UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Aplicación Web para la recopilación, tratamiento y visualización de datos públicos

Presentado por Iván Arjona Alonso
en Universidad de Burgos — 16 de mayo de 2018
Tutores: Dr. José Francisco Díez Pastor
y Dr. Jesús Manuel Maudes Raedo

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. José Francisco Díez Pastor y D. Jesús Manuel Maudes Raedo, profesores del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Iván Arjona Alonso, con DNI 71352655P, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "Aplicación Web para la recopilación, tratamiento y visualización de datos públicos".

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 16 de mayo de 2018

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
José Francisco Díez Pastor	D. Jesús Manuel Maudes Raedo

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

La *minería de datos* es una metodología para descubrir relaciones ocultas en grandes cantidades de datos utilizando técnicas de inteligencia artificial.

Actualmente se está extendiendo bastante el uso de técnicas de minería de datos para realizar análisis sociológicos. Un tema que está siendo investigado actualmente es comprender las causas por las que se producen crímenes. Por ello, se han realizado estudios para investigar las tasas de criminalidad utilizando técnicas de minería de datos [9] [11] a partir de datos demográficos y sociológicos con el objetivo de determinar los lugares donde asignar más recursos para reducir estos índices de criminalidad.

El objetivo de este trabajo es llevar estos estudios al ámbito español. Integrando varias fuentes de datos públicas utilizando técnicas de web scraping (3.3). Para después visualizar estos datos de manera sencilla, permitir crear mapas temáticos (3.4) con esta información y aplicar técnicas de minería de datos para buscar relaciones entre los datos.

Fuentes de datos

A continuación se van a listar las fuentes de datos de organismos públicos que se han integrado.

- Servicio Público de Empleo Estatal (SEPE)
  - Paro registrado por municipio[1].
  - Contratos registrados por municipio[2].

[1] https://datos.gob.es/catalogo/e00142804-paro-registrado-por-municipios
[2] http://datos.gob.es/es/catalogo/e00142804-contratos-por-municipios

    - Demandantes de empleo por municipio[3].
- Estadísticas de la renta por municipio (Agencia tributaria)[4].
- Instituto Nacional de Estadística (INE)
    - Estadísticas de población por sexo y edad[5].
    - Relación de municipios y códigos por provincias[6].

Material entregado

Material adjunto a la memoria:

- Aplicación web en Flask.
- Scripts para la integración de fuentes de datos.
- Scripts para la ejecución.
- Aplicación de escritorio para juntar csv.
- Anexos.

Recursos disponibles en internet:

- Página web del proyecto[7] .
- Repositorio del proyecto[8] .

[3] http://datos.gob.es/es/catalogo/e00142804-demandantes-de-empleo-por-municipios

[4] https://www.agenciatributaria.es/AEAT.internet/datosabiertos/catalogo/hacienda/Estadistica_de_los_declarantes_del_IRPF_por_municipios.shtml

[5] http://www.ine.es/jaxi/Tabla.htm?path=/t20/e245/p05/a2011/l0/&file=00000001.px&L=0

[6] http://www.ine.es/daco/daco42/codmun/codmunmapa.htm

[7] Página web del proyecto: https://tfg-datos-publicos.nanoapp.io/

[8] Repositorio del proyecto: https://github.com/IvanArjona/TFG-Datos-publicos

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

2.1. Objetivos funcionales

- Integrar varias fuentes de datos públicas en una única base de datos. Estos datos son datos de carácter sociológicos, económicos y demográfico a nivel municipal en España.
- Permitir añadir nuevos conjunto de datos de forma sencilla.
- Desarrollar un algoritmo para la carga de varias fuentes de datos en una base de datos de manera automatizada.
- Desarrollo de una aplicación web que permita la consulta de los datos de manera sencilla y visual.
- Facilitar la interpretación de los datos utilizando un mapas coropléticos interactivo.
- Desplegar la aplicación web en un servidor de forma que sea fácil de actualizar cada vez que se realice un cambio. Además de funcionar en un entorno local.

2.2. Objetivos técnicos

- Seguimiento de principios de desarrollo ágiles durante el desarrollo del proyecto.
- Utilizar *Flask* como framework para desarrollar la aplicación web utilizando la arquitectura *Modelo Vista Controlador*.
- Utilizar *git* como sistema de control de versiones, junto con *Github* y *Zenhub* para la organización del proyecto.
- Utilizar herramientas para mejorar la calidad del código como *codacy*.
- Manejar bases de datos no relacionales *NoSQL*.
- Elegir erramientas de *software libre* siempre que sea posible.
- Proporcionar una instalación sencilla para el desarrollador y una interfaz visual y fácil de utilizar para el usuario.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

3.1. Datos públicos

Se entiende como datos públicos o datos abiertos aquellos que deben estar disponibles de manera libre, para acceder, utilizar, modificar y publicar sin restricciones de copyright [17].

Los gobiernos tienen la capacidad de obtener grandes cantidad de información sobre la población a través de varios organismos (como podria ser el *Instituto nacional de estadística*), cuando estos gobiernos liberan los datos para que cualquiera pueda utilizarlos libremente se conocen como 'Open Government Data'.

En este proyecto nos centraremos en los datos públicos españoles, aprovechando la 'Iniciativa Aporta' que promueve la apertura de información en el sector público en España. Esta iniciativa tiene el objetivo de favorecer el desarrollo de la reutilización de la información del sector público y ayudar a las administraciones para que publiquen sus datos de acuerdo al marco legislativo vigente [2].

Aunque nos vamos a centrar en datos proporcionados por organismos estatales, también podrían emplearse datos públicos proporcionados por empresas privadas (Por ejemplo *Google Trends* o datos meteorológicos), se dejará pendiente para trabajos futuros.

3.2. Bases de datos no relacionales

Las bases de datos no relacionales [21], también llamadas NoSQL ('Not Only SQL') son bases de datos optimiazadas para ser utilizadas con modelos de datos sin esquema y potencialmente escalables.

A diferencia de las bases de datos relacionales, aquí no hay tables, esquemas ni relaciones, sino que los datos pueden almacenarse con cualquier esquema sin tener que seguir todos la misma estructura.

En este tipo de bases de datos tenemos colecciones en lugar de tablas y dentro de estas colecciones tenemos documentos. Estos documentos pueden contener cualquier información y se almacenan siguiendo un esquema json.

Una de las razones importantes por las que usar bases de datos NoSQL en lugar de relacionales es la alta velocidad de consulta y la gran capacidad de escalabilidad.

3.3. Web scraping

Web Scraping [22] es una técnica para extraer información de sitios web directamente de su código fuente sin utilizar APIs[9] proporcionadas por el propio sitio.

Lo que hacemos con un *web scraper* es buscar información dentro de un documento web siguiendo ciertos patrones en la estructura de su código fuente y extraer esta información a nuestro entorno local.

Las técnicas de web scraping se centran en transformar datos sin estructura de una página web en datos estructurados para poder ser almacenados y analizados posteriormente. Por ejemplo, en una base de datos, hojas de cálculo de dataframes de pandas.

[9] Interfaz de programación de aplicaciones

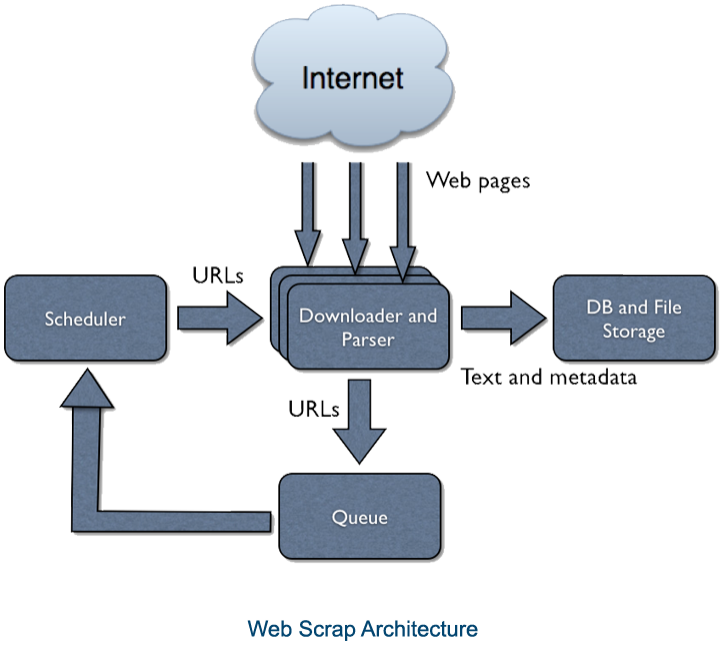

Figura 3.1: Esquema del funcionamiento de web scraping [19]

3.4. Mapas coropléticos

Un mapa coroplético [20] es un mapa topológico dividido en regiones en el que cada una de estas regiones se pinta de un color de acuerdo a una medida estadística.

Para ilustrarlo mejor pondremos el ejemplo del siguiente mapa en la figura 3.2. En él se compara el paro de todas las provincias de España pintando con colores más cálidos las provincias con mayor porcentaje de paro y con colores más fríos las provincias con menor porcentaje de paro.

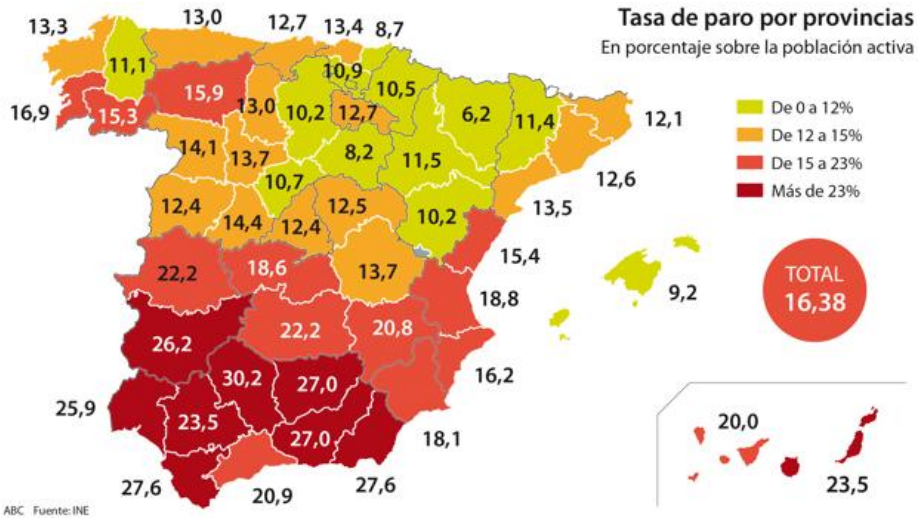

Figura 3.2: Mapa coroplético del paro en las provincias españolas [18]

Técnicas y herramientas

4.1. Técnicas

Metodología ágil

Este proyecto se ha realizado siguiendo los principios del manifiesto ágil [10]. En concreto se ha seguido la metodología *scrum* [1] con sprints normalmente de una semana y reuniones semanales de planificación y revisión.

Al utilizar esta metodología se persigue utilizar una estrategia de desarrollo incremental, en la que al final de cada sprint se tiene como resultado un incremento del software entregable. También se consigue que los requisitos cambien a lo largo del proyecto, en lugar de fijarse al principio, consiguiendo un software de mejor calidad.

Para seguir esta metodología se han utilizado las issues de Github[10] y los tableros Kanban de ZenHub[11] para organizar la pila de tareas a realizar en cada sprint.

DevOps

DevOps [3] es una metodología para la creación de software en la que se integra el desarrollo de software y la administración de sistemas.

Se ha elegido esta metodología para conseguir desplegar rápidamente todos los cambios en el servidor web. Apollandonos en herramientas como Nanobox (4.2) para facilitar este proceso.

[10] issues de Github: https://github.com/IvanArjona/TFG-Datos-publicos/issues
[11] ZenHub: https://www.zenhub.com/

4.2. Herramientas

MongoDB

- Herramientas consideradas: Riak, Cassandra, MongoDB, LevelDB.
- Herramienta elegida: MongoDB.

MongoDB[12] [14] es un sistema de bases de datos NoSQL. En esta herramienta los datos se guardan en forma de documentos con un esquema similar a json. Con este sistema se consigue una consulta de datos más rápida. Además se ha usado PyMongo [13] como herramienta para integrar MongoDB en python.

Tanto Riak como Cassandra son también bases de datos NoSQL, se descartaron porque no ofrecen soportes para equipo con sistema operativo windows.

LevelDB es una base de datos NoSQL de pares clave-valor, esta herramienta se descartó porque no se cree conveniente utilizar pares clave-valor para un proyecto como este y no hay tantos ejemplos en la documentación como en las otras herramientas.

DigitalOcean

- Herramientas consideradas: Heroku, PythonAnywhere, DigitalOcean, Amazon Web Services.
- Herramienta elegida: DigitalOcean.

Digital Ocean[13] es un proveedor de servidores privados, por ello podemos hacer lo que queramos con el servidor sin limitaciones más allá de la capacidad de procesamiento y memoria ram. Se ha elegido este servicio porque nos da total libertad y se puede probar gratuitamente con GitHub Education [16].

Una alternativa que se consideró y de hecho, se probó es Heroku, en este caso se instala el entorno necesario de forma automática. El problema es que la base de datos en la capa gratuita sólo puede pesar 500MB como máximo y no es suficiente para este proyecto.

PythonAnywhere es un hosting para aplicaciones web en python, el problema con este proveedor es que no ofrece bases de datos locales, por lo

[12]MongoDB: https://www.mongodb.com/
[13]Digital Ocean: https://www.digitalocean.com/

que habría que utilizar una remota. La única gratuita que se ha encontrado es mLab, la misma que usa Heroku, por lo que volvemos al mismo problema del límite de tamaño.

Por último, se consideró utilizar una instancia de Amazon AWS EC2. Es muy similar a DigitalOcean, se eligió el primero porque es más sencillo de utilizar.

Nanobox

- Herramientas consideradas: Heroku, Nanobox.
- Herramienta elegida: Nanobox.

Nanobox[14] es una herramienta que nos permite desplegar nuestra aplicación sin centrarnos en la infraestructura del servidor [15].

Para ello enlazamos nuestra cuenta de un proveedor en la nueve (en este caso DigitalOcean) y nanobox se encargará de instalar el sistema operativo, configurarlo, instalar nuestra aplicación y sus dependencias y ejecutarla.

Heroku es un servicio muy similar, con la diferencia de que no podemos utilizar servidores en la nuve externos, se descartó por lo ya explicado en el punto anterior.

Flask

- Herramientas consideradas: Flask, Django.
- Herramienta elegida: Flask.

Como uno de los objetivos del proyecto es el de crear una página web, se han considerado varios frameworks web para python. Entre ellos se ha seleccionado Flask.

Flask[15] es un framework fáciles de utilizar y muy flexible. No nos fuerza a utilizar una metodología específica y podemos organizar la aplicación con la estructura que queramos (a diferencia de Django) [6].

Además se incluyen herramientas para desplegar el servidor de desarrollo, para realizar pruebas de la aplicación y para hacer *APIs REST*.

[14]Nanobox: https://nanobox.io/
[15]Flask: http://flask.pocoo.org/

Folium

- Herramientas consideradas: Plotly maps, Folium, Leaflet.
- Herramienta elegida: Folium.

Un aspecto importante de este proyecto es la representación de datos en el mapa.

Para ello se han considerado varias herramientas, de las cuales se ha elegido Folium[16] [12]. Este paquete nos permite visualizar datos manipulados con Python y visualizarlos como mapas utilizando para ello LeafletJS [7].

Se ha elegido esta herramienta porque nos permite representar mapas coropléticos (3.4) y mapas con clusters de datos [5] utilizando estructuras de datos geográficas personalizadas (geojson) [8].

Dynatable

- Herramientas consideradas: Dynatable, Datatables.
- Herramienta elegida: Dynatable.

Dynatable[17] es un framework de javascript para visualizar tablas de una manera más clara y ordenadas [4].

Este framework nos permite paginar los resultados, ordenar por alguna columna y buscar por cualquier campo dentro de la tabla. Todo esto sin tener que recargar la página.

[16]Folium: http://python-visualization.github.io/folium/
[17]Dynatable: https://www.dynatable.com/

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

6.1. Mapa del paro

Mapa del paro[18] es una página web que nos permite visualizar el paro por municipios, provincias y comunidades en un mapa coroplético. Ver figura 6.3.

Los datos del paro contenidos en esta página web también están en este proyecto, con la ventaja de que además se pueden combinar con otros datos públicos y se puede descargar la información.

[18]Mapa del paro: http://mapadelparo.com/

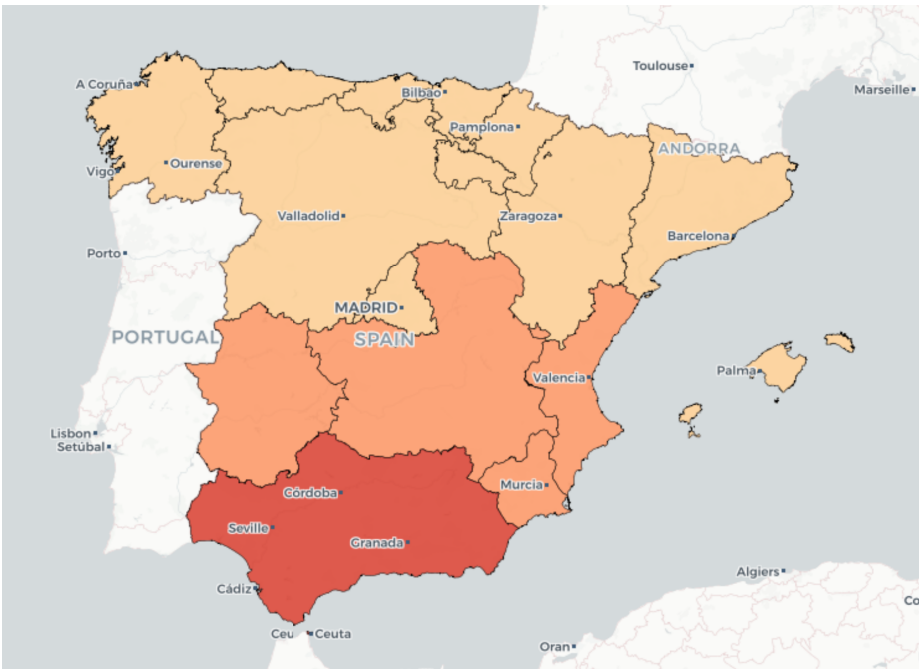

Figura 6.3: Paro por comunidades autónomas

6.2. Ciudatos

Ciudatos[19] nos permite visualizar varios datos públicos de Colombia mediante gráficas de barras y posicionarlos en el mapa. Ver figura 6.4.

Esta herramienta no nos permite visualizar los datos con mapas coropléticos, simplemente sitúa los gráficos de barras de cada región en su correspondiente sitio.

Como ventaja nuestra aplicación tiene datos de España y permite visualizar varios datos diferentes entre ellos al mismo tiempo.

[19]Ciudatos: http://www.ciudatos.com/visualizacion

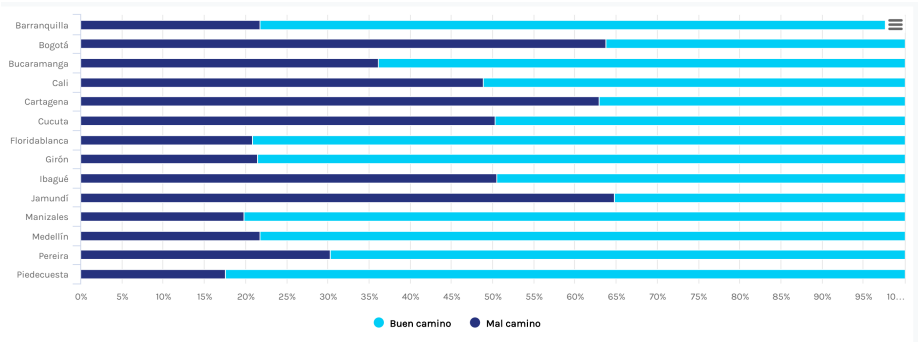

Figura 6.4: Bienestar por ciudades (Colombia)

6.3. Eurostat

Otra representación de datos públicos en mapas es Eurostat[20] .

En esta representación se muestra en el mapa los porcentajes de empleo en cada país de Europa. Ver figura 6.5.

Estos datos se escapan del alcance al que se quería llegar en este trabajo, pero podría considerarse para integrarlos en algún proyecto futuro.

[20]Eurostat: http://ec.europa.eu/eurostat/en/web/lfs/statistics-illustrated

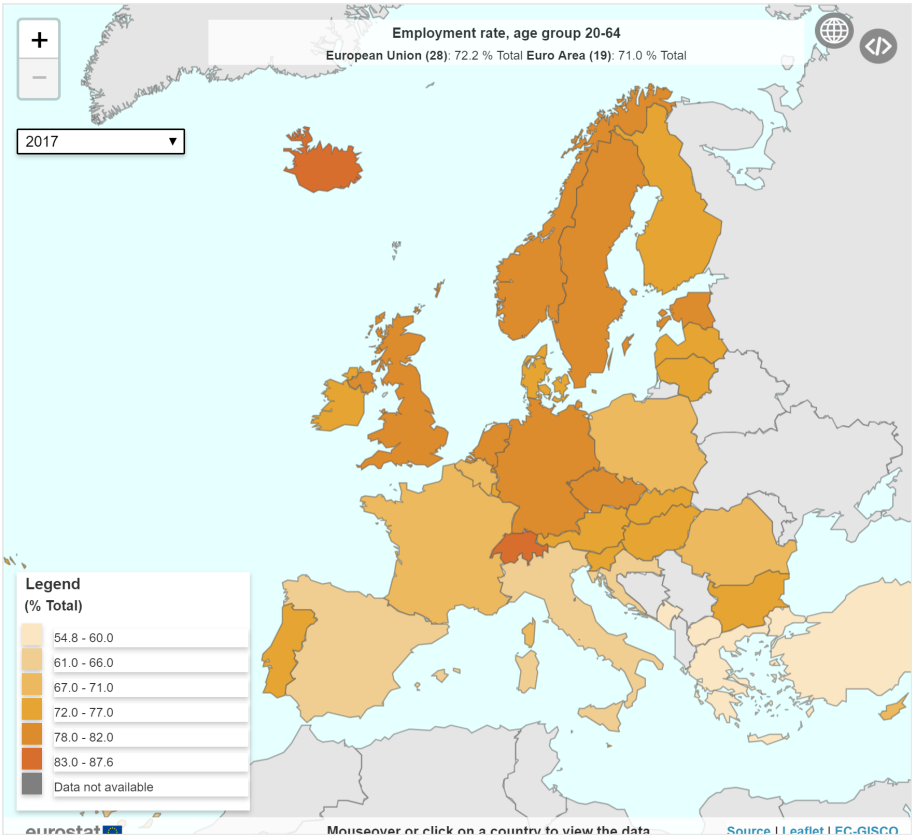

Figura 6.5: Porcentaje de empleo por país (Europa)

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] Juan Palacio Alexander Menzinsky, Gertrudis López. *Scum Manager*. Scrum Manager, 2016.

[2] Iniciativa Aporta. Acerca de la iniciativa aporta. http://datos.gob.es/es/acerca-de-la-iniciativa-aporta. [Internet; descargado 16-mayo-2018].

[3] José Ruiz Cristina. Qué es devops (y sobre todo qué no es devops). https://www.paradigmadigital.com/techbiz/que-es-devops-y-sobre-todo-que-no-es-devops/. [Internet; descargado 14-mayo-2018].

[4] Dynatable. Dynatable.js - jquery plugin for html5+json interactive tables and more. https://www.dynatable.com/. [Internet; descargado 06-mayo-2018].

[5] Folium. Markercluster – folium. http://nbviewer.jupyter.org/github/python-visualization/folium/blob/master/examples/MarkerCluster.ipynb. [Internet; descargado 06-mayo-2018].

[6] Miguel Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2014.

[7] LeafletJS. Documentation - leaflet. https://leafletjs.com/reference-1.3.0.html. [Internet; descargado 06-mayo-2018].

[8] LeafletJS. Using geojson with leaflet. https://leafletjs.com/examples/geojson/. [Internet; descargado 06-mayo-2018].

[9] Francisco A. Rodrigues Luiz G.A. Alves, Haroldo V. Ribeiro. Crime prediction through urban metrics and statistical learning. *Physica A: Statistical Mechanics and its Applications*, 505:435–443.

[10] Jim Highsmith Martin Fowler. The agile manifesto. *Software Development*, 9:28–35.

[11] Mohammad Reza Ebrahimi Mohammad Reza Keyvanpour, Mostafa Javideh. Detecting and investigating crime by means of data mining: a general crime matching framework. *Procedia Computer Science*, 3:872–880.

[12] MongoDB. Folium documentation. http://python-visualization.github.io/folium/. [Internet; descargado 06-mayo-2018].

[13] MongoDB. Pymongo 3.6.1 documentation. https://api.mongodb.com/python/3.6.1/. [Internet; descargado 22-abril-2018].

[14] MongoDB. What is mongodb? https://www.mongodb.com/what-is-mongodb. [Internet; descargado 22-abril-2018].

[15] Nanobox. Nanobox documentation. https://docs.nanobox.io/. [Internet; descargado 22-abril-2018].

[16] Digital Ocean. Product documentation. https://www.digitalocean.com/community/tags/product-documentation/tutorials. [Internet; descargado 22-abril-2018].

[17] Comisión Económica para América Latina y el Caribe. ¿qué son los datos abiertos? https://biblioguias.cepal.org/EstadoAbierto/datospublicos. [Internet; descargado 16-mayo-2018].

[18] Roberto Pérez. Cómo funcionan las mongodb injection. http://www.abc.es/economia/abci-mapa-paro-espana-peores-y-mejores-provincias-para-encontrar-trabajo-noticia.html, octubre 2017. [Internet; descargado 22-abril-2018].

[19] Globus Soft. An introduction to web scraping. https://globussoft.com/an-introduction-to-web-scraping/, julio 2014. [Internet; descargado 16-mayo-2018].

[20] Wikipedia. Choropleth map — wikipedia, la enciclopedia libre. https://en.wikipedia.org/w/index.php?title=Choropleth_map&oldid=832756109, marzo 2018. [Internet; descargado 23-abril-2018].

[21] Wikipedia. Nosql — wikipedia, la enciclopedia libre. https://en.wikipedia.org/w/index.php?title=NoSQL&oldid=836191884, abril 2018. [Internet; descargado 23-abril-2018].

[22] Wikipedia. Web scraping — wikipedia, la enciclopedia libre. https://en.wikipedia.org/w/index.php?title=Web_scraping&oldid=836897411, abril 2018. [Internet; descargado 23-abril-2018].