



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 — 6 de mayo
de 2018

Tutor: Dr. José Francisco Díez Pastor
y Dr. Jesús Manuel Maudes Raedo

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	6
Apéndice B Especificación de Requisitos	7
B.1. Introducción	7
B.2. Objetivos generales	7
B.3. Catalogo de requisitos	7
B.4. Especificación de requisitos	7
Apéndice C Especificación de diseño	9
C.1. Introducción	9
C.2. Diseño de datos	9
C.3. Diseño procedimental	9
C.4. Diseño arquitectónico	9
Apéndice D Documentación técnica de programación	11
D.1. Introducción	11
D.2. Estructura de directorios	11
D.3. Manual del programador	11

D.4. Instalación y ejecución del proyecto	11
D.5. Despliegue	12
D.6. Pruebas del sistema	13
Apéndice E Documentación de usuario	15
E.1. Introducción	15
E.2. Requisitos de usuarios	15
E.3. Instalación	15
E.4. Manual del usuario	15
Bibliografía	17

Índice de figuras

A.1. Burndown del sprint 0	2
A.2. Burndown del sprint 1	3
A.3. Burndown del sprint 2	3
A.4. Burndown del sprint 3	4
A.5. Burndown del sprint 4	5
A.6. Burndown del sprint 5	5
A.7. Burndown del sprint 6	6

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

El desarrollo del proyecto se ha llevado a cabo utilizando metodologías ágiles, basándose en la metodología *scrum* con algunas modificaciones (una sola persona y sin reuniones diarias).

Se aplicó una estrategia de desarrollo incremental, con iteraciones que llamaremos *sprints*.

El resultado de cada iteración es un entregable, sobre el que se discute en la reunión posterior a cada sprint.

Se realizó, en principio, una reunión a la semana con los tutores para exponer las modificaciones realizadas en el sprint anterior y planificar los cambios a realizar en la siguiente iteración.

Estas tareas están priorizadas por el tiempo estimado de su realización, se puede ver esta estimación en el enlace de cada sprint a sus tareas.

A continuación se va a realizar un breve resumen de las tareas realizadas en cada una de las iteraciones, así como la duración de cada sprint y el gráfico *burndown* correspondiente.

Sprint 0 (16/02/2018 - 02/03/2018)

Primer sprint del proyecto. En la reunión de planificación de este primer sprint se discute de forma general de lo que va a tratar el proyecto.

Las tareas realizadas durante este sprint fueran la creación y configuración del repositorio y sobre todo investigar sobre las tecnologías y herramientas que se podían utilizar.

Se investigaron posibles fuentes de datos para implementar más adelante: INE, sepe, aeat.

La duración fue de dos semanas en lugar de una para poder documentarse sobre todos los aspectos relevantes del proyecto.

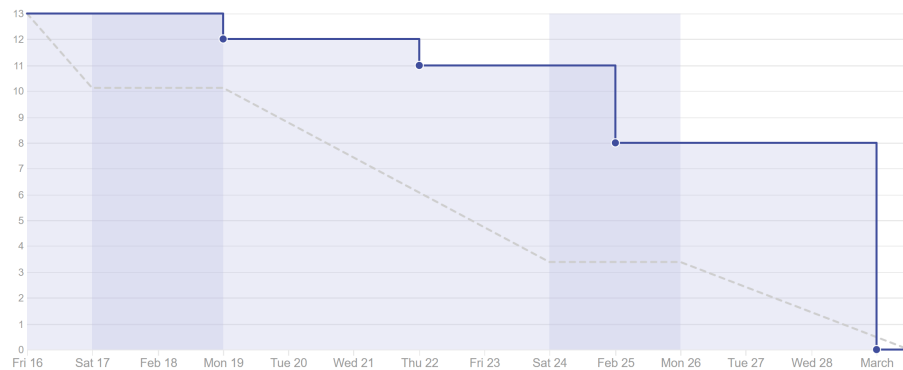


Figura A.1: Burndown del sprint 0

Tareas del sprint 0 en Github

Sprint 1 (03/03/2018 - 08/03/2018)

Un objetivo de este sprint es investigar alternativas de bases de datos no relacionadas que se podrían utilizar. Se ha elegido MongoDB.

El otro objetivo es empezar a implementar prototipos con las fuentes de datos que se habían encontrado en el sprint anterior. Se implementaron prototipos del INE, de la agencia tributaria y del sepe.

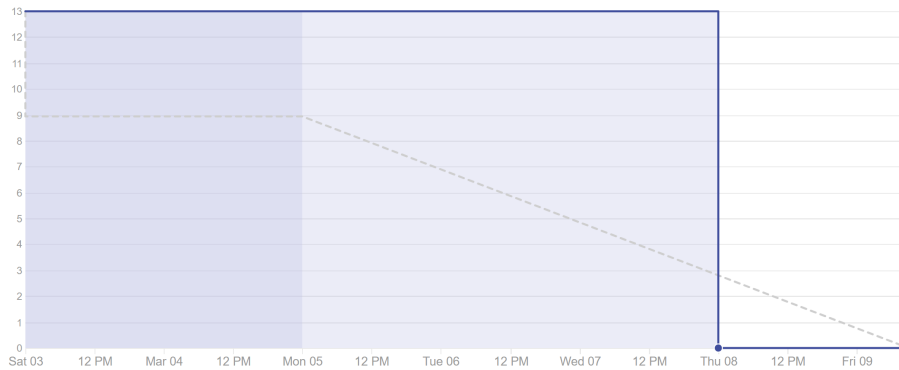


Figura A.2: Burndown del sprint 1

[Tareas del sprint 1 en Github](#)

Sprint 2 (09/03/2018 - 15/03/2018)

El primer objetivo de este sprint es crear la estructura de la página web con Flask. Utilizando un modelo vista-controlador. También se utiliza bootstrap para ahorrar trabajo en el diseño.

Se implementó un prototipo de la carga de datos hacia la base de datos y otro para la descarga de datos desde la base de datos para ser mostrados.

Se hizo una implementación de las fuentes de datos a partir de los prototipos del sprint 1 de modo que se pueda cargar todas las fuentes de manera automática.

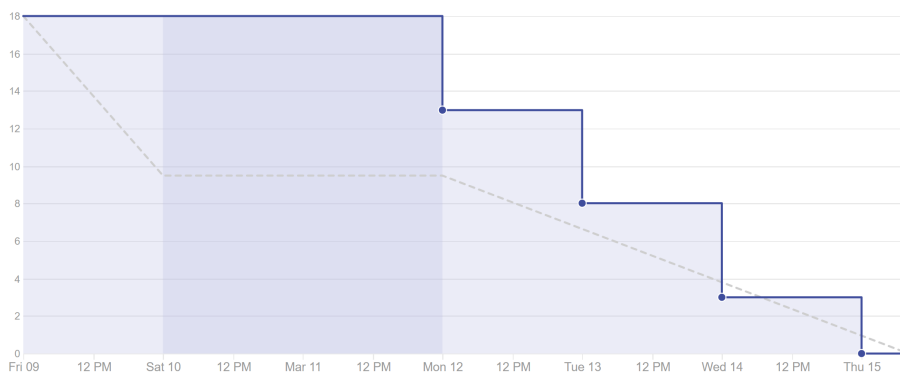


Figura A.3: Burndown del sprint 2

[Tareas del sprint 2 en Github](#)

Sprint 3 (16/03/2018 - 22/03/2018)

En este sprint se sopesaron varias plataformas para hacer el despliegue de la web. De ellas se eligió DigitalOcean y Nanobox.

Se realizó el despliegue utilizando estas plataformas. **Web desplegada.**

Se investigaron los posibles riesgos de seguridad como inyecciones NoSQL.

Se implementó un formulario para la consulta de datos en la página web. En esta primera aproximación se podía hacer una consulta comparando con una columna de una de las fuentes de datos.

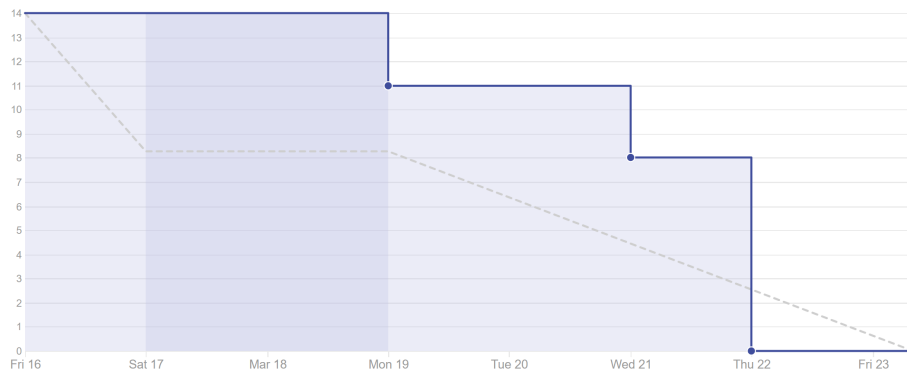


Figura A.4: Burndown del sprint 3

Tareas del sprint 3 en Github

Sprint 4 (23/03/2018 - 13/04/2018)

Este sprint coincide con semana santa, por lo que dura una semana más de lo habitual y la carga de trabajo también es mayor.

Se corrigieron errores en los tipos de las fuentes de datos al tratar con números como cadenas.

Se implementó una forma de descargar las consultas a partir del formulario.

Se mejoró interfaz gráfica y el formulario de consulta.

Se modificaron las fuentes de datos para corregir errores y añadir el código de municipio a todas ellas para más tarde poder unir las.

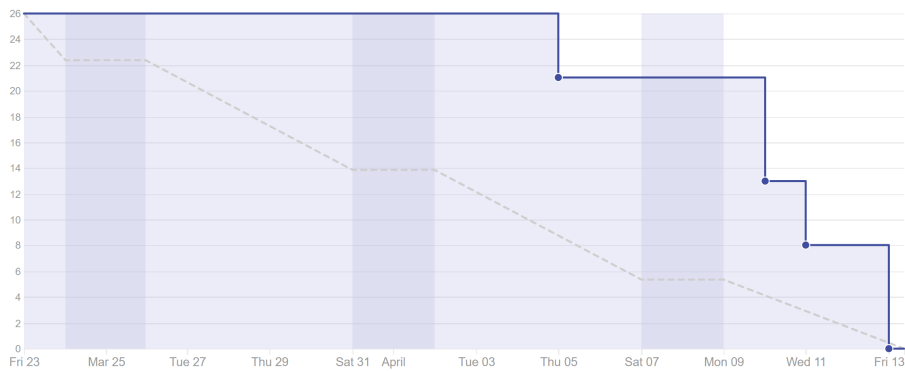


Figura A.5: Burndown del sprint 4

Tareas del sprint 4 en Github

Sprint 5 (14/04/2018 - 25/04/2018)

Este sprint se dedicó a empezar a documentar la memoria y corregir algunos errores en la página de consulta como el reenvío de formularios en firefox y la descarga de consultas en json y csv.

También se añadió una descripción a la fuente de datos para explicar de qué se trata cada una en la interfaz web.

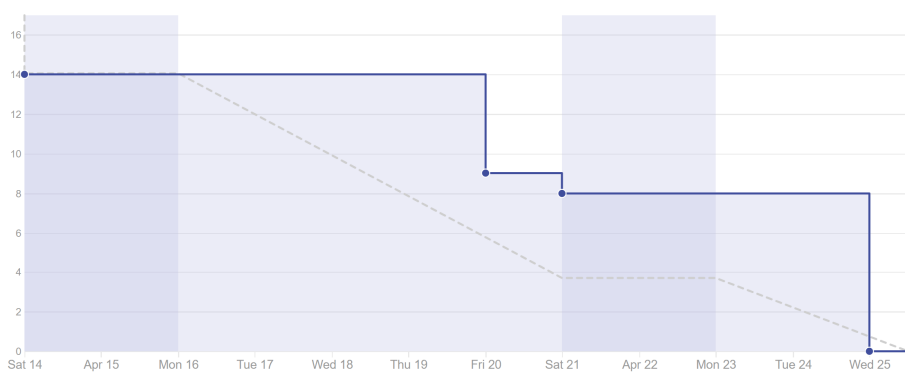


Figura A.6: Burndown del sprint 5

Tareas del sprint 5 en Github

Sprint 6 (26/04/2018 - 02/05/2018)

El objetivo principal de este sprint fue implementar la posibilidad de juntar varias subconsultas mediante un join. Otra característica implementada es la de avisar al usuario si se ha sobrepasado el límite de columnas especificado, para que pueda filtrar más fino.

También se consideró hacer cambios en el modelo de datos, pero debido a la alta dimensionalidad se ha dejado como estaba en el sprint anterior.

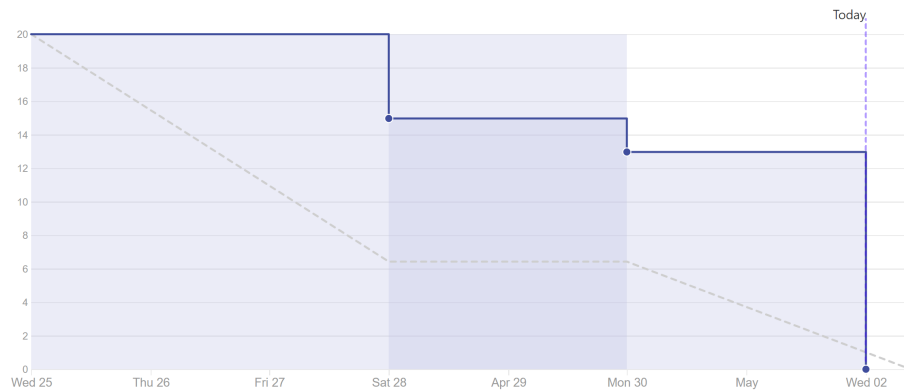


Figura A.7: Burndown del sprint 6

[Tareas del sprint 6 en Github](#)

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

- B.1. Introducción
- B.2. Objetivos generales
- B.3. Catalogo de requisitos
- B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice *D*

Documentación técnica de programación

D.1. Introducción

En este anexo se explica todo lo que tiene que conocer el programador tanto para instalar y ejecutar la aplicación como para poder seguir con el desarrollo.

D.2. Estructura de directorios

D.3. Manual del programador

D.4. Instalación y ejecución del proyecto

Instalación

MongoDB

Antes de empezar con la instalación de la aplicación tenemos que instalar la base de datos. Se ha utilizado una base de datos no relacional MongoDB.

Concretamente se instaló la versión *3.6.4 Community Server* [1] para windows.

También se ha utilizado *MongoDB Compass* [1] para visualizar el contenido de la base de datos. Esta herramienta es opcional.

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

Python

Esta aplicación se ha desarrollado utilizando la version 3.6.4 [2], por lo que se recomienda utilizar esta versión o una posterior. En cualquier caso debe de instalarse Python 3 o superior para evitar incompatibilidades.

Todos los paquetes utilizados en la aplicación están listados en en ficheros requirements.txt junto con sus correspondientes versiones.

Habrá que instalar todas las dependencias utilizando pip¹:

```
pip install -r requirements.txt
```

Ejecución

Actualización de la base de datos

Antes de ejecutar la aplicación tendremos que construir la base de datos con el contenido de todas nuestras fuentes por primera vez.

Para ello simplemente hay que ejecutar el fichero *actualiza-fuentes.py*. Puede tardar un rato debido a la gran cantidad de datos que se van a cargar.

Este paso puede repetirse cada vez que se quiera actualizar las fuentes de datos. Puede ser utilizar ejecutarlo una vez al mes para mantener al día las fuentes con datos mensuales.

```
python actualiza-fuentes.py
```

Servidor web

El servidor Flask ya se ha instalado como un paquete, por lo que no necesita más instalación. Para ejecutarlo hay un script *run.py* que nos lanza el servidor en el puerto 5000.

Una vez lanzado podremos acceder a la página web desde **localhost:5000**.

```
python run.py
```

D.5. Despliegue

Para desplegar la aplicación se ha obtado utilizar como servidor en la nube DigitalOcean y Nanobox como microservicio para facilitarnos la instalación de la máquina en la nube y la configuración del servidor.

¹Gestor de paquetes de Python [?]

Instalación

Digital Ocean

Primero tendremos que registrarnos en [DigitalOcean](#) y obtener el token de nuestro usuario. Tendremos que poner una tarjeta de crédito de la que nos cobrarán el importe del servidor.

Podría utilizarse otro proveedor de hosting como *Amazon Web services* o *Google Compute*.

Nanobox

Después nos registraremos en [Nanobox](#) y creamos una nueva aplicación utilizando el token que hemos obtenido antes en DigitalOcean.

Ahora elegiremos el plan que queramos contratar. Para este proyecto será suficiente con el plan más básico de 5\$, 1 CPU y 1GB de ram.

Tras crear la aplicación instalamos el [cliente de nanobox](#). La primera vez que lancemos un comando nos pedirá los datos para iniciar sesión en nuestra cuenta.

En el fichero *nanobox.yml* tenemos la configuración con los componentes que se van a instalar y los comandos que se ejecutan al desplegar la aplicación.

Despliegue

Una vez todo instalado y configurado, cada vez que queramos actualizar la aplicación del servidor nos situamos en la carpeta del proyecto y lanzamos el siguiente comando:

```
nanobox deploy
```

Con esto, de forma transparente para nosotros, se crea una máquina virtual con nuestro proyecto en la que se instalan la base de datos, el entorno de ejecución y los paquetes y se ejecuta la aplicación en el servidor.

D.6. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Bibliografía

- [1] MongoDB. Mongoddb download center. <https://www.mongodb.com/download-center?jmp=nav#community>, abril 2018. [Internet; descargado 29-abril-2018].
- [2] Python. Python 3.6.4. <https://www.python.org/downloads/release/python-364/>, diciembre 2017. [Internet; descargado 29-abril-2018].