tareas a realizar en este Sprint fueron:

- Estudio de personas relevantes global
- Estudio por comunidades de los usuarios más relevantes

Sprint 9 (23/5/19 – 5/6/19):

Las tareas a realizar en este Sprint fueron:

- Estudio de Hashtags más usados en toda la red
- Estudio de Hashtags más relevantes por comunidades

Sprint 10 (6/6/19 – 19/6/19):

Las tareas a realizar en este Sprint fueron:

- Aprendizaje de Flask
- Realización de la página web donde mostrar los resultados

Sprint 11 (20/6/19 – 3/7/19):

Las tareas a realizar en este Sprint fueron:

- Pulir detalles del proyecto
- Terminar documentación
- Entregar proyecto

Sprint 12 (4/7/19 - 10/8/19):

Las tareas a realizar en este Sprint fueron:

- Presentar el proyecto

A.3. Estudio de viabilidad

Este apartado muestra el estudio realizado sobre la viabilidad económica y legal del producto a entregar.

Viabilidad económica

En la siguiente sección se realiza una aproximación de los costes que tendría que asumir una empresa para llevar adelante este proyecto.

Coste del personal

El transcurso del proyecto ha sido de 3 meses desde que el desarrollador junior se puso a trabajar en él. Suponiendo que el desarrollador tiene un salario neto de 1100 euros:

La retribución a la Seguridad Social es de un 29,9% siguiendo los siguientes puntos a valorar:

Concepto	Coste
Salario mensual neto	1100 euros
Seguridad Social (29,9 %)	596,91 euros
Salario mensual bruto	1.696,91 euros
Total 3 meses	5.090,73 euros

Tabla A.1: Coste del personal

- 23,6 % de contingencias comunes.
- 5,5 % de desempleo de tipo general.
- 0,6 % de formación profesional.
- 0,2 % del fondo de garantía salarial.

Coste del material

En el coste material habría que mirar tanto el coste del Hardware como del Software. En mi caso a nivel de software no ha habido ningún coste puesto que he ido buscando siempre usar software libre. Por este motivo solo hay que valorar el coste material del Hardware. Para la realización del proyecto se ha utilizado el ordenador personal del alumno que tiene un valor de 500 euros al cual se le estima un tiempo para amortizarlo de 6 años. Dado que el tiempo que ha sido de 3 meses:

Concepto	Coste	Coste amortizado
Ordenador	500 euros	20,83 euros

Tabla A.2: Coste del Hardware

Coste Total

La suma de todos los costes es:

Beneficios

La forma de obtener beneficios es vender la aplicación para realizar estudios de mercado. Basándome en un precio teórico de 250 euros por cada estudio de mercado que se requiera, se tardaría en amortizar el producto 25 estudios de mercado. A partir de ese momento habría beneficios.

Concepto	Coste
Hardware	20,83
Personal	5.090,73 euros
Total	5.111,56 euros

Tabla A.3: Coste Total del Proyecto

Viabilidad legal

En este siguiente apartado se va a estudiar la viabilidad legal del proyecto. Hay que analizar todas las licencias de las bibliotecas que se han usado para ver que licencia se podría aplicar a nuestro producto software.

Librería	Licencia
Python and modules integrated	PSFL[Zotero]
Instagram-API-python	MIT License[Zotero]
Networkx	3-clause BSD license [Zotero]

Tabla A.4: Tabla librerías

Como podemos observar en la tablas, las licencias de las bibliotecas utilizadas son bastante permisivas y tienden a favorecer el open source. Este proyecto lo publico con la licencia “Apache 2.0”, que dictamina lo siguiente:

Permissions	Limitations	Conditions
Commercial use	Trademark	License and copyright notice
Modification	Liability	State changes
Distribution	Warranty	
Patent use		
Private use		

Tabla A.5: Tabla Licencias

Apéndice B

Especificación de Requisitos

B.1. Introducción

Este anexo va a describir y formalizar los requisitos que debe de cumplir nuestra herramienta software en cuanto a funcionalidad se respecta.

B.2. Objetivos generales

Los objetivos generales del producto son:

- Obtención de la red
- Estudio de los usuarios
- Implementación de una web para mostrar resultados

B.3. Catalogo de requisitos

Los requisitos se dividen en dos tipos:

Requisitos Funcionales:

- RF1. Crear perfil en Instagram
- RF2. Usar el perfil para loguearse
- RF3. Crear red de usuarios a estudiar

- RF4. Guardar datos en la Base de Datos
- RF5. Realizar división por comunidades
- RF6. Búsqueda de Usuarios relevantes

Requisitos no Funcionales:

- RNF1. Modularidad: El software puede ser continuado por otro alumno, por lo cual debe de ser modular.
- RNF2. Visibilidad: La web que enseña los datos debe de mostrar bien los datos, de manera intuitiva y explicativa

Catálogo de requisitos:

Caso de uso 1:

- Crear perfil en Instagram
- Descripción: El usuario debe de crear un perfil en Instagram
- Requisitos: Ninguno
- Precondiciones: Tener un correo electrónico y conexión a internet
- Secuencia normal Paso Acción
 - 1 Entrar en la página de Instagram
 - 2 Darse de alta con un correo electrónico
- Postcondiciones: Se obtiene una manera de continuar en Instagram haciendo uso de su API
- Excepciones: Tener un usuario previamente
- Importancia: Alta
- Urgencia: Alta
- Comentarios:Ninguno

Caso de uso 2:

- Usar perfil de IG para loguearse
- Descripción: El usuario debe insertar el usuario y contraseña en los scripts para poder hacer uso del proyecto
- Requisitos: RF1
- Precondiciones: Ninguna
- Secuencia normal Paso Acción
 - 1 Entrar en los scripts
 - 2 Escribir el usuario y contraseña
- Postcondiciones: Se puede hacer uso de los distintos procesos implementados
- Excepciones: Ninguna
- Importancia: Alta
- Urgencia: Alta
- Comentarios:Ninguno

Caso de uso 3:

- Creación de la red de usuarios
- Descripción: El usuario debe insertar los valores que van a buscarse en las descripciones para buscar personas que lo cumplan
- Requisitos: RF2
- Precondiciones: Tener agregados previamente a 5 personas que vayan a estar en la red para tener un punto para comenzar la búsqueda.
- Secuencia normal Paso Acción
 - 1 Entrar en los scripts
 - 2 Escribir la lista de términos que van a buscarse
 - 3 Correr el Script durante un tiempo largo

- Postcondiciones: Se seguirá a las personas que cumplen los requisitos con nuestra cuenta de Instagram
- Excepciones: Ninguna
- Importancia: Alta
- Urgencia: Alta
- Comentarios: Puede entrarse en la cuenta de Instagram para ver cuántas personas se han agregado para poder parar el script cuando sean suficientes. Puede durar días este paso.

Caso de uso 4:

- Guardar datos en la base de datos
- Descripción: El usuario debe de correr la aplicación encargada de ver las publicaciones de la gente de la red obteniendo la información.
- Requisitos: RF2, RF3
- Precondiciones: Tener instalado Mysql
- Secuencia normal Paso Acción
 - 1 Ejecutar el script
- Postcondiciones: Ninguna
- Excepciones: Tener un usuario previamente
- Importancia: Alta
- Urgencia: Alta
- Comentarios: El tiempo requerido va a ser proporcional al número de publicaciones de los usuarios y número de usuarios en sí.

Caso de uso 5:

- Dividir por comunidades
- Descripción: El usuario debe de correr la aplicación encargada de marcar las comunidades encontradas en la red.

- Requisitos: RF4
- Precondiciones: Tener los datos en CSV
- Secuencia normal Paso Acción
 - 1 Ejecutar el script teniendo los datos en la misma ubicación del programa
- Postcondiciones: Se obtiene una manera de continuar en Instagram haciendo uso de su API
- Excepciones: Tener un usuario previamente
- Importancia: Alta
- Urgencia: Alta
- Comentarios:Ninguno

Caso de uso 6:

- Búsqueda de Usuarios relevantes
- Descripción: Se obtendrá una lista con los usuarios más relevantes de la red y comunidades
- Requisitos: RF2, RF3
- Precondiciones: Usar Google Colab
- Secuencia normal Paso Acción
 - 1 Subir el notebook a google colab
 - 2 Ejecutar el Notebook y seguir los pasos que nos va indicando
- Postcondiciones: Ninguna
- Excepciones: Ninguna
- Importancia: Alta
- Urgencia: Alta
- Comentarios:Ninguno

Apéndice C

Especificación de diseño

C.1. Introducción

Este anexo va a describir la razón de ser del procedimiento que he seguido a la hora de organizar el proyecto. También se describirán las razones por las cuales se han tomado así las decisiones.

C.2. Diseño de datos

Ahora se explicarán como están implementados los datos en la base de datos en MySQL.

Tablas de la base de datos

Para guardar la información requerida de la API de Instagram usamos la base de datos con las relaciones de la siguiente manera:

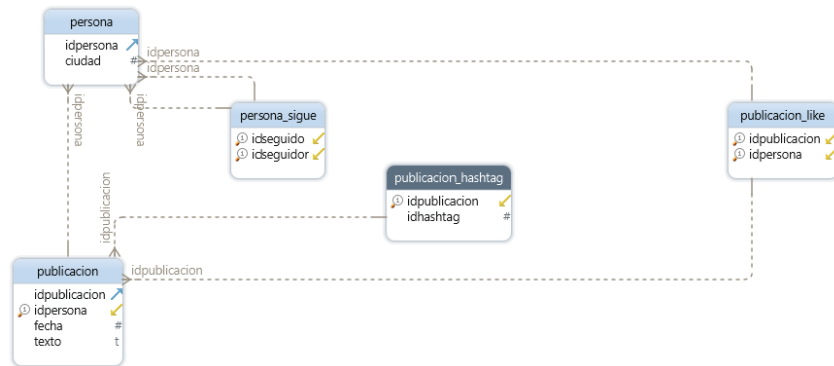


Figura C.1: Estructura de la Base de Datos

- **Persona**: Almacena las personas que van a formar la red.
- **Persona_sigue**: Almacena la relación de seguimiento entre usuarios.
- **Publicación**: Almacena la publicación con la persona que la ha realizado. No se almacena la foto, solo la fecha y el texto a investigar.
- **Publicación_Hashtag**: Almacena la publicación con los diferentes hashtags que contenga.
- **Publicación_like**: Almacena la publicación con la gente de la red que haya interactuando con ella dando un “me gusta”.

C.3. Diseño procedimental

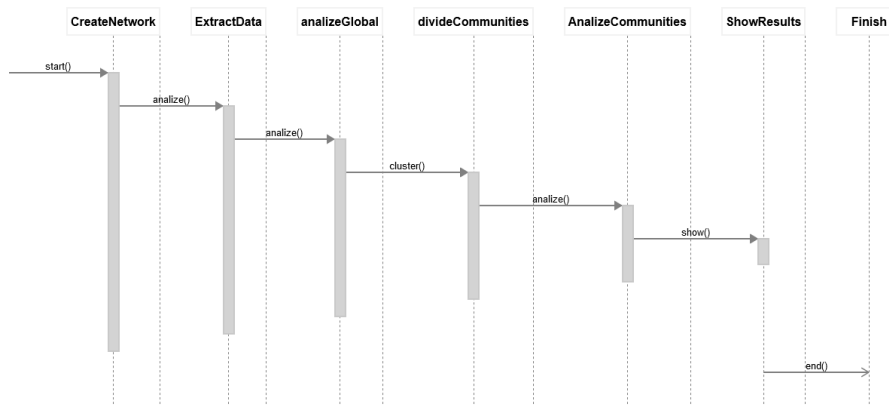


Figura C.2: Diseño procedimental del diagrama de Secuencia

Como se puede observar, es un modelo secuencial que sigue siempre la misma dirección. Se ha intentado mantener una sencillez que ayude tanto al usuario como al programador que tenga que hacer uso de ella. A continuación, comienzo la explicación del proceso que se observa del diagrama:

- 1 Se introducen los datos, tanto del usuario, como para la búsqueda y creación de la red.
- 2 Se procede a la creación de la red.
- 3 Se extraen los datos necesarios de la API y se almacenan en la base de datos de MySQL.
- 4 Se analizan los datos globales de la base de datos.
- 5 Se calculan las comunidades y se guardan los datos como csv.
- 6 Se analizan los datos para obtener los resultados de las distintas comunidades.
- 7 Se muestran los resultados en la web.

C.4. Diseño arquitectónico

El código del programa se ha llevado a cabo dividiendo el programa en 3 partes:

- Scripts principales: En los scripts principales están las principales funciones de creación de red, extracción de datos y análisis de los datos.

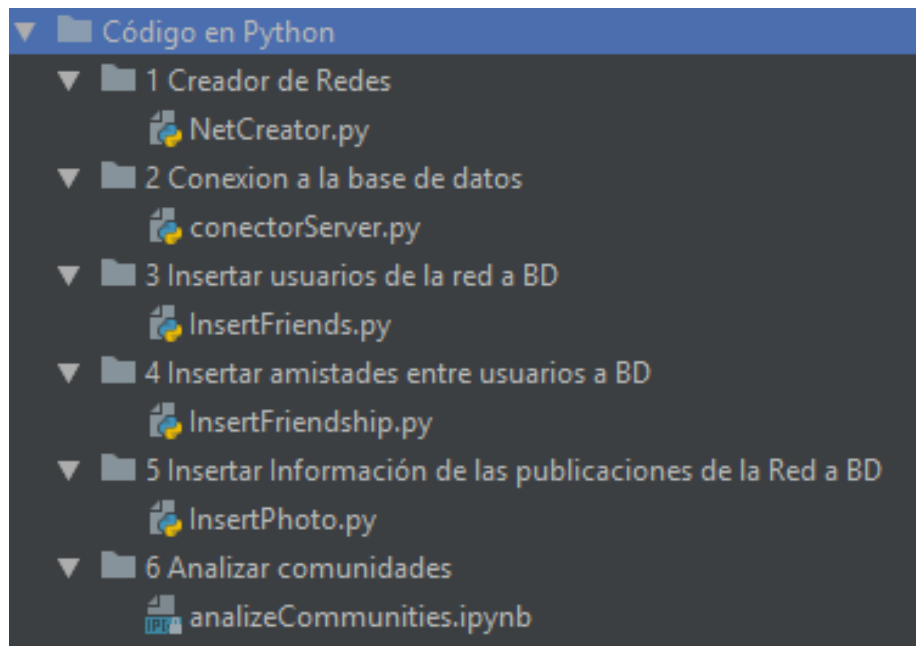


Figura C.3: Estructura de los Scripts de Python

- Static: en este apartado están los estáticos que se van a usar en la web. Imágenes, datos en csv, fuentes, etc.

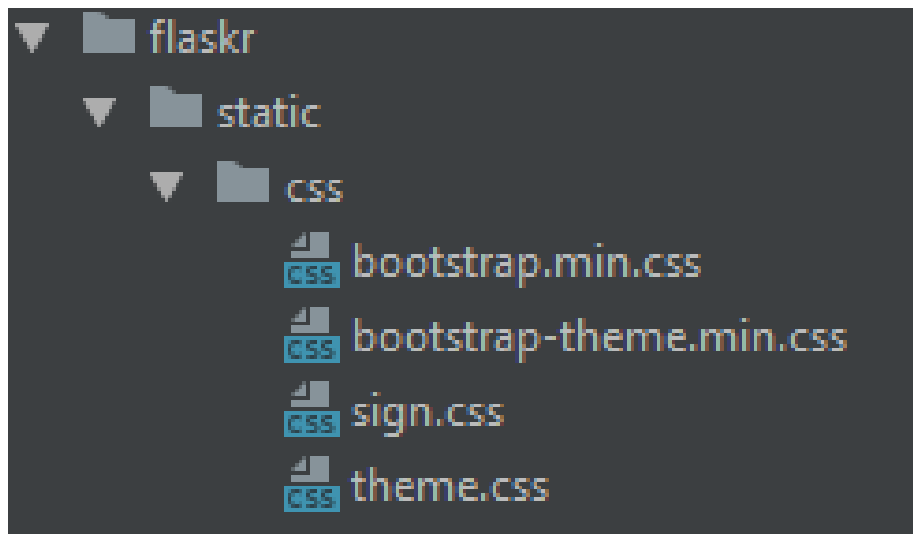


Figura C.4: Estructura del directorio donde almacenar imagenes, csv, fuentes

- Templates: en este apartado se encuentra los archivos en HTML para poder mostrar la web.

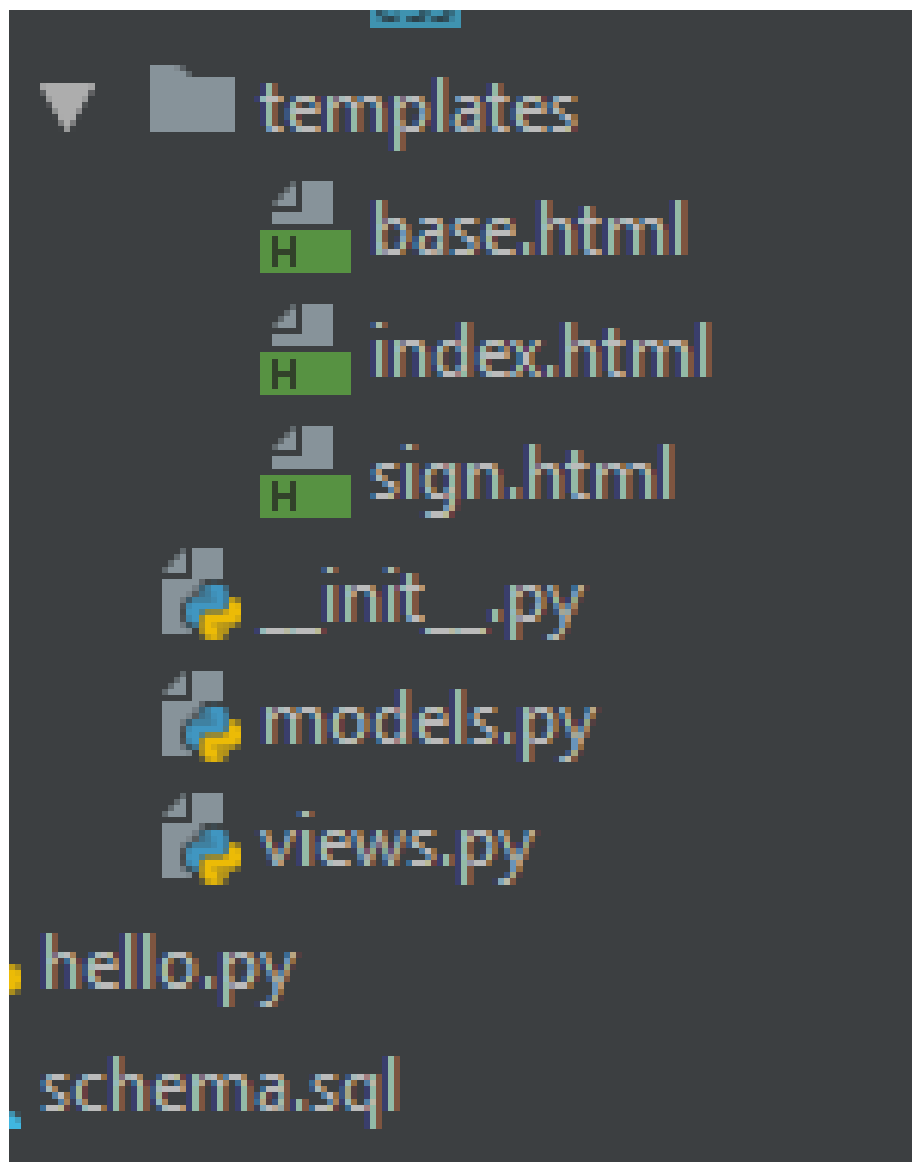


Figura C.5: Estructura de los templates de la web

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este anexo va a describir la documentación referente a la programación. Se incluirá también la instalación necesaria de todos los componentes para poder seguir desarrollando el proyecto.

D.2. Estructura de directorios

A continuación, se van a mostrar los directorios de este proyecto que están en el directorio raíz.

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

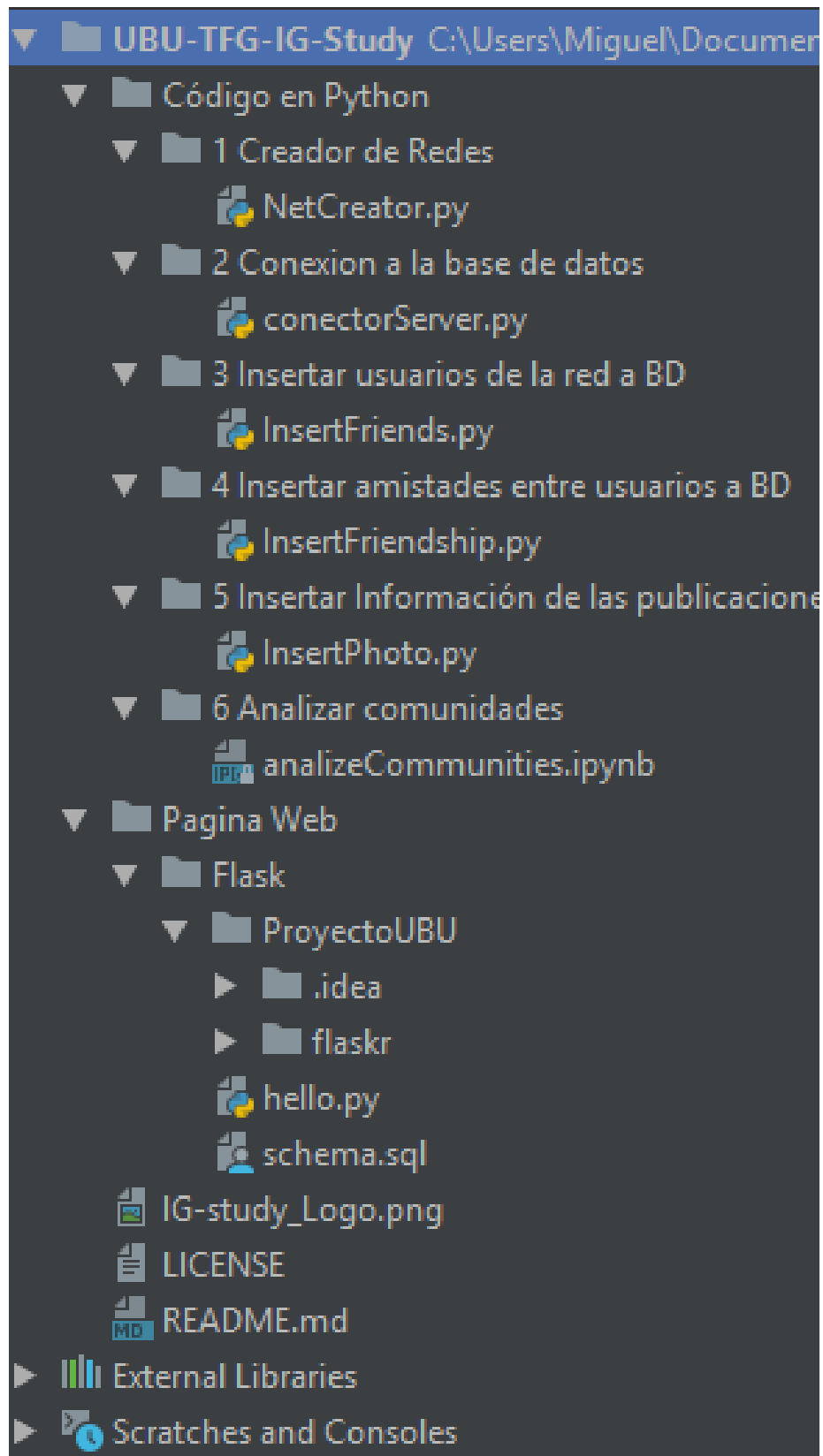


Figura D.1: Estructura de los directorios del proyecto

D.3. Manual del programador

En este apartado se facilitarán los programas y librerías que son necesarios para seguir con el desarrollo del proyecto. Para trabajar con este proyecto son necesarios los siguientes programas y librerías:

- Pycharm
- Python
- Pip
- Instagram-API-python
- Networkx
- Python-Louvain
- Flask

Instalación de PyCharm

Pycharm ha sido el IDE que se eligió para programar los scripts de nuestro proyecto. Para la instalación hay que descargarse el software desde su página web [3] eligiendo la versión que queramos instalar. Gracias a la cuenta de la Universidad de Burgos, con la cuenta universitaria podemos descargarnos la versión profesional de manera gratuita, aunque la versión gratuita también puede usarse.

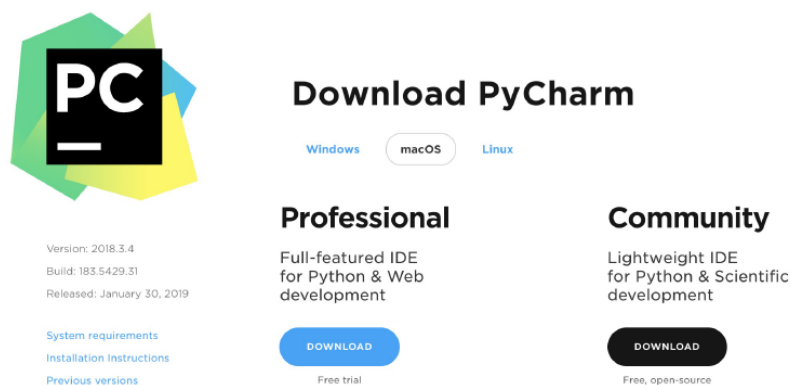


Figura D.2: Modelo que vamos a elegir de PyCharm

~~22~~ APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

Cuando esté descargado, solo es seguir el instalador que no requiere de ningún conocimiento técnico, solo seguir los pasos.

Python

Python es el lenguaje principal sobre el que se programa todo el proyecto. Esto requiere que sea imprescindible que se tenga instalado para poder seguir con el proyecto. Se ha utilizado la versión 2.7.15. En el momento de la entrega del proyecto hay versiones más avanzadas, sin embargo, esta cumplía con todos los requisitos necesarios para el uso de las librerías. Para instalarlo hay que ir a la pagina oficial de Python[5] elegir el sistema operativo con el que se va a trabajar y la versión que queramos utilizar. Una vez descarguemos el asistente de instalación, solo habría que ejecutarlo y seguir los pasos que se indiquen.

Pip

Pip es una herramienta que proporciona Python que ayuda sobremanera en la instalación, actualización y desinstalación de bibliotecas desde línea de comandos. Las librerías que se van a usar van a ser descargadas usando pip, por lo tanto, en caso de que no se haya instalado de manera automática con Python, es muy recomendable descargarlo.

Instagram-API-Python

[4] Instagram-API-Python es la biblioteca que nos acerca la api de Instagram a Python y va a ser usado para la creación de la red y extracción de la información. Es requerida para esos 2 primeros pasos y se instala escribiendo en la línea de comandos:

- `pip install InstagramApi`

para su uso en los script de Python hay que importar escribiendo:

- `from InstagramAPI import InstagramAPI`

Networkx

[2] Instagram-API-Python es la biblioteca que nos acerca la api de Instagram a Python y va a ser usado para la creación de la red y extracción

D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

de la información. Es requerida para esos 2 primeros pasos y se instala escribiendo en la línea de comandos:

- `pip install networkx`

Python-Louvain

[1] Es una biblioteca que funciona como adjunta de Networkx y que nos permite hacer la división clustering. Podría hacerse división usando otro método, pero como este es el que hemos elegido, se requiere de esta biblioteca paralela. Para su instalación basta con escribir en la línea de comandos:

- `pip install python-louvain`

Flask

Flask es el framework que se va a utilizar para la web que se encarga de mostrar los resultados finales. En caso de que se quiera hacer algún cambio o correrlo hay que tenerlo instalado. Su instalación es sencilla. Basta con escribir en la línea de comandos:

- `pip install Flask`

D.4. Compilación, instalación y ejecución del proyecto

Para la ejecución del proyecto es necesario descargarse el proyecto del repositorio de Github, donde está alojado. Para acceder al repositorio solo hay que acceder al siguiente enlace (ZOTERO). A continuación, pulsaremos el botón verde de la parte de la derecha que dice “Clone or download” Salvo que tengamos la opción de clonar el git, particularmente me parece más simple descargar el zip.

Instalación

Este proyecto no requiere instalación más allá de tener todas las librerías necesarias instaladas previamente para su uso.

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

Ejecución

Para la ejecución he dividido las distintas fases en distintos directorios. Hay que ejecutar cada archivo de Python por orden, pero previamente hay que introducir el usuario de Instagram que se vaya a usar, el termino de búsqueda de las descripciones y la base de datos. Para la extracción de los datos del mysql a CSV hay que escribir como administrador el siguiente comando en mysql:

- `select IDseguidor, IDseguido from persona_sigue into outfile '/var/lib/mysql-files/prueba.csv' fields enclosed by '/' terminated by ';' escaped by ' LINES terminated by '/r/n';`

Hay que tener en cuenta que los archivos que se necesiten a la hora de correr los scripts estén en el mismo directorio que se vaya a ejecutar.

Apéndice E

Documentación de usuario

E.1. Introducción

Este anexo va a describir la documentación referente a la ejecución de la aplicación desde el punto de vista del usuario.

E.2. Requisitos de usuarios

Los requisitos mínimos para poder usar la aplicación son:

- Tener instalado Python 2.7
- Tener instaladas las bibliotecas Instagram-API-python, Networkx y Python-Louvain. En el anexo anterior se explica cómo se pueden conseguir dichas librerías.
- Tener conexión a internet.
- Tener una cuenta de Instagram con un mínimo de 5 personas seguidas que vayan a estar dentro de nuestra red.

E.3. Instalación

Para ejecutar la aplicación no es necesario cumplimentar una instalación tradicional, sino que con una descarga del repositorio de GitHub y ejecución de los distintos Scripts valdría.

E.4. Manual del usuario

Hay que cumplimentar unos pequeños pasos antes de ejecutar cada Script.

NetCreator

Requiere que se introduzca el usuario y contraseña de la cuenta de Instagram, así como las palabras claves que se quieren buscar para crear la red. Este primer paso lleva de varios días sí

```
if name == "main":
    nombre = raw_input("Introduce tu nombre: ")
    contrasena = raw_input("Introduce tu contrasena: ")
    api = InstagramAPI(nombre, contrasena)
    api.login()
    user_id = api.username_id
```

Figura E.1: Login requerido

InsertFriends

Requieres tanto el usuario y la contraseña de instagram como las que se refieren a MySQL, que es donde guardaremos la información que extraigamos.

```
nombremysql = raw_input("Introduce tu nombre de mysql: ")
contrasenamysql = raw_input("Introduce tu contrasena de mysql: ")
databasemysql = raw_input("Introduce tu database de mysql: ")
try:
    cnx = mysql.connector.connect(user=nombremysql, password=contrasenamysql,
                                  database=databasemysql)
except mysql.connector.Error as err:
    if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
    elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
    else:
        print(err)
```

Figura E.2: Conexión a la base de datos

InsertFriendship

Al igual que InsetFriends, necesitamos introducir el usuario y contraseña del Instagram y de MySQL. Después de extraer la información de la API hay que configurar los datos como un CSV para que pueda leerlos bien Networkx. Para eso, se extraen desde la línea de comandos de la siguiente forma:

- `'/var/lib/mysql-files/prueba.csv'` fields enclosed by `'` terminated by `'`; escaped by `'` LINES terminated by `'/r/n'`;

Partition

Partition es un script que no requiere de ningún cambio y que puede ejecutar sin problemas una vez hayamos cumplimentado todos los scripts anteriores.

AnalizeCommunities

Este es un notebook de Python que en vez de hacerlo un script como los anteriores, hay que subirlo a colab.research.google.com, puesto que necesita de mayores recursos que lo que normalmente se tiene en un ordenador personal. Nos requiere en el primer paso que le introduzcamos los csv de las particiones y al final nos devuelve los datos para descargar.

Flask

Para ejecutar Flask hay que escribir en la línea de comandos (desde Windows) hay que ir a la carpeta de ProyectoUBU dentro de flask y con botón derecho+mayusc abrir una powershell y escribir los 2 siguientes comandos:

- `set FLASK_APP=flaskr`
- `flask run`

Después abrir un navegador e ir a la página por defecto que es <http://127.0.0.1:5000/>

```
* Serving Flask app "flaskr"
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
C:\python27\lib\site-packages\flask_sqlalchemy\__init__.py:835: FSADeprecationWarning: SQLAlchemy_TRACK_MODIFICATIONS adds
significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
'SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and '
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figura E.3: Información mostrada al iniciar la web con Flask

Se adjunta un video junto con los anexos que puede verse por si existieran dudas sobre la ejecución del programa.

Bibliografía

- [1] Blondel. Manual python-louvain, 2019. [Internet; descargado 01-mayo-2019].
- [2] NetworkX Developers. Pagina oficial de networkx, 2018. [Internet; descargado 03-mayo-2019].
- [3] jetbrains. Web oficial de pycharm, 2019. [Internet; descargado 07-junio-2019].
- [4] LevPasha. Repositorio de la api para python, 2019. [Internet; descargado 16-junio-2019].
- [5] Python. Pagina oficial de python, 2019. [Internet; descargado 16-junio-2019].