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 2

Presentado por Sergio López Bueno
en Universidad de Burgos — 3 de julio de 2019
Tutor: 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.

Exponen:

Que el alumno D. Sergio López Bueno, con DNI 71306605G, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "Aplicación Web para la recopilación, tratamiento y visualización de datos públicos 2".

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

En Burgos, 1 de julio de 2019

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

## Resumen

Este proyecto es la continuación del trabajo realizado por Iván Arjona Alonso *Aplicación Web para la recopilación, tratamiento y visualización de datos públicos*. En esta segunda parte del proyecto se ampliarán las funcionalidades de la aplicación web, añadiendo nuevas fuentes de datos y obteniendo predicciones sobre las tasas de criminalidad que existen en los distintos municipios de España.

Los datos han sido obtenidos de diferentes formas: derivados de otros datos proporcionados por el ministerio de Hacienda, utilizando diferentes técnicas de web scraping o extrayendo la información directamente de archivos descargados de internet.

Estos datos se han guardado en una base de datos no relacional *NoSQL* y tras un trabajo previo de procesamiento y normalización de atributos, se han utilizado para obtener las predicciones de criminalidad.



## Descriptores

Aplicación web, bases de datos NoSQL, datos públicos, estudio sociológico, machine learning, mapas temáticos, minería de datos, web scraping.

## Abstract

This project is the continuation of the work conducted by Iván Arjona Alonso *Web application for the collection, processing and visualization of public data.*

In this second part of the project the functionalities of the web application will be expanded, adding new data sources and obtaining predictions about the crime rates exisitng in the different municipalities of Spain.

The data has been obtained in different ways: derived from other data provided by the Ministry of Hacienda using different web scraping techniques or extracting information directly from files downloaded from the internet.

Data has been stored in a non-relational *NoSQL* database and after a previous work of processing and normalization of attributes, they have been used to obtain the predictions of crime.



## Keywords

Web application, NoSQL databases, public data, sociological study, machine learning, thematic maps, data mining, web scraping.

# Índice general

# Índice de figuras

# Índice de tablas

# Introducción

En este trabajo se hará uso de la *minería de datos* para poder analizar y encontrar patrones a diferentes fuentes de datos. En la primera parte del proyecto ya se trabajó este tema pero en esta segunda parte se profundizará mucho más con un volumen de datos superior cuyas formas de obtención difieren bastante entre ellas.

También se tratarán otros temas como el *web scraping*, técnica que nos permitirá obtener nuevas fuentes de datos de diferentes sitios web. Se estudiarán y utilizarán diferentes tipos de web scraping para alcanzar este cometido.

Otro tema en el que se trabajará será el *machine learning* ya que se utilizará todo el trabajo anterior de minería y web scraping para calcular predicciones a partir de los datos obtenidos anteriormente. Para poder realizar estas predicciones también se hará un trabajo de procesamiento y normalización de los datos para que puedan ser más útiles para el clasificador. Los datos que se van a predecir son los datos de criminalidad que se pudieran producir en los diferentes municipios de España, posteriormente se obtendrán qué factores demográficos o sociológicos son los que más afectan en este ámbito criminológico.

El estudio de variables relacionadas con la comisión de un crimen es algo que ya está siendo investigado y en este proyecto se utilizarán las técnicas ya mencionadas para comprobar la influencia de estas variables dentro del territorio español.

## 1.1. Estructura de la memoria

A continuación se muestra las partes en las que está dividida el resto de la memoria:

- Objetivos del proyecto: Explicación de los objetivos que afronta este proyecto.
- Conceptos teóricos: Descripción de los conceptos utilizados en el trabajo.
- Técnicas y herramientas: Descripción de las técnicas utilizadas durante el desarrollo del proyecto así como de las herramientas de gestión, documentación y desarrollo empleadas.
- Aspectos relevantes del desarrollo del proyecto: Explicación de los aspectos más interesantes del trabajo así como de problemas encontrados.
- Trabajos relacionados: Explicación de diferentes artículos y sitios web que estén relacionados o hagan un trabajo similar al realizado en este proyecto.
- Conclusiones y líneas de trabajo futuras: Explicación de las conclusiones obtenidas tras la realización del trabajo además de las posibles tareas con las que continuar con el proyecto.

### Flujo de datos

A continuación se muestra un diagrama de flujo que muestra de forma simplificada el recorrido de la información en este proyecto. Se han omitido numerosos pasos y archivos utilizados en este trabajo para mayor claridad en el diagrama.

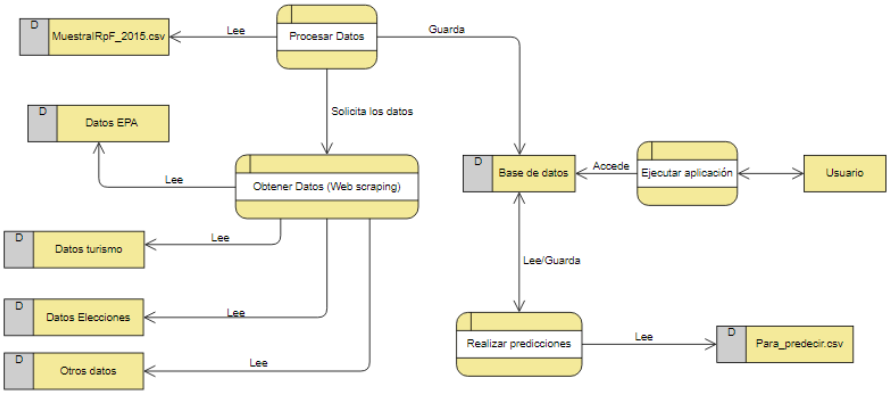


Figura 1.1: Diagrama de flujo del proyecto

## Fuentes de datos

La aplicación mantiene y añade nuevas fuentes de datos. Estas fuentes son las siguientes:

- Servicio Público de Empleo Estatal (SEPE).
  - Paro registrado por municipio[1].
  - Contratos registrados por municipio[2].
  - 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, edad y procedencia[5].
  - Relación de municipios y códigos por provincias[6].
- Ministerio del interior (MIR)
  - Resultados de elecciones municipales y generales por año[7].
- Declaraciones IRPF 2015.[8]

[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
[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] http://www.infoelectoral.mir.es/infoelectoral/min/areaDescarga.html?method=inicio
[8] Proporcionadas en un CD por el ministerio de hacienda

- Encuesta de población activa (EPA).[9]
- Turismo: alojamientos por municipio.[10]
- Delitos por municipio.[11]

| Fuente de datos | TFG 2018 | TFG 2019 |
|---|---|---|
| Municipios | ✓ | ✓ |
| Irpf2015 | ✗ | ✓ |
| Epa | ✗ | ✓ |
| AeatRenta | ✓ | ✓ |
| InePoblacion | ✓ | ✓ |
| Sepe | ✓ | ✓ |
| MirElecciones | ✓ | ✓ |
| MirEleccionesGenerales | ✗ | ✓ |
| Turismo | ✗ | ✓ |
| Predicciones | ✗ | ✓ |

Tabla 1.1: Tabla con las fuentes de datos usadas en cada version de TFG

[9] https://datosmacro.expansion.com/paro/espana/municipios
[10] https://www.booking.com/index.es.html
[11] Proporcionados por un doctorando del Programa de Investigación en Ingeniería.

# Objetivos del proyecto

Este trabajo conserva los objetivos que tenía su predecesor y añade otros nuevos. El objetivo principal que tenía en su origen era el de integrar diferentes fuentes de datos en un mismo entorno para poder analizarlos y encontrar diferentes patrones.

En esta segunda parte además de seguir añadiendo más fuentes para su análisis, se profundiza más en esos objetivos y se busca encontrar patrones para poder obtener predicciones de los datos introducidos. El objetivo final de este proyecto es calcular los posibles delitos que puedan darse en los distintos municipios de España gracias a los diferentes datos que tenemos sobre esos territorios y su población (o que se han calculado a través de ellos). De esta forma, un objetivo final a futuro podría ser el de crear un sistema que pudiese optimizar la dotación de efectivos de las fuerzas y cuerpos de seguridad del estado en función de la estimación de delitos.

Los objetivos podrían dividirse en dos categorías, 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

- Añadir nuevas fuentes de datos que puedan servir para análisis demográficos, sociológicos, económicos o criminológicos entre otros.

- Hacer uso de las fuentes de datos para calcular índices que muestren la desigualdad de ingresos, tasas de paro y ocupaciones hoteleras.

- Extraer la información de diferentes sitios web y correlacionarla para poder trabajar con los datos conjuntamente.
- Integrar todos los datos en una misma base de datos haciendo uso de los mapas coropléticos ya implementados en la aplicación.
- Utilizar los datos obtenidos para la predicción de nuevos datos.

## 2.2. Objetivos técnicos

- Seguir usando las herramientas software utilizadas en la primera parte del proyecto. (Flask como framework de la aplicación y MongoDB como base de datos no relacional)
- Utilización de Github como sistema de control de versiones junto a su aplicación de escritorio Github Desktop.
- Uso de la librería Pandas en Python para trabajar con las tablas y series temporales de la aplicación y de la librería Sklearn para el cálculo de las predicciones.
- Uso de diferentes técnicas de Web Scraping para la obtención de los datos de los sitios web. Principalmente uso de expresiones regulares para analizar sintácticamente el código HTML o la utilización de la herramienta Selenium para hacer uso del buscador de la página web.

# Conceptos teóricos

Como ya se ha explicado con anterioridad este proyecto continúa el trabajo realizado por Iván Arjona [5] por lo que no se profundizará en los conceptos ya desarrollados en la primera parte del proyecto y solo se indicará el uso que se les ha dado en este trabajo.

## 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* [5][20]

A parte de los datos que ya se incluían con anterioridad en este trabajo se han añadido los siguientes datos públicos:

- Muestra IRPF 2015: Se han usado los datos proporcionados por el ministerio de hacienda y función pública. Se trata de una muestra de 2.7 millones de declaraciones con información relativa a 512 variables. El tamaño de la muestra está calculado para un error, en la media de la variable renta de un 1.1 %.
- Paro por municipios: Obtenidos del periódico Expansión.[9]
- Datos turismo: Obtenidos de la página web Booking.[6]
- Datos de criminalidad: Proporcionados por un doctorando del Programa de Investigación en Ingeniería. Los datos fueron transformados de manera que aunque se sigue manteniendo la proporción relativa de los delitos en los municipios, no se puede obtener su valor original.

Para la muestra del IRPF se ha calculado la desigualdad de renta que existe en cada municipio, para ello se ha utilizado el coeficiente Gini que se explica a continuación. [28]

## Coeficiente Gini

El coeficiente Gini es una medida creada por el estadístico italiano Corrado Gini que sirve para medir la desigualdad, normalmente utilizada para calcular la desigualdad de ingresos en una población concreta. [32] Este coeficiente tiene un valor entre cero y uno, donde cero representa la igualdad absoluta y uno todo lo contrario.

Existen dos formas de calcular el índice Gini:

1. Mediante la curva de Lorenz: El coeficiente se calcula obteniendo la proporción entre el área que ocupa la zona desde la linea de perfecta igualdad hasta la curva de Lorenz y el area total bajo la línea de igualdad absoluta.

   En el siguiente gráfico puede verse las áreas de las que se habla.


Figura 3.2: Cálculo del coeficiente Gini [17]

2. Mediante la fórmula de Brown: Para facilitar este cálculo a menudo se utiliza la fórmula de Brown para calcular en índice Gini. La fórmula es la siguiente:

$$G = \left|1 - \sum_{n=1}^{n-1}(X_{k+1} - X_k)(Y_{k+1} + Y_k)\right|$$

Donde:

- X es la proporción acumulada de la variable población
- Y es la proporción acumulada de la variable ingresos [32]

En este proyecto se han probado las dos formas para calcular el índice Gini. En el fichero *Gini.py* se encuentran dos métodos, el primero realiza el cálculo mediante la curva de Lorenz [1], mientras que el segundo [3] utiliza una derivación de la fórmula de Brown [2]. Al final se decidió utilizar el primer

método pero se ha mantenido el segundo por si interesara cambiar la forma del cálculo en el futuro.

#### Índice de Reynolds-Smolensky

El índice de Reynolds-Smolensky (IRS) es una medida que expresa el grado de redistribución del impuesto. En este caso se calcula mediante la diferencia del índice Gini antes de impuestos menos el índice Gini después de ellos.

Es decir, es una medida que nos sirve para saber el grado en el que los impuestos reducen o aumentan la desigualdad económica.

## 3.2. Web scraping

Web scraping es una técnica de obtención de información de algún sitio web de manera automatizada. El objetivo suele ser el de extraer un tipo de información concreta para poder guardarla en una base de datos y poder usar esa información posteriormente. [4]

Existen diferentes técnicas de web scraping, las principales serían las siguientes:

- Copiar y pegar humano
- Uso de expresiones regulares para encontrar coincidencias en el texto de una página.
- Revisión de código HTML obtenido haciendo peticiones HTTP.
- Análisis de la estructura DOM del documento.
- Aplicaciones de web scraping
- Métodos de inteligencia artificial.

A continuación se explicarán más detalladamente las dos técnicas usadas en este proyecto.

### Análisis de la estructura DOM del documento

DOM (Document Object Model) es una interfaz de programación para documentos HTML, XHTML y XML que define de que forma está estructurado el documento para que otros programas puedan acceder a él.

De esta forma podemos buscar, añadir o cambiar elementos concretos dentro del código del documento y trabajar con ellos para obtener los datos que necesitemos.

En este trabajo se ha utilizado Selenium como la herramienta que nos permita hacer esto a través del navegador web Firefox.

### Revisión de código HTML

Esta técnica consiste en en hacer ciertas peticiones HTTP para conseguir el código HTML deseado. Una vez obtenido se revisa el código en busca de los datos y se repite la operación en caso necesario.

En el caso concreto de este trabajo se han utilizado además expresiones regulares para encontrar los nombres necesarios para acceder a las diferentes direcciones URL donde se encuentran los datos a buscar.

## 3.3. Machine learning

Como ya sabemos el machine learning o aprendizaje automático *es el estudio científico de algoritmos y modelos estadísticos que utilizan los sistemas informáticos para realizar una tarea específica de manera efectiva sin utilizar instrucciones explícitas, confiando en los patrones y la inferencia*[34]

En este proyecto se ha utilizado el machine learnig para encontrar solución a un problema de regresión cuyo objetivo es calcular las predicciones del número de delitos que se pudieran cometer en los diferentes municipios de España, concretamente se ha utilizado la herramienta Scikit-learn que se explicará más adelante.

Existen numerosos algoritmos de aprendizaje automático (redes neuronales, arboles de decisión, algoritmos genéticos, etc) pero en este trabajo, después de haber estudiado varias opciones se optó por el algoritmo random forest.

## Aprendizaje supervisado

El aprendizaje supervisado es una técnica para resolver problemas a partir de unos datos de entrenamiento. Existen dos tipos de problemas en función de las soluciones que busquemos:

1. Problemas de regresión: Si lo que tenemos que predecir es un valor continuo estaremos hablando de problemas de regresión. Este es el caso en este proyecto, ya que lo que se desea predecir es el "número" de delitos.

2. Problemas de clasificación: En este caso la solución se encuentra dentro de un rango discreto, es decir, la solución es una "etiqueta" que asociar a cada elemento del conjunto a predecir.

## Validación cruzada

La validación cruzada es una técnica que consiste en dividir la muestra que se tiene de los datos de entrenamiento en diferentes conjuntos, utilizando todos los conjuntos menos uno como datos de entrenamiento y el conjunto sobrante como datos para comprobar la precisión de la predicción. Esto lo realiza con todos los conjuntos hasta que al final se dispone de la predicción completa de la muestra con la que se puede valorar la precisión del modelo.[35]

Para este proyecto se ha utilizado la función *cross_val_predict* que se encuentra en la librería *Scikit-learn* dividiendo la muestra en 10 conjuntos.

# Técnicas y herramientas

A continuación se describen las técnicas necesarias para llevar a cabo el proyecto y las herramientas utilizadas.

## 4.1. Técnicas

### Web scraping

Para la obtención de los datos de la encuesta de población activa como de los datos de turismo se tuvo que hacer uso de dos técnicas diferentes de web scraping (ver sección 3.2). Gracias a estas técnicas se pudo obtener la información necesaria de dos sitios web distintos para utilizar los datos en las predicciones futuras.

### Random forest

Random forest [7] o bosque aleatorio es un algoritmo que sirve tanto para problemas de regresión como de clasificación. Esto lo realiza mediante una técnica denominada Bootstrap Aggregation (bagging) que consiste en un conjunto de arboles de decisión que calculan individualmente una solución. A cada árbol se le entrena con una muestra de datos diferente y después se obtiene una solución final haciendo el promedio de las soluciones individuales. Durante el proceso, cada árbol se construye usando un subconjunto de atributos diferente, elegido de manera aleatoria, en cada nodo del árbol. [15]

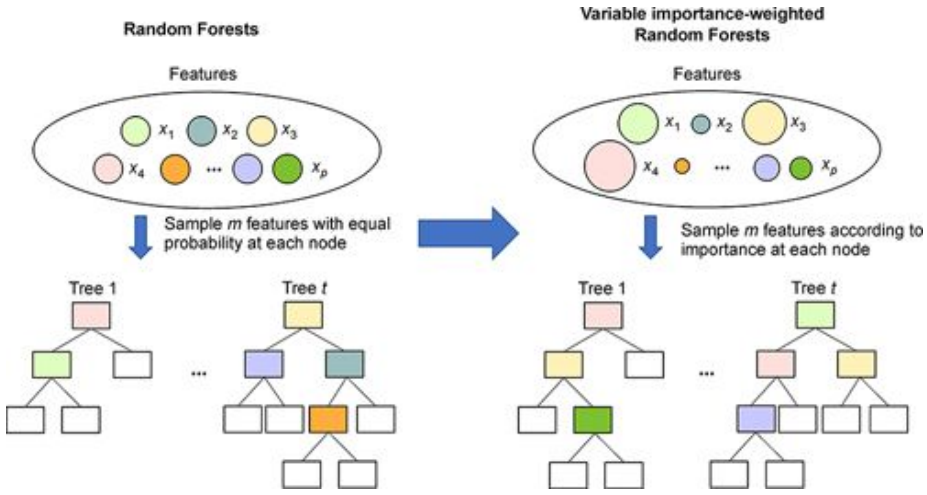


Figura 4.3: Esquema del funcionamiento del Random Forest [36]

### Metodología ágil

Para el desarrollo del proyecto seguido los principios básicos del manifiesto ágil. Se ha planificado y dividido el desarrollo del trabajo en diferentes sprints y se ha utilizado Github para hacer un registro de estos spints así como de las issues que se han ido solventando durante el proyecto.

## 4.2. Herramientas

Dividiremos las herramientas en tres tipos, las herramientas de desarrollo, de gestión y de documentación.

### Herramientas de desarrollo

En esta sección se van a explicar las herramientas utilizadas durante el desarrollo del proyecto.

Destacar que no se volverán a explicar las herramientas utilizadas en la primera parte del proyecto (versión del 2018) [5], las cuales, siguen siendo las bases para la continuación de esta segunda parte. Estas herramientas son las siguientes:

- MongoDB como sistema de bases de datos NoSQL.
- Flask como framework para el funcionamiento de la página.

- La librería Folium para el dibujado de los mapas.
- El framework Dynatable para la visualización de las tablas.
- El framework Bootstrap para el desarrollo de la página HTML y CSS.

A continuación se muestras las nuevas herramientas utilizadas en el trabajo.

### Scikit-learn

Scikit-learn [14] es una librería de Python de software libre que integra diferentes algoritmos de aprendizaje automático para resolver problemas.

Dispone de diversos algoritmos para diferentes tipos de problemas, ya sean supervisados o no supervisados. Entre los algoritmos más comunes se encuentran los algoritmos de regresión, clasificación o análisis de grupos (e.g. bosques aleatorios).

En nuestro caso se trata de un problema supervisado donde la solución es un valor numérico continuo por lo que utilizaremos un algoritmo de regresión. Tras probar diferentes opciones en este trabajo se optó por usar el algoritmo random forest, integrado en esta librería. También se uso la función *feature_importances_* de esta librería para obtener el peso que han tenido las diferentes variables a la hora de realizar las predicciones.

### Selenium

Selenium [26] es un framework de código abierto disponible para numerosos lenguajes de programación (Java, JavaScript, C, PHP o Python entre otros) que permite hacer pruebas a través de un navegador para comprobar el funcionamiento de una aplicación web.

El uso que se le ha dado en este trabajo es algo distinto ya que se ha utilizado como herramienta de webscraping para obtener los datos de turismo buscando en la página web Booking el número de alojamientos de cada municipio.

El motivo por el que se ha optado por esta herramienta en vez de otra enfocada totalmente para webscraping (como pudiera ser Scrapy), a parte de su mayor facilidad de uso, es por su soporte para aplicaciones web que usen JavaScript para mostrar los datos.

### PyDev

Pydev es un plug-in para el entorno de de desarrollo Eclipse que añade soporte para Python. De esta forma mantenemos muchas de las funciones que ya disponía este entorno de desarrollo adaptadas para este nuevo lenguaje. PyDev permite entre otras cosas, autocompletar código, imports automáticos, análisis de código, refactorización, debugger, etc. [23]

Para mayor facilidad de uso y evitar confusiones al trabajar en un mismo IDE en distintos proyectos con diferentes lenguajes de programación se optó por instalar Liclipse.

Liclipse es una versión más ligera de Eclipse que incluye el plug-in PyDev y permite trabajar de forma independiente manteniendo todas las funciones para desarrollar en Python.

### Pandas

Pandas [16] es una librería de Python de código abierto para el manejo de estructuras de datos (series y tablas principalmente) y su correspondiente análisis.

En este proyecto se ha utilizado Pandas para trabajar con las fuentes de datos que se habían obtenido con antelación o para trabajar con los datos leídos de la base de datos que ya se habían introducido con anterioridad.

### Difflib

Difflib es una librería de Python que dispone de varias clases y funciones para la comparación de cadenas.

En este proyecto se usó la función *get_close_matches* que incluye difflib. Como su nombre indica se trata de una función que busca resultados similares a un string determinado dentro de una lista, ambos pasados por parámetro. Finalmente devuelve los resultados que más se acercan a la cadena original ordenados por similitud.

## Herramientas de gestión

Aquí se explican las herramientas usadas para gestionar el proyecto.

### Github

Github es una plataforma web que nos permite tener un control de versiones de nuestro proyecto.

Se ha elegido GitHub junto a su aplicación de escritorio GitHub Desktop para tener este control de versiones de manera rápida y sencilla.

El proyecto se dividió en distintos Sprints que disponían de sus propios issues y milestones para la planificación durante el desarrollo del proyecto.

## Herramientas de documentación

### LaTeX

LaTeX es un sistema para la creación de documentos escritos. Esta pensado para usarlo principalmente en la creación de artículos y libros científicos. Es una herramienta de código abierto que funciona especialmente bien a la hora de trabajar con expresiones matemáticas. [33] [31]

Se ha optado por esta herramienta por la comodidad en el uso de sus plantillas que facilita el trabajo de dar el formato adecuado.

### Overleaf

Como editor LaTeX, se ha utilizado Overleaf, un editor online que se accede mediante un navegador. El código LaTex se compila en el servidor y devuelve el documento PDF correspondiente. Algunas de sus características gratuitas son la creación de proyectos públicos o privados, la revisión ortográfica o la colaboración entre varias personas.

Se ha escogido esta herramienta online para poder acceder a la documentación en cualquier momento y poder compartirla con facilidad durante el desarrollo de la misma.

# Aspectos relevantes del desarrollo del proyecto

Este apartado recoge los aspectos más interesantes del desarrollo del proyecto.

## 5.1. Fuentes de datos

En esta sección se explicará el uso que se le ha dado a las diferentes fuentes de datos introducidas en el proyecto así como las modificaciones que fue necesario realizar para su integración con las demás fuentes de datos.

### Datos de IRPF

IRPF o el Impuesto sobre la Renta de las Personas Físicas *es un tributo de carácter personal y directo que grava, según los principios de igualdad, generalidad y progresividad, la renta de las personas físicas de acuerdo con su naturaleza y sus circunstancias personales y familiares.*[11]

Para la realización de este proyecto se disponía de una muestra de 2.700.000 declaraciones de renta proporcionadas por el ministerio de hacienda y función pública.

El trabajo realizado con estos datos fue el siguiente:

1. Filtrar los datos correctos: algunas declaraciones contenían errores ya que al sumar algunos de sus datos daban como resultado soluciones incoherentes como pudieran ser cuotas negativas o rentas inferiores a su correspondiente base imponible.

2. Agrupación de las declaraciones por municipio: para poder realizar un análisis más detallado y poder mostrar los datos en el mapa, los datos se agrupan por municipio haciendo la media de las columnas de información necesarias.

3. Cálculo del coeficiente Gini: para obtener una medida de desigualdad económica se calcula el coeficiente Gini de cada municipio junto con el índice de Reynolds-Smolensky (IRS).

4. Incorporación del nombre del municipio: se utilizó un documento csv que contenía la relación entre el código de municipio y su nombre (documento incluido en los datos proporcionados por Hacienda) para incluir el nombre de municipio en la visualización de los datos.

5. Ajustar los códigos de municipio para que coincidan con las coordenadas en el mapa: debido a que algunos códigos de municipio con los que trabajaba Hacienda no coincidían con los códigos para la visualización de los municipios en el mapa se tuvo que corregir este problema. (Ver sección 5.3 "Problemas encontrados")

6. Otros ajustes menores a los datos: para el correcto funcionamiento de la aplicación web se hicieron los ajustes necesarios a los datos como guardar los códigos como strings de tamaño fijo o el sexo como código numérico.

Por ultimo se devuelve el dataframe con todos los cambios y se guarda en la base de datos para poder mostrar la información en la página.

## Datos de la EPA

Los datos sobre la encuesta de población activa se obtuvieron de la sección datosmacro del periódico Expansión. [9]

Para la obtención y manipulación de los datos se realizaron los siguientes pasos:

1. Obtención de las URL: para incluir los datos de paro en nuestra base de datos se utilizó la técnica de webscraping de revisión de código HTML (ver sección 3.2). Para ello se fueron obteniendo los nombres de todas las comunidades autónomas para poder incluirlos en las direcciones URL. Después se hizo lo mismo con los nombres de las provincias y finalmente con los nombres de los municipios, de manera que se

consiguieron las direcciones URL para acceder a los datos de paro de todos los municipios.

2. Obtención de los datos: una vez conseguidas las URL, basta con leer los documentos HTML de cada dirección y extraer la información necesaria.

3. Asignación del código de municipio correspondiente: para poder asignar cada municipio con su correspondiente código (necesario para unificación de los datos y visualización en el mapa), primero se trató el nombre de los municipios para que coincidieran con los nombres de los datos de IRPF. Posteriormente se buscó y añadió el código correspondiente.

4. Otros ajustes menores: al igual que con los datos de IRPF se hicieron ligeros cambios en los datos y se calculó el codigo de provincia correspondiente (derivado del codigo de municipio).

Finalmente se devuelve el dataframe y se guarda en la base de datos.

## Datos de turismo

Los datos de turismo se consiguieron a traves de la página web Booking.[6] Para ello se obtuvo el número de alojamientos que tiene cada municipio.

Dado que la ejecución de esta parte puede llevar bastante tiempo, (más de 5 horas para los 9356 municipios) los datos obtenidos se guardan en un documento csv llamado turismoMunicipios, de esta manera una vez ejecutado no es necesario que vuelva a leer todos los datos de la página web.

Los pasos seguidos para la obtención de estos datos de turismo fueron los siguientes:

1. Obtención de los municipios a buscar: lo primero es saber que municipios nos falta por encontrar, para ello se lee un documento csv destinado para ello y que se irá actualizando.

2. Uso del buscador: a continuación se utiliza el nombre del municipio para introducirlo en el buscador de Booking, se ejecuta la búsqueda y se espera la respuesta del servidor.

3. Comprobar que la página es la correcta: algunas búsquedas nos llevaban a resultados que no eran los deseados (bien porque Booking no tenia

registrado ese municipio y nos llevaba a otro de nombre similar o porque buscaba alojamientos en los alrededores del municipio en vez de en el municipio en sí). Por esta razón se realizan estas comprobaciones de nombre y zona de los alojamientos y así evitamos leer datos incorrectos.

4. Leer los datos necesarios: una vez comprobado que la página es la correcta, se busca el número de alojamientos dentro del municipio

5. Guardar los datos: para evitar volver a tener que ejecutar toda la obtención de los datos de turismo desde el principio, cada cierto número de municipios se guarda en un documento csv los datos obtenidos hasta el momento. De esta forma podemos parar la ejecución y reanudarla en otro momento desde el punto donde la dejamos. De la misma forma también nos sirve en caso de algún error o fallo de conexión.

Una vez obtenidos los datos de turismo, se lee el csv con esos datos, se guardan en un dataframe y se hacen los ajustes necesarios para su incorporación a la base de datos.

## Datos electorales

En el TFG que sirve como base al presente [5] ya se introdujeron datos de elecciones municipales, en el presente trabajo, además se han introducido los datos de las elecciones generales de diciembre de 2015 y junio de 2016, ya que los resultados de las elecciones de abril de 2019 aún no estaban disponibles.

El objetivo del presente TFG es usar estos nuevos datos para la predicción de criminalidad y comprobar si la ideología política afecta en algo al número de delitos del municipio.

Con esta fuente de datos se realizaron los siguientes pasos:

1. Se introdujeron los nuevos datos: para ello se creó una nueva clase llamada "MirEleccionesGenerales"que reutilizaba el código de la primera parte del proyecto que sirve para introducir los nuevos datos a la aplicación.

2. Procesamiento de los datos: una vez obtenido el número de votantes de cada partido por municipio, es necesario adaptar los datos a algún tipo de valor numérico que sitúe cada partido en un lugar dentro de una escala política.

En nuestro caso hemos situado a los partidos siguiendo los datos de una encuesta del Centro de Investigaciones Sociológicas (CIS)[10], que registra la percepción que tienen los ciudadanos de cada partido (cero para la extrema izquierda, 10 para la extrema derecha)

Una vez hecho esto se calcula la media y la varianza de cada municipio y se introduce en una Serie de Pandas para su posterior uso en la obtención de las predicciones.

### Datos de criminalidad

Los datos de criminalidad fueron proporcionados por los tutores del proyecto. Éstos consistían en un documento csv con datos numéricos que guardaban correlación con un tipo de delito y que sirven de un estimador de los mismos. Todo ello con el fin de preservar la privacidad de los datos originales procedentes de la Guardia Civil. Asimismo, las transformaciones hechas por los tutores para obtener estos datos y el tipo de delito utilizado no son conocidas tampoco por el alumno.

En este caso no se guardan los datos directamente como tabla en la base de datos ya que se incluirán posteriormente junto a los datos de las predicciones.

## 5.2. Predicciones

En esta sección se explicará que uso se le ha dado a todo el conjunto de datos y los resultados que se han obtenido.

Una vez que ya tenemos todas las fuentes de datos añadidas a la base de datos ya podemos empezar con las predicciones. En esta parte del trabajo se busca obtener una predicción lo mas cercana posible al valor real.

### Obtención de las predicciones

Como datos de entrenamiento se han usado los siguientes datos:

- Datos del instituto nacional de estadística (INE): datos obtenidos en la primera parte de este proyecto. De aquí se usaron numerosas métricas que pudieran ayudar a la obtención de las predicciones. Algunas de estas métricas son: la población, la población discriminada por diferentes rangos de edad, el numero de extranjeros o el ratio de hombres/mujeres.

- Datos electorales (ver Sección5.1): se utilizan los datos referentes a la participación, al número de votos blancos o nulos y el porcentaje de votos que ha ido a parar a candidaturas mayoritarias. Además se añaden la media y la varianza de la escala política de cada municipio anteriormente calculada.
- Datos de IRPF (ver Sección 5.1): se utiliza el índice Gini así como la renta media del municipio después de aplicar los impuestos.
- Datos de turismo (ver Sección 5.1): se usa el número de alojamientos de los que dispone cada municipio.

Una vez leídos y procesados estos datos de la base de datos, se unifican todas las tablas en una sola, utilizando como identificador el código del municipio. A continuación se leen los datos de criminalidad, que van a ser los datos objetivos, y se añaden a la tabla que contenía los datos de entrenamiento.

Posteriormente se dividen los datos en datos de entrenamiento y datos objetivos y se obtienen las predicciones. Finalmente se calcula el error absoluto y se añade, junto a las predicciones, al dataframe que será guardado en la base de datos.

## Resultado de las predicciones

Los datos de entrenamiento que más influencia han tenido a la hora de realizar las predicciones, ordenados por grado de importancia han sido los siguientes:

1. Población: como es lógico, el número de habitantes de un municipio influye en la cantidad de delitos que se cometen en el mismo.
2. Número de alojamientos: en los lugares más turísticos es donde se registra mayor cantidad de delitos.
3. La varianza en la escala política: es decir, lugares donde hay mayores diferencias ideológicas dentro del espectro político.
4. Otros datos: el orden del resto de los datos puede variar entre ejecuciones, además, los que más importancia suelen tener son datos demográficos no concluyentes.

Del resultado de las predicciones podemos sacar algunas conclusiones:

- Al contrario de lo que se pudiera pensar en un principio, el índice Gini o la renta media no parecen afectar demasiado al número de delitos del municipio.
- La ideología política tampoco afecta a la cantidad de delitos lo que sí influye es la varianza de esta.
- Lo que afecta en gran medida es el número de alojamientos del municipio o provincia, es decir, el turismo influye considerablemente a la criminalidad de una localidad.

## 5.3. Problemas encontrados

En esta sección se explicarán los problemas e inconvenientes mas importantes que ocurrieron durante el desarrollo del proyecto.

### Ajustar códigos de los municipios

En un principio, cuando se empezó con el trabajo se pensó que el código de municipio de los datos de las declaraciones (IRPF) coincidía con el código correspondiente del municipio en el mapa. Pero una vez procesados los datos de IRPF, estos no se mostraban correctamente en el mapa (algunos no aparecían coloreados y otros no se correspondían). Esto mismo ocurría con el código de los datos de criminalidad. El motivo por lo que se pensó esto en un principio fue que en realidad, la gran mayoría de los códigos sí que coincidían, por lo que al comprobar únicamente unos pocos municipios, los códigos parecían coincidir.

La causa de este error venía porque en los datos de Hacienda aparecían más municipios de los que deberían. Esto se debe a que además de incluir municipios, incluía entidades de ámbito territorial inferiores al municipio (EATIM), además de antiguos municipios. Dado que el código de los municipios viene dado de forma ordenada, en el momento que aparecía una de estas entidades, el resto de municipios de la provincia se les asignaba un código de municipio diferente al que ya estaba registrado en la aplicación.

Para solucionar este problema se tuvo que adaptar el código de municipio de los datos de IRPF. Para ello se tuvo que relacionar, discriminando por provincias, el nombre del municipio de los datos de criminalidad con el nombre de los datos de Hacienda. De esta manera, después de hacer algunos cambios de formalización, se consiguió obtener el código correcto para la mayoría de los municipios.

Para los municipios que no fueron encontrados (porque venían escritos en diferente lengua o de distinta forma), se usó la librería difflib (ver Sección 4.2) que sirve para comparar cadenas. Con esta librería se buscó en nombre más cercano a cada nombre y después de comprobar que efectivamente se refieren al mismo municipio se actualiza el código de municipio al valor correcto. [24]

Una vez realizado este ajuste, los datos de IRPF ya podían ser relacionados con el resto de los datos y se mostraban correctamente en el mapa coroplético de la aplicación.

### Problemas con Selenium

Debido al gran número de municipios a los que se debía obtener sus datos de turismo de la página web Booking, el tiempo que tardaba la ejecución era de varias horas.

Debido a esto hubo varios problemas que ocurrían durante la ejecución. El primero de ellos era que al cabo de unos pocos cientos de municipios el ordenador se ralentizaba hasta que finalmente saltaba una excepción por falta de memoria. Este primer problema se solucionó haciendo una llamada explícita al recolector de basura cada cierto número de iteraciones.

En otras ocasiones el error venía porque Selenium no encontraba el elemento que se le pedía, esto normalmente se debía a que la página no había cargado correctamente, por lo que se tuvo que añadir la posibilidad de realizar varios intentos.

A pesar de estos ajustes, tras varias horas de ejecución, existía la posibilidad de que saltara alguna excepción (normalmente relacionada con problemas de conexión o respuesta del servidor) lo que obligaba a repetir la ejecución desde el principio. Para solucionar este problema se optó por que la aplicación fuera guardando en un archivo csv los datos que iba obteniendo. (ver Sección 5.1)

### Actividad Empresarial

En los datos proporcionados por Hacienda existían unas columnas (desde Par087_1 hasta Par088_6) que codificaban el sector profesional al que pertenecía el declarante. Se pensó que conocer en qué sector profesional trabaja cada persona podría resultar interesante para la elaboración de las predicciones, así que se decidió incluir esta información.

La codificación está dividida en: sección, división, agrupación, grupo, y epígrafe. Gracias a otro documento csv proporcionado junto a los datos de IRPF que indicaba a qué tipo de profesión pertenecía esta codificación, se decidió añadir a la muestra de datos IRPF la actividad empresarial a la que se dedicaba cada individuo.

Debido a la gran cantidad de declaraciones, casi 3 millones, y a las más de 1.400 actividades empresariales en las que se dividían las diferentes distribuciones, el tiempo de ejecución para encontrar todas las profesiones era de varias horas. Para solucionar esto se optó por utilizar las diferentes secciones de la codificación como datos de indexación, utilizando indexación multidimensional de forma que el acceso a la actividad empresarial se hiciera de manera mucho más rápida.

Una vez realizado este proceso se comprobó que el número de declaraciones que incluía esta información no era muy elevado (apenas un 15 %). Con esta escasa cantidad de datos había municipios con muy poca información sobre las actividades empresariales que se ejercían en el mismo. Por este motivo se determinó que estos datos no podían ser de utilidad para usarlos en las predicciones. A pesar de esto, se decidió conservar el código que obtenía las actividades empresariales en la aplicación por si pudiera ser de utilidad esta información para trabajos futuros.

## 5.4. Formación

Para adquirir los conocimientos necesarios para la realización del proyecto se han utilizado los siguientes recursos:

- Documentación de Pandas. [19]
- Documentación de Scikit-learn. [27]
- Documentación de Selenium. [22]
- Documentación Latex. [21]
- La memoria del TFG (versión 2018). [5]
- Documentación de Difflib. [24]
- Libro: Amartya Kumar Sen *La desigualdad económica*. [28]
- Otros recursos menores obtenidos en internet. (ver Bibliografía)

# Trabajos relacionados

Este apartado está dividido en dos secciones, en el primero se describirán algunos artículos de investigación relacionados con el trabajo realizado en este proyecto. En la segunda parte se mostrarán algunas páginas web que realicen una función similar a algunas de las características de este proyecto.

## 6.1. Artículos de investigación

### Predicción datos criminológicos

#### Predicción de delincuencia con datos públicos

Este artículo [25] trata al igual que en este proyecto de realizar una predicción de delincuencia utilizando datos públicos. De las predicciones obtenidas concluye que donde peor se comporta el modelo es en zonas con mayor ocupación turística. Por esta razón en el presente trabajo se ha seguido investigando en este tema y se han añadido los datos de turismo para obtener mejores predicciones.

### Relación Delitos-Turismo

Existen varios artículos científicos que relacionan el turismo con la tasa de criminalidad de un territorio. En este apartado vamos a explicar de que trata el siguiente artículo que analiza esta relación dentro del territorio español.

**¿Estimula el turismo la actividad criminal? Evidencia para las provincias españolas**

Este artículo [8] busca estimar el impacto que tiene la llegada de turistas a las diferentes provincias de España y su relación con el número de delitos contra las personas y contra el patrimonio.

Los resultados que obtiene parecen afirmar que realmente existe esa relación, la llegada de turistas tiene un impacto positivo y significativo en las tasas de criminalidad. Concretamente se calcula que la llegada de unos 100.000 turistas a una provincia provocaría un aumento aproximado de un 1.8 % en las tasas de criminalidad [8]

## 6.2. Páginas web

A continuación se mostrarán algunos ejemplos que tratan de manera similar los datos para obtener mapas parecidos a los que se pueden a llegar a generar en la aplicación

### Renta media

Existen diferentes páginas que muestran la información sobre las rentas medias por municipio pero todas obtienen los datos de la página web de la agencia tributaria. Esta página [29] muestra los datos en el mapa por provincias o municipios pero solo dibuja los municipios con mayor población. Al igual que en este proyecto tampoco disponen de los datos del País Vasco y Navarra.

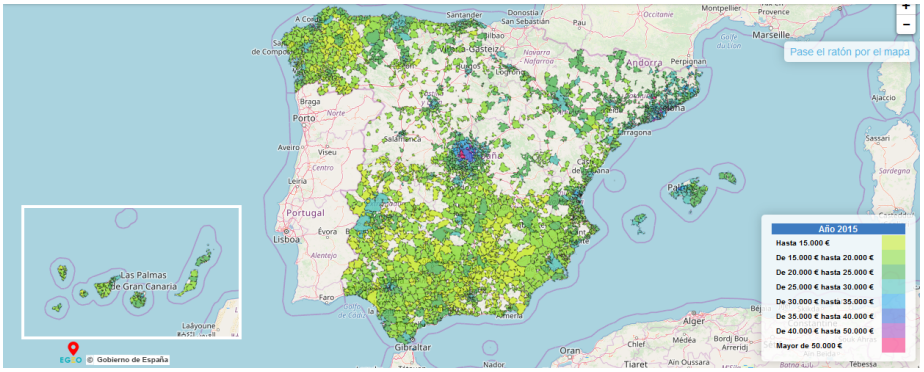


Figura 6.4: Mapa coroplético de la Agencia Tributaria. [29]

## Paro por municipios

La misma página [13] de donde se obtuvo el paro de los distintos municipios también tiene su propio mapa. En este caso, los mapas son prácticamente idénticos en los resultados.

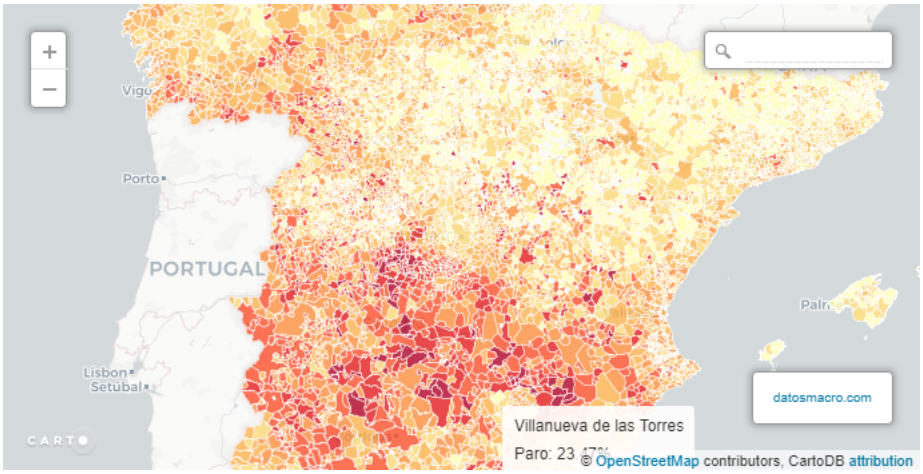


Figura 6.5: Mapa coroplético del paro por Datosmacro. [13]

## Resultados electorales

Con los datos de los resultados electorales hay numerosas páginas que permiten ver los resultados individuales de cada municipio, datos que obtienen de los resultados que publica el Ministerio del Interior. También hay numerosas páginas que muestran un mapa con la principal fuerza política en cada municipio, lo que se podría aproximar a una de las muchas posibilidades que se pueden representar en la aplicación de este proyecto. (votos nulos, número de mesas, total de votantes, posición dentro del espectro político, etc)

Figura 6.6: Un ejemplo de un mapa que refleja el resultado de las elecciones. [18]

## Delitos por Provincia

La página del Ministerio del Interior donde se publican los datos criminológicos también ofrece la posibilidad de mostrar un mapa con los resultados. En este caso y al contrario que en este trabajo, no permite reducir a una entidad territorial inferior a la provincia.

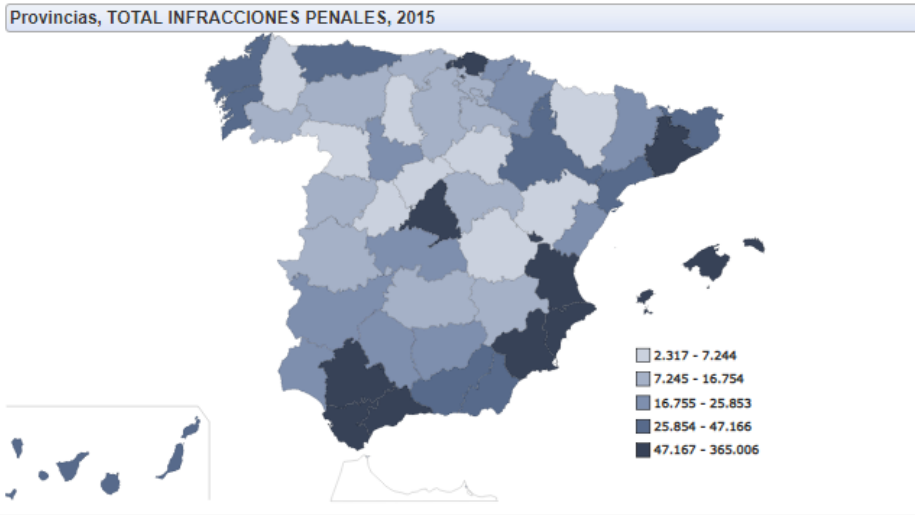


Figura 6.7: Mapa coroplético de los datos de criminalidad [12]

# Conclusiones y Líneas de trabajo futuras

En este apartado primero se explicarán las conclusiones obtenidas tras la realización del trabajo y de cómo se pudiera continuar el trabajo en el futuro

## 7.1. Conclusiones

En este proyecto se ha trabajado en diferentes temas para llegar al objetivo final de poder calcular unas predicciones sobre las tasas de criminalidad de los distintos municipios o provincias del territorio español. Para llegar a esta solución se ha trabajado con numerosos datos.

Primero se trataron los datos de Hacienda para introducirlos en la aplicación y se extrajeron y calcularon los datos necesarios para su utilización en la obtención de las predicciones. Para conseguir esto se tuvo que adquirir conocimientos en el manejo de bases de datos no relacionales.

A continuación se hizo un trabajo de *web scraping* para obtener otras fuentes de datos como fueron los datos de paro o, especialmente útil para las predicciones, los datos de turismo. También se actualizaron las fuentes de datos que ya incluía el proyecto en su versión de 2018 y se reutilizaron para obtener las predicciones.

Finalmente se hizo un trabajo de *machine learning* para poder calcular el número de delitos producidos en los diferentes municipios utilizando como datos de aprendizaje toda la información obtenida anteriormente para comprobar que las predicciones que se podían obtener de esos datos tenían sentido, así como medir la fiabilidad de las mismas.

En conclusión estoy satisfecho con el resultado del trabajo ya que se han conseguido cumplir los objetivos propuestos y he podido aprender mucho de diferentes temas.

## 7.2. Líneas de trabajo futuras

Este proyecto es la segunda parte de un trabajo que tenía por objetivo la creación de una aplicación web para la recopilación, tratamiento y visualización de datos públicos. [5] En esta parte se ha trabajado más en incluir nuevas funciones o diferentes formas de añadir nuevas fuentes de datos, que en mejorar las características que ya estaban en la aplicación. Por este motivo algunas de las líneas de trabajo futuras que ya se describían en la primera parte del proyecto siguen estando vigentes y también se podrían aplicar a los nuevos datos añadidos. Algunas de estas tareas futuras podrían ser las mejoras en las columnas calculadas o mejorar la representación de los mapas.

Además se podrían añadir las siguientes tareas a realizar en trabajos futuros:

- Añadir nuevas fuentes de datos que pudieran ser de utilidad para obtener mejores predicciones.
- Utilizar otras técnicas de machine learning que pudieran dar mejores resultados en el cálculo de las predicciones, como por ejemplo uso de redes neuronales.
- Agrupar las actividades empresariales calculadas en este proyecto en diferentes grupos para poder hacer uso de ellas en las predicciones.
- Utilizar los datos incluidos en la aplicación para realizar predicciones de otro tipo.
- Incluir otro tipo de fuentes de datos como por ejemplo, fuentes de datos RDF (*Resource Description Framework* [30])

# Bibliografía


[1] How to calculate gini coefficient from raw data in python. https://planspace.org/2013/06/21/how-to-calculate-gini-coefficient-from-raw-data-in-python/, 2013. [Internet; descargado 18-marzo-2019].

[2] Gini coefficient of inequality. https://www.statsdirect.com/help/default.htm#nonparametric_methods/gini.htm, 2016. [Internet; descargado 18-marzo-2019].

[3] Calculate the gini coefficient of a numpy array. https://github.com/oliviaguest/gini, 2017. [Internet; descargado 18-marzo-2019].

[4] JetRuby Agency. The most effective web scraping methods. https://expertise.jetruby.com/the-most-effective-web-scraping-methods-62e7e34ada69, 2018. [Internet; descargado 15-marzo-2019].

[5] Iván Arjona Alonso. Aplicación web para la recopilación, tratamiento y visualización de datos públicos. Master's thesis, Universidad de Burgos, 2018.

[6] Booking. Datos de turismo. https://www.booking.com/index.es.html, 2019. [Internet; descargado 04-junio-2019].

[7] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] Simón Planells Daniel Montolio. "¿estimula el turismo la actividad criminal? evidencia para las provincias españolas". *Research Gate*, 2012.

[9] Expansión Datosmacro. Paro por municipios. https://datosmacro.expansion.com/paro/espana/municipios, 2019. [Internet; descargado 07-junio-2019].

[10] Centro de investigaciones sociológicas. Barómetro de enero 2016. http://www.cis.es/cis/export/sites/default/-Archivos/Marginales/3120_3139/3124/Es3124mar.html, 2016. [Internet; descargado 5-abril-2019].

[11] Jefatura del Estado. Boletín oficial del estado. https://www.boe.es/buscar/act.php?id=BOE-A-2006-20764, 2006.

[12] Ministerio del interior. Portal estadístico de criminalidad. https://estadisticasdecriminalidad.ses.mir.es/jaxiPx/Datos.htm?path=/Datos1//l0/&file=01002.px, 2016. [Internet; descargado 15-junio-2019].

[13] Periódico Expansión-Datosmacro. Paro por municipios. https://datosmacro.expansion.com/paro/espana/municipios, 2019. [Internet; descargado 15-junio-2019].

[14] Alexandre Gramfort Vincent Michel Bertrand Thirion Olivier Grisel Mathieu Blondel Peter Prettenhofer Ron Weiss Vincent Dubourg Jake Vanderplas Alexandre Passos David Cournapeau Matthieu Brucher Matthieu Perrot Edouard Duchesnay Fabian Pedregosa, Gael Varoquaux. "scikit-learn: Machine learning in python". *Journal of Machine Learning*, 12:2825–2830, 2011.

[15] Krishni Hewa. A beginners guide to random forest regression. https://medium.com/datadriveninvestor/random-forest-regression-9871bc9a25eb, 2018. [Internet; descargado 4-abril-2019].

[16] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[17] Marco Antonio Moreno. ¿qué es el coeficiente de gini? https://www.elblogsalmon.com/conceptos-de-economia/que-es-el-coeficiente-de-gini, 2011. [Internet; descargado 13-junio-2019].

[18] Periódico El Mundo. Mapa municipal de los resultados 26j. https://www.elmundo.es/grafico/espana/2016/06/27/

57709ec1e5fdea870f8b4618.html, 2016. [Internet; descargado 15-junio-2019].

[19] Pandas. pandas: powerful python data analysis toolkit. http://pandas.pydata.org/pandas-docs/stable/, 2019. [Internet; descargado 10-febrero-2019].

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

[21] Latex Project. Documentation distributed with latex. https://www.latex-project.org/help/documentation/#documentation-distributed-with-latex, 2019. [Internet; descargado 6-junio-2019].

[22] Selenium Project. Selenium documentation. https://www.seleniumhq.org/docs/, 2019. [Internet; descargado 22-febrero-2019].

[23] PyDev. What is pydev? https://www.pydev.org/, 2019. [Internet; descargado 11-junio-2019].

[24] Python. difflib — helpers for computing deltas. https://docs.python.org/2/library/difflib.html, 2019. [Internet; descargado 8-abril-2019].

[25] José-Francisco Díez-Pastor Ivan Arjona Roberto Cuesta Calvo, Jesús Maudes Raedo. "predicción de delincuencia con datos públicos". *XVIII Conferencia de la Asociación Española para la Inteligencia Artificial*, pages 840–845, 2018.

[26] Sagar Shivaji Salunke. *Selenium Webdriver in Python: Learn with Examples*. CreateSpace Independent Publishing Platform, USA, 1st edition, 2014.

[27] Scikit-learn. Scikit-learn: Api reference. https://scikit-learn.org/stable/modules/classes.html, 2019. [Internet; descargado 5-mayo-2019].

[28] Amartya Kumar Sen. *La desigualdad económica*. Fondo de Cultura Economica, 2016.

[29] Agencia Tributaria. Estadística de los declarantes del irpf por municipios: 2015. https://www.agenciatributaria.es/AEAT/Contenidos_Comunes/La_Agencia_Tributaria/

Estadisticas/Publicaciones/sites/irpfmunicipios/2015/jrubikf4f548f53ce61f391620583e2ecbb94f0a134360d.html, 2015. [Internet; descargado 15-junio-2019].

[30] W3C. Rdf 1.1 turtle. https://www.w3.org/TR/turtle/, 2014. [Internet; descargado 25-junio-2019].

[31] Wikilibros. Manual de latex. https://es.wikibooks.org/wiki/Manual_de_LaTeX, 2017. [Internet; descargado 2-junio-2019].

[32] Wikipedia. Coeficiente de gini — wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Coeficiente_de_Gini, 2019. [Internet; descargado 13-junio-2019].

[33] Wikipedia. Latex — wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/LaTeX, 2019. [Internet; descargado 3-junio-2019].

[34] Wikipedia. Machine learning — wikipedia, la enciclopedia libre. https://en.wikipedia.org/wiki/Machine_learning, 2019. [Internet; descargado 09-junio-2019].

[35] Wikipedia. Validación cruzada — wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Validación_cruzada, 2019. [Internet; descargado 09-junio-2019].

[36] Hongyu Zhao Yiyi Liu. "variable importance-weighted random forests". *Quantitative Biology*, 5:338–351, 2017.