



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 — 22 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, 22 de mayo de 2018

Vº. Bº. del Tutor:

Vº. Bº. del Tutor:

D. 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 general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	5
2.1. Objetivos funcionales	5
2.2. Objetivos técnicos	6
Conceptos teóricos	7
3.1. Datos públicos	7
3.2. Bases de datos no relacionales	8
3.3. Web scraping	8
3.4. Mapas coropléticos	9
Técnicas y herramientas	11
4.1. Técnicas	11
4.2. Herramientas	12
Aspectos relevantes del desarrollo del proyecto	19
Trabajos relacionados	21
6.1. Crime prediction through urban metrics and statistical learning	21
6.2. Detecting and investigating crime by means of data mining: a general crime matching framework	21

6.3. Mapa del paro	22
6.4. Ciudades	22
6.5. Eurostat	23
Conclusiones y Líneas de trabajo futuras	25
Bibliografía	27

Índice de figuras

3.1. Esquema del funcionamiento de web scraping	9
3.2. Mapa coroplético del paro en las provincias españolas	10
4.3. Certificado de codacy del proyecto	17
6.4. Paro por comunidades autónomas	22
6.5. Bienestar por ciudades (Colombia)	23
6.6. Porcentaje de empleo por país (Europa)	24

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, como por ejemplo, cuándo se van a cometer crímenes incluso antes de que se cometan, como ocurría en la película ‘Minority Report’ [1], pero ahora ya es una realidad. Otro ejemplo es el reciente uso de *big data* en China como un sistema disciplinario [12]. En ambos ejemplos destaca la importancia de los datos demográficos y sociológicos que hasta ahora a penas se han tratado en el ámbito español.

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 [13] [15] a partir de datos demográficos y sociológicos con el objetivo de determinar los lugares donde asignar más recursos para reducir estos índices de criminalidad.

Se dice que los datos se están convirtiendo en el nuevo petróleo, en el sentido de que con unos datos de buena calidad pueden descubrirse relaciones insospechadas hasta el momento. Una fuente importante de datos son los datos públicos, pero tiene inconvenientes, estos datos se obtienen de múltiples fuentes, por lo que cada una de ellas nos proporciona la información organizada de forma diferente y en distintos formatos, también podrían ser accesibles de modo diferente, lo que nos complica la labor al analizarlos.

El objetivo de este trabajo es llevar estos estudios al ámbito español y unificar estas fuentes de datos de forma que el acceso sea uniforme, mucho más sencillo y permitiendo combinar fuentes de datos de forma directa. Se ha realizado 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) y facilitar la exportación de los datos para después ser utilizados en procesos de minería de datos de forma directa.

Por último, agradecimiento a la idea original de Roberto Cuesta, capitán de la guardia civil y estudiante del doctorado de la Universidad de Burgos, con una tesis enfocada a la aplicación de técnicas de minería de datos en el ámbito criminológico.

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 municipio¹.
 - Contratos registrados por municipio².
 - Demandantes de empleo por municipio³.
- Estadísticas de la renta por municipio (Agencia tributaria)⁴.
- Instituto Nacional de Estadística (INE)
 - Estadísticas de población por sexo y edad⁵.
 - Relación de municipios y códigos por provincias⁶.

Material entregado

Material adjunto a la memoria:

- Aplicación web en Flask.
- Scripts para la integración de fuentes de datos.
- Scripts para la ejecución.

¹<https://datos.gob.es/catalogo/e00142804-paro-registrado-por-municipios>

²<http://datos.gob.es/es/catalogo/e00142804-contratos-por-municipios>

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

⁴https://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/a2011/10/&file=00000001.px&L=0>

⁶<http://www.ine.es/daco/daco42/codmun/codmunmapa.htm>

- Aplicación de escritorio para juntar csv.
- Anexos.

Recursos disponibles en internet:

- [Página web del proyecto⁷](#) .
- [Repositorio del proyecto⁸](#) .

⁷Página web del proyecto: <https://tfg-datos-publicos.nanoapp.io/>

⁸Repositorio del proyecto: <https://github.com/IvanArjona/TFG-Datos-publicos>

Objetivos del proyecto

Este trabajo pretende colocar la primera piedra en un proyecto mucho mayor y más ambicioso.

El objetivo es desarrollar un sistema que permita integrar múltiples fuentes de datos para trabajar con ellas de manera unificada. Y sentar las bases para que, en el futuro, se puedan integrar más datos, formas de visualizarlos y procesarlos.

El usuario objetivo de esta aplicación es el científico de datos que quiera encontrar patrones entre datos públicos a nivel municipal. Los usuarios podrán ser académicos o también periodistas, políticos o servicios públicos.

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, de forma que el administrador pueda añadir nuevos conjuntos de datos de forma sencilla.

- 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 herramientas 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

A continuación se van a explicar los conceptos teóricos más importantes del proyecto, de modo que se pueda entender con exactitud el proyecto en su totalidad.

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 [21].

Los gobiernos tienen la capacidad de obtener grandes cantidad de información sobre la población a través de varios organismos (como podría 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 [3].

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 [28], también llamadas NoSQL (‘Not Only SQL’) son bases de datos optimizadas para ser utilizadas con modelos de datos sin esquema y potencialmente escalables.

A diferencia de las bases de datos relacionales, aquí no hay tablas, 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 [29] es una técnica para extraer información de sitios web directamente de su código fuente sin utilizar APIs⁹ 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.

⁹Interfaz de programación de aplicaciones

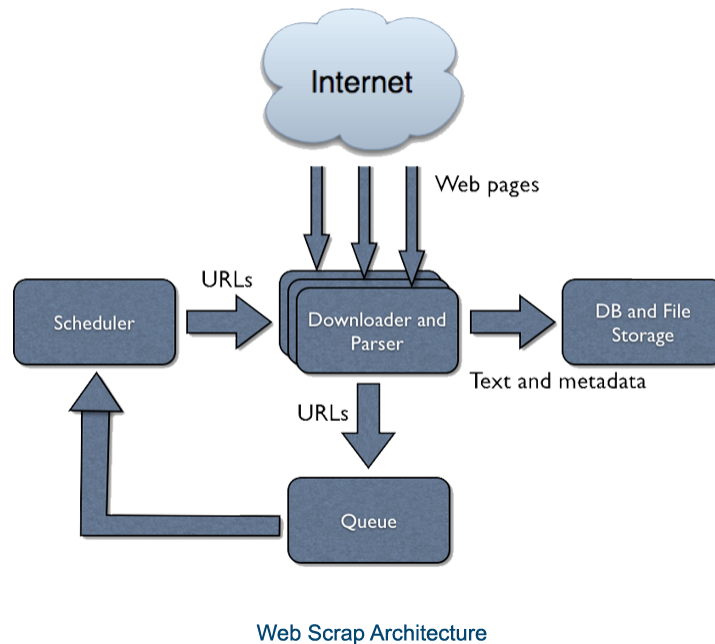


Figura 3.1: Esquema del funcionamiento de web scraping [23]

3.4. Mapas coropléticos

Un mapa coroplético [27] 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.

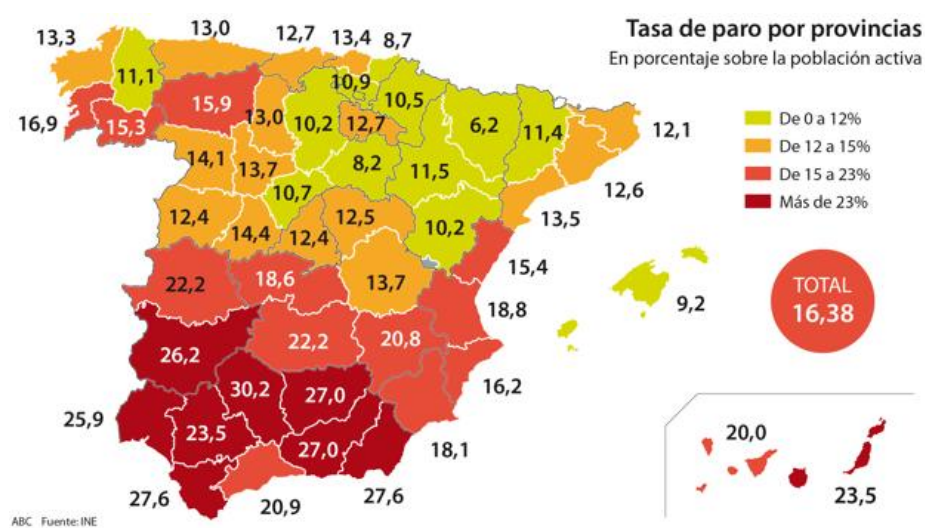


Figura 3.2: Mapa coroplético del paro en las provincias españolas [22]

Técnicas y herramientas

En esta sección se van a explicar, por un lado, las técnicas que se han seguido durante el desarrollo del proyecto y por otro, las herramientas utilizadas, que se han dividido en herramientas de desarrollo, gestión y documentación.

4.1. Técnicas

Metodología ágil

Este proyecto se ha realizado siguiendo los principios del manifiesto ágil [14]. En concreto se ha seguido la metodología *scrum* [2] 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 permite que los requisitos cambien a lo largo del proyecto, en lugar de fijarse al principio, consiguiendo un software de mejor calidad.

Un software creado con unos requisitos claros y fijos desde el principio puede ser un software de muy alta calidad y podría realizarse con una metodología tradicional, pero para proyectos exploratorios como este, en los que los requisitos pueden cambiar, es mejor utilizar una metodología ágil.

Para seguir esta metodología se han utilizado las **issues de Github**¹⁰ (4.2) y los tableros Kanban de **ZenHub**¹¹ (4.2) para organizar la pila de tareas a realizar en cada sprint.

¹⁰issues de Github: <https://github.com/IvanArjona/TFG-Datos-publicos/issues>

¹¹ZenHub: <https://www.zenhub.com/>

DevOps

DevOps [5] 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. Apoyándonos en herramientas como Nanobox (4.2) para facilitar este proceso.

4.2. Herramientas

Herramientas de desarrollo

A continuación se van a explicar las herramientas utilizadas para desarrollar el proyecto.

MongoDB

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

MongoDB¹² [18] 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 [17] como herramienta para integrar MongoDB en python.

Tanto Riak como Cassandra son también bases de datos NoSQL, se descartaron porque no ofrecen soporte para equipos 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.

¹²MongoDB: <https://www.mongodb.com/>

- Herramienta elegida: **DigitalOcean**.

Digital Ocean¹³ 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** [20].

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 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**.

Nanobox¹⁴ es una herramienta que nos permite desplegar nuestra aplicación sin centrarnos en la infraestructura del servidor [19].

Para ello enlazamos nuestra cuenta de un proveedor en la nube (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 nube externos, se descartó por lo ya explicado en el punto anterior.

¹³Digital Ocean: <https://www.digitalocean.com/>

¹⁴Nanobox: <https://nanobox.io/>

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*) [9].

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

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 [16]. Este paquete nos permite visualizar datos manipulados con Python y visualizarlos como mapas utilizando para ello *LeafletJS* [10].

Se ha elegido esta herramienta porque nos permite representar mapas coropléticos (3.4) y mapas con clusters de datos [7] utilizando estructuras de datos geográficas personalizadas (*geojson*) [11].

Dynatable

- Herramientas consideradas: [Dynatable](#), [Datatables](#).
- Herramienta elegida: [Dynatable](#).

¹⁵Flask: <http://flask.pocoo.org/>

¹⁶Folium: <http://python-visualization.github.io/folium/>

Dynatable¹⁷ es un framework de javascript para visualizar tablas de una manera más clara y ordenadas [6].

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.

PyCharm

- Herramientas consideradas: **Visual Studio Code**, **PyCharm**.
- Herramienta elegida: **PyCharm**.

PyCharm¹⁸ es un IDE (*Entorno de Desarrollo Integrado*) para Python basado en IntelliJ. Posee herramientas para ayudarnos con el desarrollo como el autocompletado y herramientas para refactorizar de forma automática.

Se ha considerado porque, a diferencia de otros IDEs similares, tiene soporte para Flask y MongoDB, lo que nos facilita programar más rápidamente.

En este proyecto se ha utilizado la versión *Professional*, pero con la versión *Community* gratuita sería suficiente.

Herramientas de gestión

A continuación se van a describir las herramientas que se han utilizado para gestionar el proyecto.

Github

- Herramientas consideradas: **Github**, **BitBucket**, **GitLab**.
- Herramienta elegida: **Github**.

GitHub¹⁹ es un servicio web para alojar proyectos utilizando un sistema de control de versiones *Git*.

Se ha utilizado esta herramienta para alojar el **repositorio del proyecto** llevando un registro de todos los cambios realizados desde el inicio.

Además se han utilizado las *Issues* y *Milestones* de GitHub para planificar las tareas a realizar en cada sprint.

¹⁷Dynatable: <https://www.dynatable.com/>

¹⁸PyCharm: <https://www.jetbrains.com/pycharm/>

¹⁹GitHub: <https://github.com/>

ZenHub

- Herramientas consideradas: Github Projects, ZenHub, GitKraken Glo.
- Herramienta elegida: ZenHub.

ZenHub²⁰ es una herramienta de gestión de proyectos integrada en Github mediante una extensión para el navegador. Nos proporciona un tablero kanban sobre el que mover las *issues* entre varios estados.

Las tareas se pueden estimar estableciendo un número ‘*Story points*’ basados en la serie de Fibonacci en función del trabajo que se cree que se va a necesitar. Esto ha sido especialmente útil para planificar el tiempo empleado en cada tarea [30].

También se ha utilizado esta herramienta para extraer los *gráficos burn-down* de cada sprint presentes en el anexo.

GitKraken

- Herramientas consideradas: GitKraken, Github Desktop, Sourcetree.
- Herramienta elegida: GitKraken.

GitKraken²¹ es un cliente con interfaz gráfica para gestión de proyecto *Git* para Windows, Mac y Linux. Nos permite realizar todas las acciones posibles con git, pero de manera mucho más sencilla y visual.

Permite sincronizarse con Github y otros servicios de alojamiento de repositorios de forma automática, de forma que el contenido del repositorio local y el remoto sea consistente.

Codacy

- Herramientas consideradas: SonarQube, Code Climate, Codacy.
- Herramienta elegida: Codacy.

Codacy²² es una herramienta de análisis de calidad de código que ayuda a los desarrolladores a realizar código de más rápido y de mayor calidad.

²⁰ZenHub: <https://www.zenhub.com/>

²¹GitKraken: <https://www.gitkraken.com/>

²²Codacy: <https://www.codacy.com/>

Tiene métricas para medir la complejidad ciclomática, código duplicado, cubrimiento por tests y estadísticas para cada commit o pull request [8].

Durante el desarrollo de este proyecto se ha intentado mantener siempre una Certificación A en Codacy con todas las métricas al 100 % para asegurar la calidad del código.

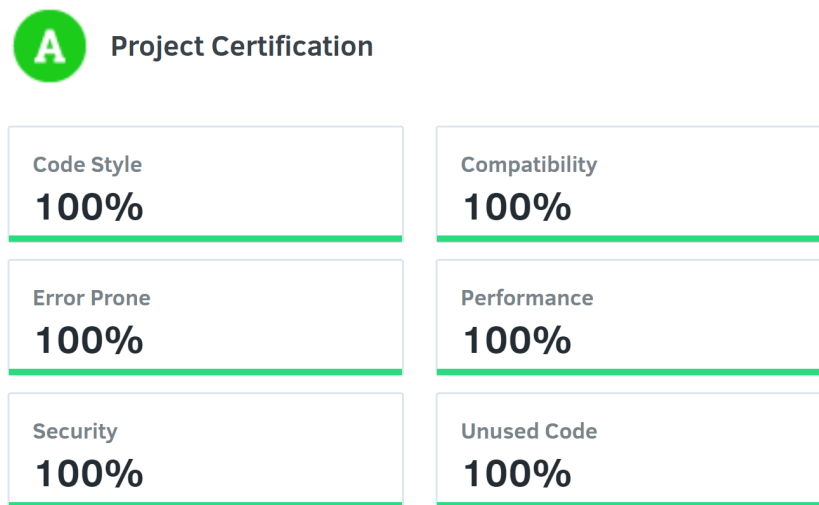


Figura 4.3: Certificado de codacy del proyecto [4]

Herramientas de documentación

LaTeX

- Herramientas consideradas: Microsoft Word, LaTeX, LibreOffice, OpenOffice.
- Herramienta elegida: LaTeX²³.

Esta memoria está escrita en L^AT_EX. Se trata de un sistema de preparación de documentos para la creación de documentos con una alta calidad tipográfica. Se usa frecuentemente en la escritura de libros y artículos científicos, entre otra cosas por la facilidad de incluir expresiones matemáticas [25] [26].

²³LaTeX: <https://www.latex-project.org/>

Para utilizar esta herramienta se ha utilizado **TeXstudio**²⁴. Es un editor gratuito y multiplataforma para editar documentos escritos en Latex. Incluye iconos para localizar los comandos más comunes, corrector ortográfico, visor de PDF y descarga los paquetes de forma automática.

Este documento se ha escrito utilizando la siguiente plantilla de Latex [24].

StarUML

- Herramientas consideradas: **StarUML**, **ArgoUML**, **Astah**.
- Herramienta elegida: **StarUML**.

StarUML²⁵ es una herramienta para modelar diagramas UML que nos permite crear diagramas de clases, objetos, casos de uso, componente, estructurales, de comunicación, de estado y de actividad y entidad-relación.

Funciona de manera gráfica, arrastrando objetos y conectándolos entre ellos permitiendo añadir atributos tanto a los objetos como a las relaciones.

²⁴TeXstudio: <https://www.texstudio.org/>

²⁵StarUML: <http://staruml.io/>

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. Crime prediction through urban metrics and statistical learning

En este artículo se estudian las causas que pueden aumentar el índice de criminalidad utilizando técnicas de minería de datos. Para ello se utiliza un *Random Forest* para predecir el crimen a partir de índices urbanos sobre homicidios. [13]

Como resultado se obtiene un 97 % de precisión en la predicción de índices de criminalidad y se ordenan los indicadores que causan estos crímenes en función de su importancia.

6.2. Detecting and investigating crime by means of data mining: a general crime matching framework

En este artículo, al igual que el anterior, se utilizan técnicas de minería de datos para identificar las características que pueden llevar a cometer un crimen [15].

En esta aproximación se toma como entrada informes policiales narrativos escritos en texto plano. De estos textos se extraen las características para determinar la gravedad del delito utilizando análisis de grupos mediante *clustering*.

6.3. Mapa del paro

Mapa del paro²⁶ es una página web que nos permite visualizar el paro por municipios, provincias y comunidades en un mapa coroplético. Ver figura 6.4.

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.

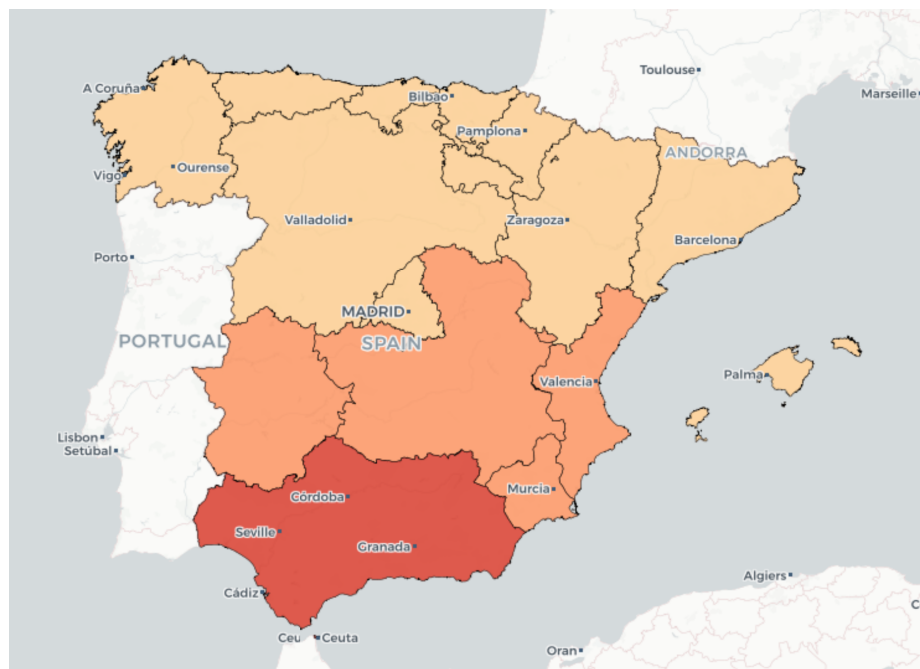


Figura 6.4: Paro por comunidades autónomas

6.4. Ciudatos

Ciudatos²⁷ nos permite visualizar varios datos públicos de Colombia mediante gráficas de barras y posicionarlos en el mapa. Ver figura 6.5.

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

²⁶Mapa del paro: <http://mapadelparo.com/>

²⁷Ciudatos: <http://www.ciudatos.com/visualizacion>

Como ventaja nuestra aplicación tiene datos de España y permite visualizar datos resultantes de juntar varias subconsultas.

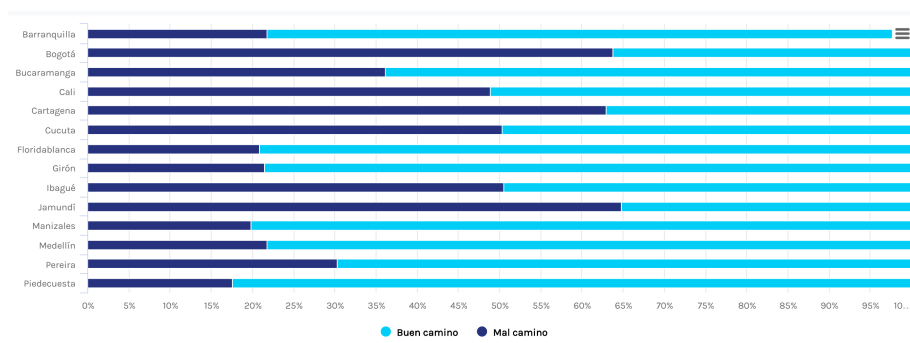


Figura 6.5: Bienestar por ciudades (Colombia)

6.5. Eurostat

Otra representación de datos públicos en mapas es [Eurostat²⁸](http://ec.europa.eu/eurostat/en/web/lfs/statistics-illustrated).

En esta representación se muestra en el mapa los porcentajes de empleo en cada país de Europa. Ver figura 6.6.

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

²⁸Eurostat: <http://ec.europa.eu/eurostat/en/web/lfs/statistics-illustrated>

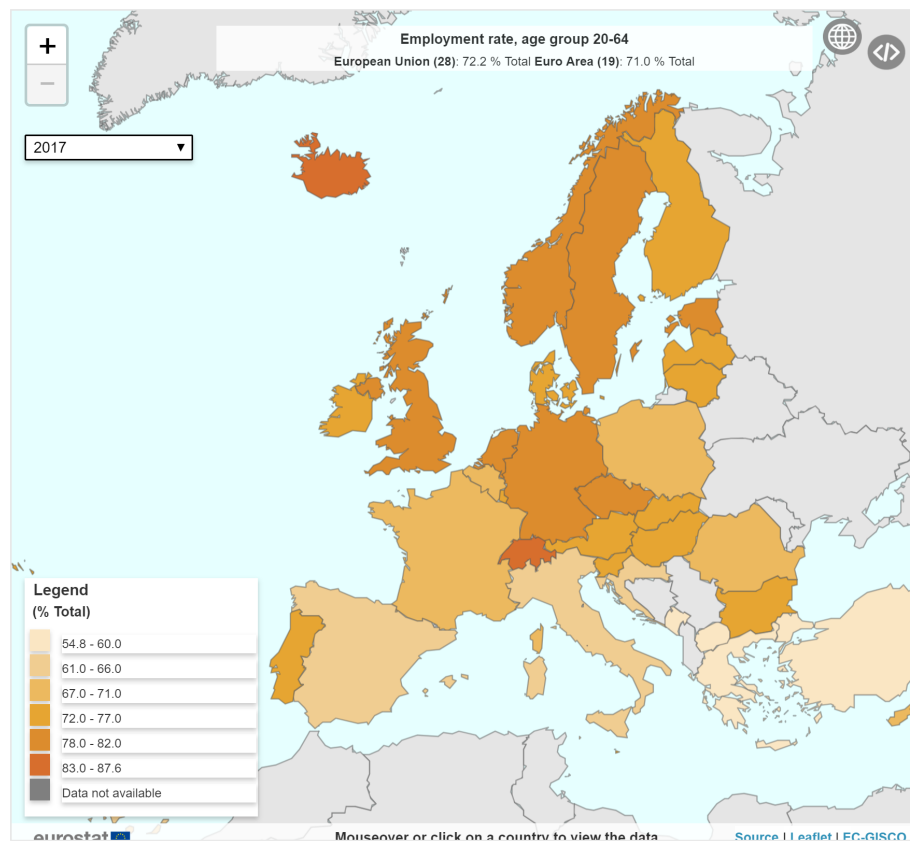


Figura 6.6: Porcentaje de empleo por país (Europa)

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] Minority report. Dir. Steven Spielberg. Amblin Entertainment, Dream-Works, 20th Century Fox, 2002.
- [2] Juan Palacio Alexander Menzinsky, Gertrudis López. *Scum Manager*. Scrum Manager, 2016.
- [3] Iniciativa Aporta. Acerca de la iniciativa aporta. <http://datos.gob.es/es/acerca-de-la-iniciativa-aporta>. [Internet; descargado 16-mayo-2018].
- [4] Codacy. Project certification – code dashboard – codacy. <https://support.codacy.com/hc/en-us/articles/207993995-Code-Dashboard#2-project-certification>. [Internet; descargado 16-mayo-2018].
- [5] 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].
- [6] Dynatable. Dynatable.js - jquery plugin for html5+json interactive tables and more. <https://www.dynatable.com/>. [Internet; descargado 06-mayo-2018].
- [7] Folium. Markercluster – folium. <http://nbviewer.jupyter.org/github/python-visualization/folium/blob/master/examples/MarkerCluster.ipynb>. [Internet; descargado 06-mayo-2018].
- [8] GitHub. Github marketplace – codacy. <https://github.com/marketplace/codacy>. [Internet; descargado 16-mayo-2018].

- [9] Miguel Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2014.
- [10] LeafletJS. Documentation - leaflet. <https://leafletjs.com/reference-1.3.0.html>. [Internet; descargado 06-mayo-2018].
- [11] LeafletJS. Using geojson with leaflet. <https://leafletjs.com/examples/geojson/>. [Internet; descargado 06-mayo-2018].
- [12] Sidney Leng. In china, big data is watching you ... and that could be a huge 'challenge to the west' – south china morning post. <http://www.scmp.com/news/china/policies-politics/article/2129897/china-big-data-watching-you-and-could-be-huge-challenge>, enero 2018. [Internet; descargado 22-mayo-2018].
- [13] 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.
- [14] Jim Highsmith Martin Fowler. The agile manifesto. *Software Development*, 9:28–35.
- [15] 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.
- [16] MongoDB. Folium documentation. <http://python-visualization.github.io/folium/>. [Internet; descargado 06-mayo-2018].
- [17] MongoDB. Pymongo 3.6.1 documentation. <https://api.mongodb.com/python/3.6.1/>. [Internet; descargado 22-abril-2018].
- [18] MongoDB. What is mongodb? <https://www.mongodb.com/what-is-mongodb>. [Internet; descargado 22-abril-2018].
- [19] Nanobox. Nanobox documentation. <https://docs.nanobox.io/>. [Internet; descargado 22-abril-2018].
- [20] Digital Ocean. Product documentation. <https://www.digitalocean.com/community/tags/product-documentation/tutorials>. [Internet; descargado 22-abril-2018].
- [21] Comisión Económica para América Latina y el Caribe. ¿qué son los datos abiertos? <https://biblioguias.cepal.org/EstadoAbierto/datospublicos>. [Internet; descargado 16-mayo-2018].

- [22] Roberto Pérez. Cómo funcionan las monogodb injection. <http://www.abc.es/economia/abci-mapa-paro-espana-peores-y-mejores-provincias-para-encontrar-trabajo-201710-noticia.html>, octubre 2017. [Internet; descargado 22-abril-2018].
- [23] Globus Soft. An introduction to web scraping. <https://globussoft.com/an-introduction-to-web-scraping/>, julio 2014. [Internet; descargado 16-mayo-2018].
- [24] ubutfgm. Plantilla latex. <https://github.com/ubutfgm/plantillaLatex>. [Internet; descargado 20-mayo-2018].
- [25] Wikibooks. Latex — wikibooks, open books for an open world. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=107741266>, mayo 2018. [Internet; descargado 16-mayo-2018].
- [26] Wikibooks. Latex — wikipedia, la enciclopedia libre. <https://en.wikibooks.org/w/index.php?title=LaTeX&oldid=3425262>, mayo 2018. [Internet; descargado 16-mayo-2018].
- [27] 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].
- [28] Wikipedia. Nosql — wikipedia, la enciclopedia libre. <https://en.wikipedia.org/w/index.php?title=NoSQL&oldid=836191884>, abril 2018. [Internet; descargado 23-abril-2018].
- [29] 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].
- [30] ZenHub. Estimate work using story points. <https://www.zenhub.com/guides/what-are-software-estimates>. [Internet; descargado 16-mayo-2018].